國立交通大學

資訊工程學系

博士論文

行動網路內安全多路徑連結架構之設計

The Design of Secure Multi-Homed Architecture in Mobile Networks

研 究 生：羅嘉寧

指導教授：謝續平　博士

中華民國九十四年七月

行動網路內安全多路徑連結架構之設計
The Design of Secure Multi-Homed Architecture in Mobile Networks

研究生：羅嘉寧　　Student：Jia-Ning Luo

指導教授：謝續平 博士　　Advisor：Dr. Shiuhpyng Shieh

國立交通大學
資訊工程學系
博士論文

A Dissertation
Submitted to Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Computer Science and Information Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年七月

行動網路內安全多路徑連結架構之設計

學生：羅嘉寧　　　　　　　　　　　　　　　　指導教授：謝續平 博士

國立交通大學資訊工程學系

摘　　要

隨著行動通訊的發展與普及，現今的行動裝置常具有兩種以上的網路介面，稱之為多路徑連結架構網路。在一個多路徑連結架構網路中，行動裝置可以根據不同的應用與需求，即時選擇一個最佳的網路介面，如此可以承受多種網路障礙，網路延遲或封包遺失之情形，進而提升整體網路的傳輸品質。然而，設計一個完善的多路徑連結網路需要多方面的考量，因此本論文將探討以下三項主題：

本論文所要探討的第一個主題是網際網路的擴充問題。在現今的網路架構中，最常用來解決網際網路擴充問題的方式是使用網際網路位址轉譯器 (NAT)。然而網際網路位址轉譯器有許多的缺失，例如無法連結至多階層私有網路中，以及可能存在網際網路位址衝突的問題。因此我們提出了一種解決方案，稱之為 MRSIP 架構，以取代網際網路位址轉譯器。使用 MRSIP 架構將使得在 NAT 架構下的前述問題加以解決。

本論文的第二部份著眼於改進一個在多連結路徑網路下所使用的通訊協定，稱之為資料流控制傳輸協定 (Stream Control Transmission Protocol; SCTP)。然而因為原先資料流控制傳輸協定並不是針對多連結路徑網路而設計，因此我們探討使用該傳輸協定的不足，如路徑選擇及網路轉換效率問題等，並針對這些問題提出一系列的解決方案及加以分析。.

在本論文的第三部分中，我們提出了一個新的身份確認及金匙交換協定，以用於多連結路徑網路。在這個協定裡，並不需要有一個公正的第三者以作為金匙交換的中介者，如此可以避免因網路傳輸中斷導致無法進行身份確認的情形。在這個身份確認及金匙交換協定中，我們解

決了以前學者發現的問題並加以改良，而且只需從事較少的指數運算及記憶體，因此非常適用於只具備些許運算能力及記憶體的行動裝置中。

關鍵詞：密碼學，多連結路徑網路，身份確認，網際網路位址轉譯器，資料流控制傳輸協定

The Design of Secure Multi-Homed Architecture in Mobile Networks

Student：Jia-Ning Luo　　Advisors：Dr. Shiuhpyng Shieh

Department of Computer Science and Information Engineering
National Chiao Tung University

ABSTRACT

With the growth of mobile computing, currently a mobile device may have one or more network interfaces, which is called as 'multi-homed network.' In a multi-homed network the data connections can be placed in the best possible interface or forwarded through several paths thereby decreasing end-to-end delivery delay and increasing the network capacity. Also, using a multi-homed network can improve the network performance because it is against network failure or network partitioning. To address these situations, this thesis investigates solutions in multi-homed architecture in mobile networks. Research consists three parts:

Part one investigates the solution of Internet scaling problem. The well-known solution of Internet scaling problem is using the Network Address Translator (NAT). However, there are still many problems cannot be solved by NAT. For example, NAT cannot access to multi-level private network, or prevent the address collision. To overcome these problems, we propose the MRSIP framework to replace the NAT.

Part two investigates the enhancement of communication protocol to be used in multi-homed network architecture, the Stream Control Transmission Protocol (SCTP). Since the original designing of SCTP protocol is not to be used in multi-homed network, we discuss the drawback of SCTP protocol such as path selection and changeover decision problems, and propose several algorithms to solve these problems in the SCTP protocol.

In part three, we propose a new authenticated key agreement protocol to be used in the multi-homed network environment. In the propose protocol, the key information center is needed only when the secure network system is being set up or when new users request to register. Furthermore, our protocol needs fewer exponential computations and memory, which is suitable for the low-end mobile devices. Finally, we discuss the possible extensions and conclude.

Keywords: Cryptography, key agreement, multi-homed network, authentication, NAT, SCTP

Acknowledgement

I am grateful to my advisor Shiuhpyng Shieh for his advices and patience over these years. Special thanks are also to the other readers of this work including L. M. Tseng, W.B. Lee, R. J. Hwang, J.H. Chiu, R. J. Chen, and S. K. Hwang. Their valuable comments and suggestions have helped make this thesis much better.

I also thank for all the classmates, seniors and juniors in the distributed system and network security lab, especially C. T. Lin, C. H. Wu and S. I. Hwang. Without their support, I cannot finish this thesis smoothly.

Finally, I dedicate this thesis to my family, and my girlfriend Y. T. Yang, for their understanding and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the current Internet environment, enterprises often use private address space to setup their network environment for a variety of reasons. Three major reasons that use private network are firewall, Virtual Private Network (VPN) and Network Address Translator (NAT). By having more address space, this enables operationally and administratively convenient addressing schemes as well as easier growth paths. For example, companies run firewall systems may use private address space to isolate internal networks. People also use NAT to build home networks when they cannot get enough IP addresses from their Internet Service Provider (ISP).

The hosts that reside in a private network are unreachable by the Internet routers by default because Internet routers will not route packets that come from private network addresses. NAT is proposed to solve half of this problem. However, Internet users may wish to communicate with those unreachable hosts, especially when they use peer-to-peer applications such as Internet-Telephony, MSN, Internet game, file sharing or terminal service.

The cascade private network occurred when users in a private network build another private network inside. For example, assume one big company built a private network to reduce the amount of public IP addresses. The research department in this company may wish to build a small private network to protect their sensitive data. Assume these two private networks are interconnected by a NAT device. To allocate the resources inside the private networks and to connect with them becomes an important issue.

Furthermore, In the quest for network redundancy, enterprises often subscribe more than one leased line from several network service providers to build a multihomed network environment. In a multi-homing network the data connections can be placed in the best possible interface or forwarded through several paths thereby decreasing end-to-end delivery delay and increasing the network capacity.

Another aspect of performance improvement due to multi-homing is against network failure or network partitioning. To address these situations, this thesis investigates solutions in multi-homed architecture in mobile networks.

This dissertation is organized as follows:

Chapter 2 briefly introduce the related work of Internet scaling problems and multi-homed network architecture, includes the Network Address Translation (NAT) and it's variants, the multi-homing protocol such as Stream Control Transport Protocol (SCTP), and the authentication protocols to be used in the multi-homed networks.

Chapter 3 investigates the MRSIP architecture that to be used in communicating . The basic design objectives of MRSIP can be summarized as follows: transparent-access capabilities with the private network, cascade private network architecture, redundant path reducing between source and destination nodes, and reduce the address-collision probability.

Chapter 4 discuss the problems exists in the SCTP, including the path selection problem and failover problem. We proposed four algorithms to enhance the weakness of the original SCTP protocol.

In chapter 5, we propose a new authenticated key agreement protocol to be used in the multi-homed network. In the propose protocol, the key information center is needed only when the secure network system is being set up or when new users request to register. A new subsequent authentication phase is used to reduce the computation overhead and network traffic. Finally, we discuss the possible extensions and conclude.

Chapter 2

Related Work

In the quest for network redundancy, enterprises often subscribe more than one leased line from several network service providers to build a multi-homing network environment.

In a multi-homed network, the data connections can be placed in the best possible interface or forwarded through several paths thereby decreasing end-to-end delivery delay and increasing the network capacity. Another aspect of performance improvement due to multi-homing is against network failure or network partitioning.

The most recently popular multi-homed environment is the mobile networks. Current mobile devices are often equipped with several network interfaces such as WLAN, GPRS, IrDA, or Bluetooth. During the communication period, the mobile device is able to migrate from one associated network behind an interface to the other.

In this chapter, we briefly introduce three major problems impacting the internet: the Network Address Translation (NAT), the multi-homing protocol such as

Stream Control Transport Protocol (SCTP), and the authentication protocols to be used in the multi-homed networks.

2.1 Network address translation (NAT) and it's variants

Network Address Translation is a method by which IP addresses are mapped from one realm to another, in an attempt to provide transparent routing to hosts [54]. NAT has been widely applied in network equipment due to the leakage of IPv4 addresses space. NAT's fundamental role is to alter the addresses in the IP header of a packet [56]. Many applications cannot work under NAT environment because NAT servers try to modify the IP headers to provide transparent routings. These applications include well-known FTP and H.323 protocols [14]. To solve this problem, application-level gateway (ALG) is used [65]. Application-level gateway works fine with common protocols such as FTP, SNMP and VoIP, but does not provide end-to-end security such as Kerberos, RPC, and IPsec [19, 52, 4, 40, 30].

Network Address Translation (NAT, as shown in Figure 2.1) has become a common Internet technology for a variety of reasons. In figure 2.1, an NAT router with two interfaces, 10.1.1.1 and 140.113.215.1, provides transparent routing between two address realms, the Intranet realm (using private IP addresses) and the Internet realm (public IP addresses). NAT converts the inside addressing realm (e.g., 10.1.1.2) into another address realm (e.g., 140.113.215.2) before forwarding packets to public networks. In other words, NAT can be used to connect a private network, which uses unregistered private IP addresses with a public network that uses limited registered IP addresses.

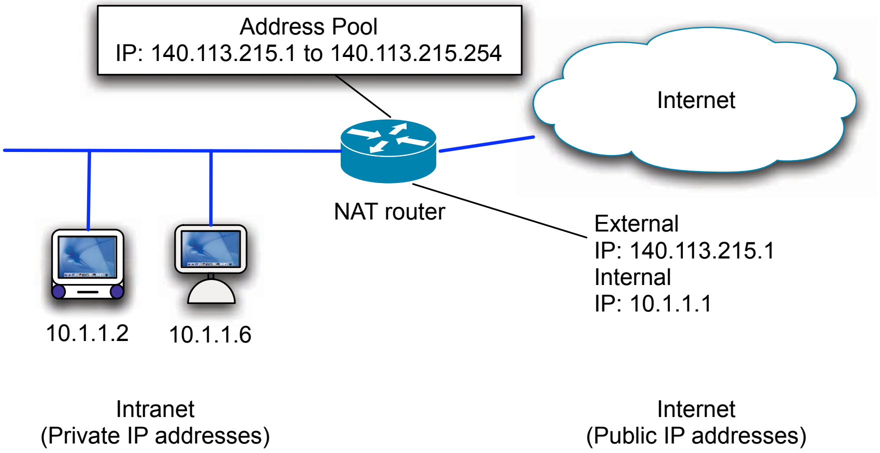


Figure 2.1: Network Address Translation Framework

NAT allows hosts within a private network to uni-directionally access remote hosts in the external network. The IP addresses of the hosts in the private network are only unique within the network and may not be valid in the external network.

Traditionally, NAT is used to bind many private IP addresses and TCP/UDP ports into one globally unique IP address and its TCP/UDP port. Such kind of NAT methods allow hosts within a private network to transparently access hosts in the external network.

2.1.1 NAT variants

There are many variations of address translation that lend themselves to different applications: static NAT, dynamic NAT, network address port translation (NAPT), bi-directional NAT and twice NAT.

- Static NAT

  With static NAT, a block of external addresses are set aside for translating addresses of hosts in a private domain as they originate sessions to the external domain. For packets outbound from the private network, the source IP address and related fields such as IP, TCP, UDP, and ICMP header checksums are translated. For inbound packets, the destination IP address and the checksums as listed above are translated.

In static NAT, the computer with the IP address of 10.1.1.2 will always translate to 140.113.215.2, as shown in table 2.1.

Source IP Address	Destination IP Address
10.1.1.2	140.113.215.2
10.1.1.3	140.113.215.3
10.1.1.6	140.113.215.6

Table 2.1: Address Mapping of Static NAT

- Dynamic NAT

  With dynamic NAT, all the available public IP addresses are stored in one address pool. NAT boxes will create (or update) an address mapping only when a session is established.

  In dynamic NAT, it also establishes a one-to-one mapping between private and public IP address, but the mapping could vary depending on the public IP address available in the pool. For example, the computer with the IP address of 10.1.1.2 will translate to the first available address in the range from 140.113.215.2 to 140.113.215.254, which is 140.113.215.2. Later the computer with the IP address of 10.1.1.6 will translate to the second available address, 140.113.215.3, as shown in table 2.2

- Network Address Port Translation (NAPT)

  NAPT extends the notion of translation one step further by also translat-

Source IP Address	Destination IP Address
10.1.1.2	140.113.215.2
10.1.1.6	140.113.215.3
10.1.1.3	140.113.215.6

Table 2.2: Address Mapping of Dynamic NAT

ing transport identifier (for example, the TCP and UDP port numbers, or ICMP's query identifiers). This allows the transport identifiers of a number of private hosts to be multiplexed into the transport identifiers of a single external address. That is, NAPT allows a set of hosts to share a single external address. NAPT can be combined with traditional NAT so that a pool of external addresses are used in conjunction with port translation.

For packets outbound from the private network, NAPT would translate the source IP address, source transport identifier and related fields such as IP, TCP, UDP and ICMP header checksums. Transport identifier an be one of TCP/UDP port or ICMP query ID. For inbound packets, the destination IP address, destination transport identifier and the IP and transport header checksums are translated.

In table 2.3, the TCP packet from 10.1.1.2 with port 80 will be translate to 140.113.215.2 with port 80, and the TCP packet from 10.1.1.6 with port 80 will be translate to the same IP address 140.113.215.2 but the port number is different.

Source IP Address:Port	Destination IP Address:Port
10.1.1.2:80	140.113.215.2:80
10.1.1.6:80	140.113.215.2:81
10.1.1.3:80	140.113.215.2:82

Table 2.3: Address Mapping of NAPT

- Bi-directional NAT or Two-Way NAT

  By using traditional NAT, hosts in an external network are not able to initiate a session request to a host inside the private network. This is in contrast with anther kind of NAT, Bi-directional NAT (see RFC 2663) [56] (also known as Two-way NAT).

  A bi-directional NAT server allows sessions in both inbound and outbound directions. When the connection is established in either direction, the private network address is statically or dynamically mapped to a globally unique address. The assumption of bi-directional NAT is that Fully Qualified Domain Names (FQDN) of hosts both in private networks and public networks are end-to-end unique. Therefore, a DNS Application Level Gateway (DNS-ALG) [57] is used with bi-directional NAT to facilitate name to IP address and TCP/UDP port mapping.

- Twice NAT

  Twice NAT is a variation of NAT in that both the source and destination IP addresses are translated by NAT box. This is in contrast to Traditional-NAT and Bi-Directional NAT, where only one of the addresses (either source or destination) is translated.

  Twice NAT is necessary when private and external realms have address collisions. The most common case where this would happen is when a site may have changed from one provider to another, but chosen to keep the addresses it had been assigned by the first provider. In such cases is that the address of the host in the external realm may have been assigned the same address as a host within the local site. If that address were to appear in a packet, it would be forwarded to the internal node rather than through the NAT device to the external realm. Twice-NAT attempts to bridge these realms by translating both source and destination address of an IP packet, as the packet transitions realms.

2.1.2 Application level gateway

Despite the convenience brought by NAT, there are still some limitations and security issues when NAT is applied. Not all applications are able to pass through NAT server transparently; especially those that carry IP address and TCP/UDP port information inside their payloads. In such applications, an Application Level Gateway (ALG, as shown in Figure 2.2) [56] is required to perform address translations on application packets that contain IP address and TCP/UDP port information for outbound sessions. In Figure 2, all payloads routed by the NAT server will be forwarded to the ALG. The ALG interprets the payloads and performs the necessary address translations on the payloads of the connection. However, ALG is not intended for inbound connections. If a session is initiated from the public network, ALG cannot break through the NAT server, either. If inbound sessions must be allowed, ALG must be integrated with the NAT server. For some simple protocols that use fixed ports only, NAT with port forwarding [63] can be performed on the NAT server. Port forwarding function on the NAT server can forwards all packets from certain ports to their dedicated servers. In this case, we can also treat the NAT server as a virtual server that distributes the traffic among designated server farms.

The combination of NAT server and an ALG cannot provide end-to-end security; especially when the NAT server and the application level gateway are not located in a trusted boundary. Furthermore, an ALG may become a bottleneck and forwarding throughput of the border router combined with the NAT server could be degraded considerably.

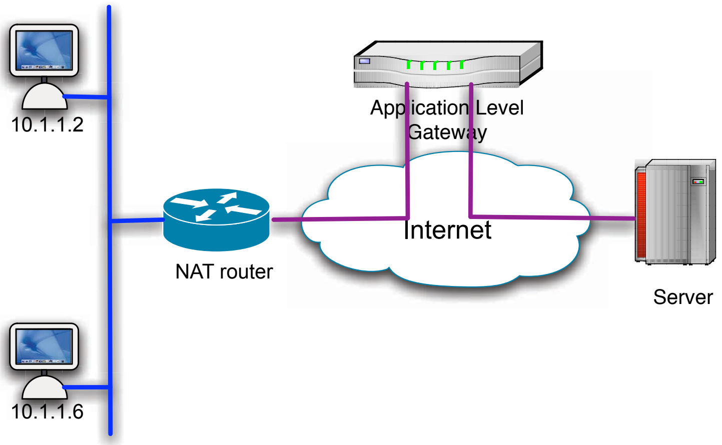


Figure 2.2: Application Level Gateway (ALG)

2.1.3 NAT limitations

There are some other limitations when using NAT to handle translations between private addresses and a small range of addresses that were allocated for public use. For example, since a NAT router keeps the mapping relations of all sessions established through it, the requests and responses of those sessions must be routed via the same NAT router. For this reason, it is usually recommended to combine a NAT router with a border router in a domain. However, such configuration makes an NAT router the target of attacks and intrusions.

No matter what kind of NAT method is used, the TCP/IP checksums in the forwarded packets cannot be encrypted since NAT requires the capability to translate any part of the headers and packets according to the referred addressing scheme. If data encrypted within an IP packet contains information that must be translated, it becomes extremely difficult for a NAT server to perform any network address translation. Thus, any host applying encryption to TCP/IP checksums should be assigned a globally unique IP address, exempted from NAT.

In addition, NAT may potentially break the end-to-end nature of applications on the Internet; therefore, the use of NAT threatens the end-to-end security of the Internet. Actually, many security protocols exchange IP addresses or TCP/UDP port related information in their authentication packets. These security protocols are vulnerable to disability for passing through the NAT server. As a result, a certain group of security protocols may fail when applying such addresses translations to their authentication packets.

2.1.4 Realm-specific IP

Realm-Specific IP (RSIP) is another approach that based on the concept of granting a host from one addressing realm a presence in another addressing realm by allowing it to use IP address from the second addressing realm [3]. An RSIP server/gateway replaces the NAT box, and RSIP-aware hosts on the private network are referred to as RSIP clients. RSIP clients inside a private can lease a public IP address to communicate with outside hosts. The RSIP protocol is extended to support both IKE (a UDP application) and the IPsec-defined AH and ESP headers [39, 28, 29, 12].

Due to the limitation of NAT mechanism, Realm Specific IP (RSIP) provides an alternative solution. RSIP is based on the concept of granting a host from one address-ing realm a presence in another addressing realm by allowing it to use IP addresses from the second addressing realm. An RSIP server replaces the NAT router, and RSIP-enabled hosts are referred to as RSIP clients.

An RSIP server maintains a pool of IP addresses to be leased by RSIP clients. Upon client request, the RSIP server allocates a public address to the client. An RSIP client may lease more than one public address from the RSIP servers. Once an address is allocated to a particular client, only that specific client may use the address until the address is returned to the pool. An RSIP server may provide all the NAT functions. An RSIP-disabled host inside the private network could still communicate with outside networks by using traditional NAT.

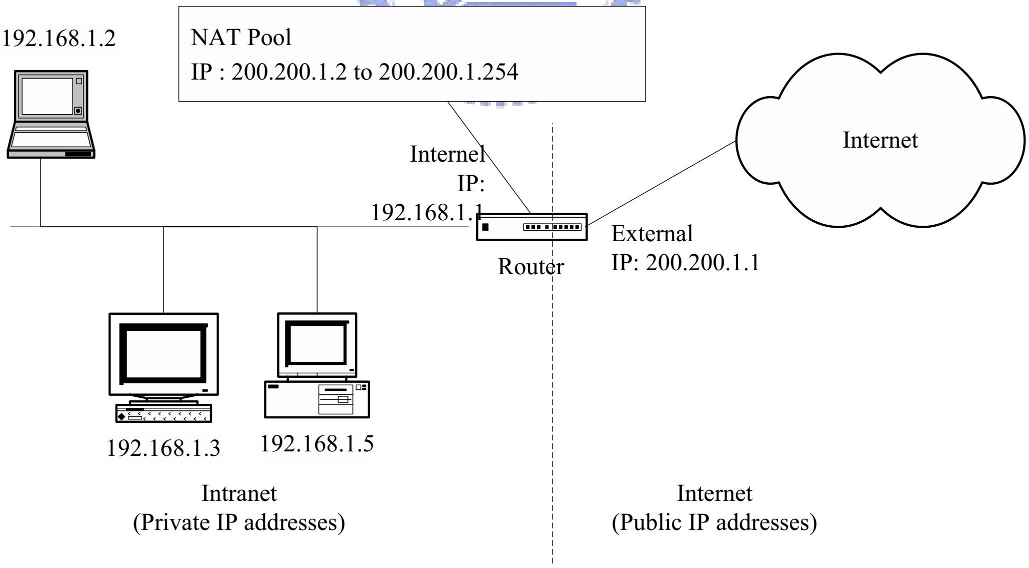


Figure 2.3: Example Network of RSIP architecture

As shown in Figure 2.3, an RSIP server C with two interfaces, 192.168.1.1 and 200.200.1.1, connects both public and private networks. Host A is an RSIP client with private address 192.192.1.5, and host B with address 140.113.216.164 belongs to public network. C has a pool of public IP addresses, 200.200.1.2 to 200.200.1.254, which it can assign to host A and other RSIP clients in private network.

When RSIP client A wants to connect to host B, client A first requests a public IP ad-dress, 200.200.1.5, from RSIP server C. Client A tunnels data packets across private network to C. C stripping off the outer headers and routing the inner packets to B. When a packet from B arrives at C, C will also tunnel those packets to A.

Since A can lease a public address from RSIP server C to communicate with B in public network, the end-to-end nature of the Internet connectivity is guaranteed in the RSIP architecture. Most of the ALG in NAT implementation is no longer required.

It is possible for RSIP to allow for cascading of RSIP servers. For example, consider an ISP that uses RSIP for address sharing amongst its customers. It might assign only a private IP address to a particular customer. This customer may use RSIP again in his home network. No matter how many levels of RSIP, RSIP servers only assign public IP addresses to client. As shown in Figure 2.4, if RSIP client A requests an IP address from the nearest RSIP server C, C leases one public address 200.200.1.5 for host A from RSIP server Ds public address pool.

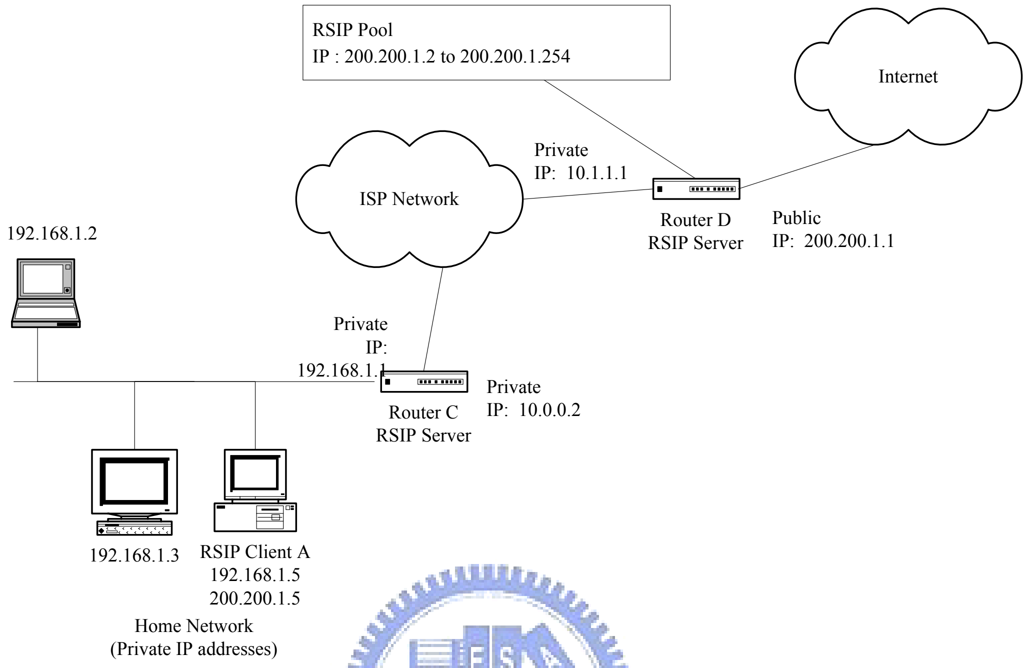


Figure 2.4: Cascaded RSIP Network

2.1.5 Problems with RSIP

RSIP provides a mechanism for end hosts to lease public IP addresses from RSIP server, which avoid the limitation of NAT. Hosts without RSIP client-enabled could still communicate with public network by using traditional NAT. However, RSIP ar-chitecture still has the following drawbacks:

First, the RSIP architecture does not concern about security issues to authorize or authenticate clients. The RSIP server cannot manage the resources efficiently, and it may meet the Denial of Service attack [11]. Second, The RSIP tunnel establishes be-tween client and server is not encrypted. Third, the RSIP server contains only one public addresses pool. Hosts in the public network cannot lease private IP address from the RSIP server in the reverse direction. For example, consider a mobile host that uses other ISP to access the Internet. This mobile host cannot use RSIP to grant access to his home network.

Furthermore, considering two customer-networks use cascade RSIP to connect to the same ISP, as shown in Figure 2.5. If RSIP client A that resides in home network H1 wants to communicate with RSIP client B that resides in home network H2, both of them should request public addresses from ISPs RSIP Server D. The communication link between RSIP client A and RSIP client B will be: client A → Router C → Router D → Router E → client B. Where the both the two pubic address and the two tunnels that setting up from Router C to Router D and Router E to Router D are unnecessary.

2.2 Multi-Homed networks

In a multi-homed network, a mobile host may have one or more network interfaces. The network interfaces may support the mobile host one or more public IP address or private IP address. If the IP address is a private one, an NAT box is resides in the network. Here we summarize five different situations of the network architecture:

Case 1 The mobile host has only one network interface and the edge router has n links. The host A connects to a single network realm X; the edge router of realm X has one or more path to link to Internet. The IP address assigned to host A is a public IP address.

Case 2 The mobile host A connects to a single network realm X; realm X has more

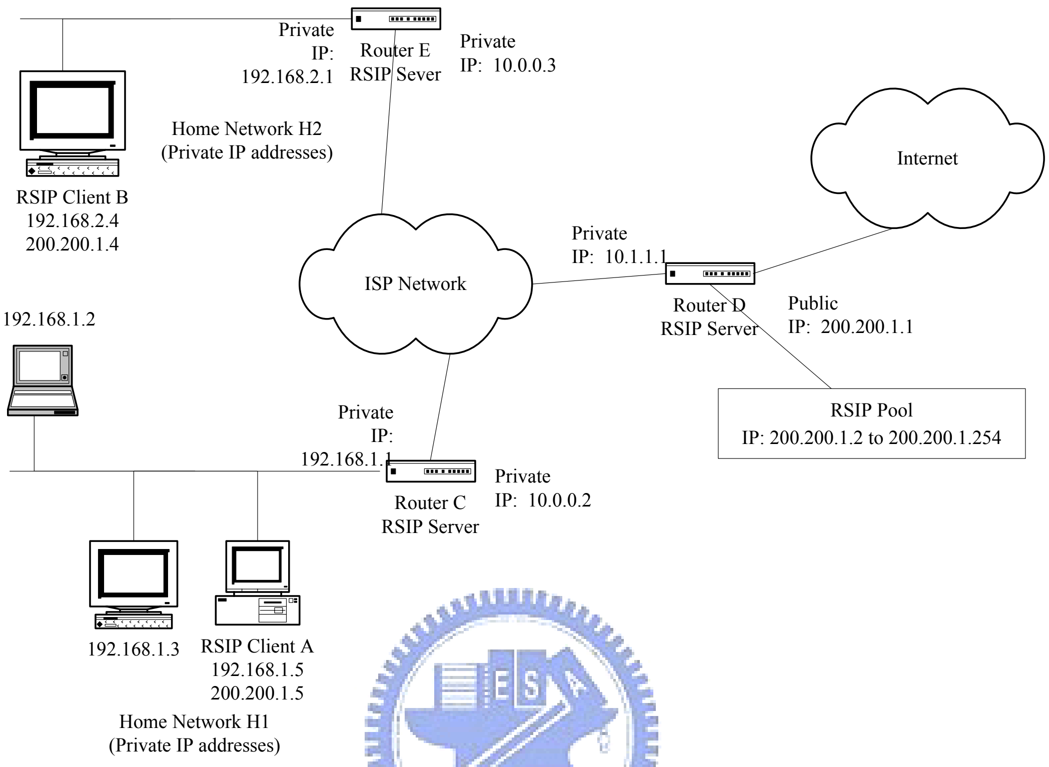


Figure 2.5: Redundant link in RSIP

than one edge router with several interface to link to Internet. The IP address assigned to host A is a private IP address. NAT function is performed in the edge router.

Case 3 The mobile host has only one network interface and the edge router has n links. The host A connects to a single network realm X; the edge router of realm X has one or more path to link to Internet, whereby one NAT box is resides in one link. The IP address assigned to host A is a private IP address.

Case 4 The mobile host A connects to several network realms; the IP addresses assigned to host A are all public IP addresses. Each realm may contain several outbound links.

Case 5 The mobile host A connects to several network realms, the IP addresses assigned to host A may be public IP address or private IP address.

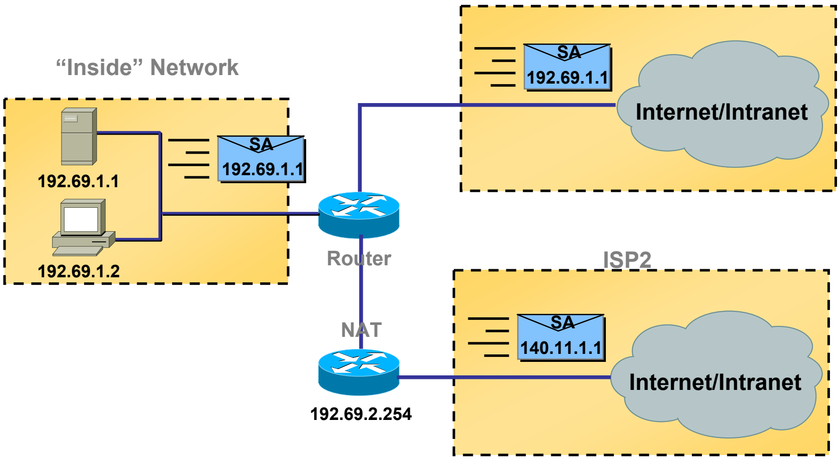


Figure 2.6: Multi-homed Network with NAT, case 1

To overcome the problems described above, the Internet Engineering Task Force (IETF) proposed two protocols to support terminal mobility among IP subnets, the Mobile IP protocol and the Stream Control Transmission Protocol (SCTP) [61]. However, both of them still contain several problems. In the mobile IP network, a mobile host sends a binding update message to perform a roaming operation when a mobile host migrates from one interface to another. If the home

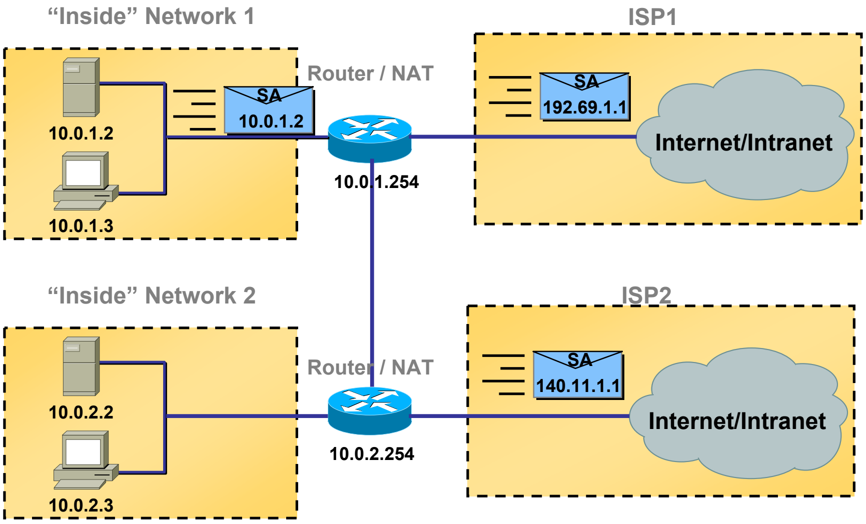


Figure 2.7: Multi-homed Network with NAT, case 2

agent (HA) is unreachable at this time, the foreign agent (FA) cannot process this location update request. Packets send to mobile node (MN) cannot be forwarded to the newest location and the connection will be terminated.

Another two solutions are based on DNS: Round robin DNS and Dynamic DNS. Round robin DNS is usually used for balancing the load of geographically distributed Web servers, but can be used in a multi-homing environment. Dynamic Domain Name System (DDNS) is a method of keeping a domain name linked to a changing IP address as not all computers use static IP addresses. An mobile host with DDNS supports will update it's current IP addresses with the DNS server, which means other users just need to use DNS query to find out the current location of the mobile node.

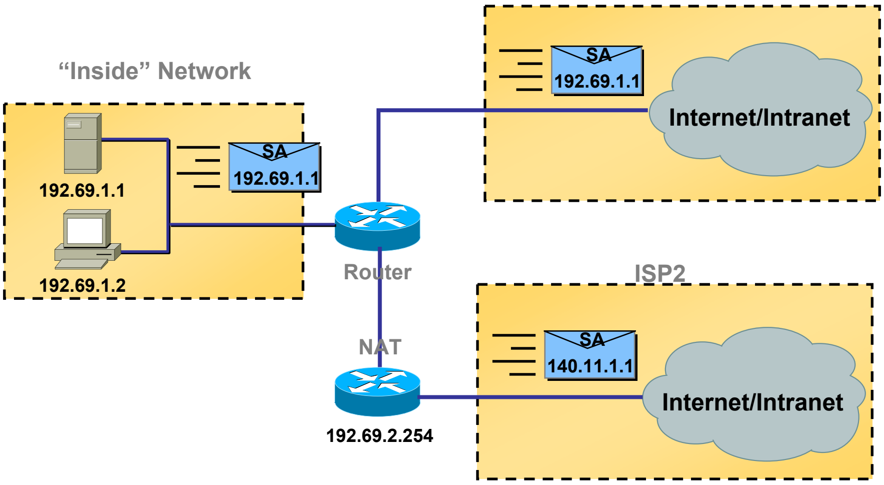


Figure 2.8: Multi-homed Network with NAT, case 3

2.2.1 Mobile IP and mobile IPv6

In Mobile IP network [45], a mobile node (MN) gets a Home Address from its home agent (HA). When a mobile node handoffs to a foreign network, it gets a Care-of Address (CoA) from foreign agent (FA) and informs home agent (HA) its care-of address by sending a registration request message to the home agent. The home agent maintains the binding between the care-of address and the home address of each mobile node. When a valid binding for a mobile node exists, the home agent will capture all the packets sent from correspondent nodes (CNs) to the mobile node's home address and forward them by tunneling to the care-of address. In MIPv6 [25], the mobile node uses can inform correspondent nodes about its current location by using a binding-update message; the correspondent nodes will be able to send packets directly to mobile node's care-of address, instead of sending packets through mobile node's home address.

2.2.2 SCTP protocol

The Stream Control Transmission Protocol (SCTP) [61] is an IP-based end-to-end, connection oriented transport protocol developed by the Internet Engineering Task Force (IETF) Signaling Transport working group for the transport of signaling data. However, SCTP is a general purpose transport protocol which provides numerous advantages over user datagram protocol (UDP) and transmission control protocol (TCP). For instance, SCTP combines the datagram orientation of UDP with the sequencing and reliability of TCP. Additionally, SCTP uses multi-stream, message-oriented routing in multi-homed environments. SCTP provides applications with enhanced performance, reliability, and control functions.

SCTP protocol overview

In SCTP, data is transmitted between endpoints through a connection referred to as an association. An association begins with an initiation of a four-way handshake between two endpoints and is maintained until all data has been successfully transmitted and received. Within SCTP, user data and control messages are assembled into chunks. An SCTP packet contains a common header and zero or more chunks.

SCTP message streams

The term ”stream” is used in SCTP to refer to a sequence of user messages that are to be delivered to the upper-layer protocol in order with respect to other messages within the same stream [61]. SCTP multi-streaming logically divides user data into unidirectional streams with each stream having its own delivery mechanism.

All streams within a single association share the same congestion and flow control parameters. Through multi-streaming, SCTP eliminates unnecessary blocking that often occurs in TCP transmission.

In TCP, user data is delivery in a single sequence of bytes which is strictly ordered. This delivery mode results in a major drawback known as ”head-of-the-line blocking (HOL),” where messages are not allowed to bypass each other. Multi-streaming decouples data delivery and transmission, and in doing so prevents Head-of-Line blocking.

SCTP streams are effectively unidirectional channels, within which messages are usually transported in sequence, unless the user requests a message to be delivered by an unordered service. The stream mechanism may reduce the effects of head-of-line blocking, especially in the case of a large number of small messages and a large number of stream. SCTP also provides a mechanism for unordered delivery service as UDP. User messages sent using this mechanism are delivered to the SCTP user as soon as they are received without any processing.

SCTP Multi-Homing

The SCTP supports multi-homed endpoints with more than one IP address. SCTP has a built in failure detection and recovery scheme, known as failover, which allows associations to dynamically send traffic to an alternative destination address when needed without losing the end-to-end association or requiring the application to intervene. This failover occurs after a threshold number of transmission timeouts to the primary destination address have occurred. SCTP also exploits this path redundancy in its retransmission policy.

2.2.3 IP round-robin and dynamic DNS

IP Round robin works on a rotating basis in that one server IP address is handed out, then moves to the back of the list; the next server IP address is handed out, and then it moves to the end of the list; and so on. Round robin DNS is usually used for balancing the load of geographically distributed Web servers. For example, a company has one domain name and three identical home pages residing on three servers with three different IP addresses. When one user accesses the home page it will be sent to the first IP address. The second user who accesses the home page will be sent to the next IP address, and the third user will be sent to the third IP address. In each case, once the IP address is given out, it goes to the end of the list. The fourth user, therefore, will be sent to the first IP address, and so forth. Although very easy to implement, round robin DNS has important drawbacks, such as those inherited from the DNS hierarchy itself and TTL times, which causes undesired address caching to be very difficult to manage. Moreover, its simplicity makes remote servers that go unpredictably down inconsistent in the DNS tables. However, this technique, together with other load balancing and clustering methods, can produce good solutions for some situations.

Dynamic Domain Name System (DDNS) [66] is a method of keeping a domain name linked to a changing IP address as not all computers use static IP addresses. Typically, when a user connects to the Internet, the user's ISP assigns an unused IP address from a pool of IP addresses, and this address is used only for the duration of that specific connection. This method of dynamically assigning addresses extends the usable pool of available IP addresses. A dynamic DNS service provider uses a special program that runs on the user's computer, contact-ing the DNS service each time the IP address provided by the ISP changes and subsequently updating the DNS database to reflect the change in IP address. In this way, even though a domain name's IP address will change often, other users do not have to know the changed IP address in order to connect with the other computer.

2.3 Authentication protocols to be used in multi-homed networks

Network users potentially need to access sensitive private data or digital sign transaction over the network, which means secure transmission of information over insecure communication channels is a major issue in current Internet. To build a secure communication environment in multi-homed networks, an authentication protocol is required. Authentication protocol is an important technique to verify the identities of the communication parties when they start a connection. This service is usually provided in combination with a key generation scheme between the parties.

In recent years, a variety of protocols for authentication and key agreement have been proposed and applied to many communication environments. There are two different approaches to used in designing the authentication protocols: centralized or de-centralized.

Most current authentication mechanisms are based on the centralized approach, such as the well-known Kerberos authentication protocol [58], which required a trusted third party (key information center) between the communication parties.

In the Internet-scale multi-homing environment, the trusted third party becomes a bottleneck because the server should have enough computation power, network bandwidth and data storage to deal with large amount of users. When it is unable to process requests or vulnerable to attacks, millions people may be influence. Furthermore, in a large-scaled multi-homed network, the mobile host may not reach the server all the time.

Another approach is decentralized schemes. In a de-centralized authentication protocol, there are no central servers to deal with authentication and encryption, only two peers are involved in the communication. A lot of research has been done and many algorithms have been proposed to make the decentralized authentication schemes more secure [13, 35, 9, 42, 48]. Two of the famous decentralized authentication schemes are the RSA scheme and the Diffie et al's public key distribution system [13, 35]. Both of them still need a server to keep user's public information, and the correctness and security of that information must be guaranteed.

In a multi-homed network, it is better to use the decentralized authentication schemes because the communicating peers may come from different networks, which means communication with a central server is difficult.

To avoid the central server bottleneck to be occurred in large-scale multi-homed networks, identity-based (ID-based) authentication schemes are developed [9, 42, 48]. In an ID-based authentication scheme, the public information of a user, such as name and address, are used as user's public key, which need not be stored in a central server. The first ID-based scheme proposed by Shamir [48], which rely on the existence of a trusted central authority, supports only digital signature rather than message encryption. In 1989, Okamoto and Tanaka extended Shamir's idea and combined digital signature and key distribution in a sample ID-based scheme, which supports message encryption and withstands the conspiracy problem [42]. Okamoto and Tanaka's scheme has the following problems: user identifications may be forged, user secret information may be disclosed, and the high overhead of exponential computations is needs. Tsujii proposed another ID-based cryptosystem, which suffers from the conspiracy problem, still needs high overhead of exponential computations.

2.4 Summary

In this chapter, we briefly introduce three major problems impacting the internet: the Network Address Translation (NAT), the multi-homing protocol such as Stream Control Transport Protocol (SCTP), and the authentication protocols to be used in the multi-homed networks.

NAT is considered a solution to the insufficiency of IPv4 address space. However, if IPv6 is widely deployed on the Internet, address space will no more be an issue and we can obviate the need of NAT. Regardless of address space consideration, NAT can still be considered a solution in IPv6 for security or load-sharing concerns.

In the near future, Internet service providers may only provide NAT solutions to small enterprise networks due to the shortage of address space. In such trends, existing Internet security protocols must be re-examined together with this new network environment.

The Stream Control Transmission Protocol is an IP-based end-to-end, connection oriented transport protocol which is a general purpose transport protocol which multi-stream, message-oriented routing in multi-homed environments.

SCTP provides applications with enhanced performance, reliability, and control functions.

Finally, we evaluate two different approaches to used in designing the authentication protocols: centralized or de-centralized. We found it is better to use the decentralized authentication schemes in the multi-homed networks.

Chapter 3

Multi-Layer RSIP Management Architecture

With the growth of virtual private networks, a user hides in a private network may wish to communicate with other users reside in another private network. Most of the private networks use IP ranges 10.0.0.0/24, 172.16.0.0/16, or 192.168.0.0/8 defined by IETF [46]. The address collision problem may occur if both sides use the same address space. Currently both NAT and RSIP architectures not only lack server connection ability, but also cannot solve the address-collision problem. Eun-Sang Lee et al. proposed an architecture that modified the mapping table inside NAT-server to provide server connection ability [33]. However, Lee's architecture does not solve the end-to-end security problem.

In this chapter, we propose a new architecture, the MRSIP framework, to solve the above problems. The specific goals of our framework are as follows.

Each of the network hierarchies described in the above chapter addresses some of the deficits of Internet scaling problems. However, as has been pointed out above, none of these architectures can be considered a completely satisfying solution yet. The aim of MRSIP is to define a clear, simple, and flexible architecture which integrates the advantages of each of the abovementioned approaches while avoiding their disadvantages, and which provides a solid bases for adding new features in a consistent and straight-forward manner. The basic design objectives of MRSIP can be summarized as follows:

- Transparent-access capabilities with the private network.
- Cascade private network architecture.
- Redundant path detection between source and destination nodes
- Reduce the address-collision probability

Our proposed framework is useful in real world implementations. Our framework is also very well suited to small private networks such as home networks and security devices like firewalls.

3.1 System components and terminology

In order to convert a standard RSIP network into an MRSIP network, it is at least necessary to insert an MRSIP agent into the framework, and to replace all RSIP server and RSIP client to MRSIP server and MRSIP client, respectively. In the following, we will motivate and explain the functionality of the MRSIP network infrastructure.

- MRSIP Gateway

  An MRSIP gateway is a router situated on the boundary between two address realms and owns one or more IP addresses in each realm that can be assigned to MRSIP clients. An MRSIP gateway contains two major components, the MRSIP server and the MRSIP agent.

- MRSIP Client

  An MRSIP client replaces the original RSIP client with several modifications: each time when MRSIP client initiates a new connection, it requests a new pair of IP/port resources from MRSIP gateway. When the connection is terminated, the MRSIP client returns those resources back to MRSIP gateway. An MRSIP client may lease many IP/port resources for several communications at the same time.

- MRSIP Agent

  An MRSIP agent provides the resource management, tunnel establishment, and redundant path detection. It manages resources that will lease to or return from MRSIP clients or other MRSIP gateways. The MRSIP agent contains two addresses pools, the inner address pool with private addresses and the outer address pool with public addresses. In a cascade MRSIP framework, the outer address pool also contains private address.

  An MRSIP agent may lease its address to another MRSIP gateway. In Figure 1, the MRSIP agent C contains two address pools, 10.10.40.1 to 10.10.40.254 and 192.168.1.128 to 192.168.1.253, whereby both of them are private addresses. But agent C can request public addresses (200.200.1.2 to 200.200.1.254) from its parent RSIP gateway D.

  The MRSIP framework is more portable compares to the original RSIP network by using the several address pools: hosts inside the private network can request the public IP address and hosts outside can get private addresses to access local servers.

- MRSIP Server

  The MRSIP server is responsible for tunnel establishment and data forwarding with MRSIP clients and other MRSIP gateways. Consider that an MRSIP client resides in a private network want to send data packets to the public network, the MRSIP client first register and requests a public IP from MRSIP gateway, establishes a tunnel with the correspondent MRSIP server, and encapsulates those data packets into that tunnel. The MRSIP server receives data in the tunnel, decapsulates those tunneled data packets, and sends them to the destination host.

  Consider another case that both the source and destination hosts reside in two MRSIP private networks. When the MRSIP client requests an IP from a local MRSIP gateway, the MRSIP gateway forwards the request to the destination network's MRSIP to get a private IP in the destination network. The MRSIP client establishes a tunnel to the local MRSIP gateway as mention above. Another tunnel is established between two correspondent MRSIP gateways. The MRSIP client uses the requested private IP and these two tunnels to reach the destination host.

3.1.1 Registration of MRSIP clients

When an MRSIP client startup, it first determines where is the location of the local MRSIP gateway, and sends a registration request to the MRSIP gateway. The MRSIP gateway checks the registration request and authenticates the client's identity. After that, it generates a client-ticket, inserts client's information and this client-ticket to a host table, and returns this ticket to the client. The MRSIP client uses this ticket to do addressing binding before establishing a connection outside.

3.1.2 De-registration of MRSIP clients

When an MRSIP client determines it does not need RSIP service anymore, it sends a de-registration request with its client-ticket to the local MRSIP gateway. The MRSIP gateway checks the client-ticket and removes the client's entry in its host table. If a specific interactive period timeout reached after the MRSIP client registered itself to the local MRSIP gateway, the MRSIP gateway deregister this specific client and removes the client's entry automatically.

3.1.3 Address binding

As far as described in this paper, when the RSIP client requests an IP from the local RSIP gateway, the RSIP servers always returns a public IP to the client in the original RSIP framework. By the way, the original RSIP client cannot communicate with hosts resides in another private network, except the remote host is also an RSIP client that binds a public address already. To solve this problem, our modified MRSIP client requests IP in the destination realm. The link properties can be summarized as follows:

1. Both the source host and the destination host are all in the public network.

2. The source host resides in a private network, but the destination host is in public network.

3. The source host is in public network but the destination host resides in a private network.

4. Both the source host and the destination host reside in private networks interconnected by the public network.

5. Both the source host and the destination host reside in private networks interconnected by a private network.

In case 1, the source host communicates with destination host by public IP directly. In case 2, the source host uses MRSIP client to request a public IP address from local MRSIP gateway and establishes a tunnel with MRSIP server.

In case 3, a public IP address should be previously assigned to the destination host. The source host should have the ability to query destination host's IP address by dynamic domain name system or other service allocation protocols.

Now considering case 4, there are two approaches that can solve this problem. First, both source and destination hosts requests public IP address from MRSIP gateway to communicate with each other. This is the traditional RSIP strategy. The second approach required the source host requests a private address from the MRSIP gateway resides in the destination network. The source host establishes tunnels from itself to the destination's MRSIP server and uses that private IP address to communicate with the destination host. In the second approach, the destination host is no longer required to get a public IP previously.

In case 5, there are two approaches similar to the case 4. First, both source and destination hosts requests public IP address from MRSIP gateway to communicate with each other. All the data packets transmit from the sender are encapsulated in a tunnel routed to the public networks, and routed back to another encapsulated tunnel to reach the destination. Figure 1 shows the example. Assume host A and B gets the public addresses 200.200.1.5 and 200.200.1.3 from RSIP gateway D, respectively.

This approach requires two unnecessary tunnels (Host A → Gateway C → Gateway D and Host B → Gateway E → Gateway D) and two public IP address (200.200.1.3 and 200.200.1.5). The public IP addresses are expansive resources to those large networks that only have a little range of public IP.

The second approach required the source host requests a private address from the MRSIP gateway resides in the destination network. The MRSIP gateways in both sides negotiate a shortest path between them, and establish a server-to-server tunnel within the path. The source host establishes a tunnel from itself to the local MRSIP gateway and uses that private IP address to communicate with the destination host. In the second approach, the data packets will not route to public network and no public IP address is required.

As shown in Figure 3.1, router C establishes a client-to-server tunnel from A to C and a server-to-server tunnel from gateway C to gateway E. Router C gives client A one private address 10.10.40.5, instead of the 200.200.1.5. Client A uses the leased private address and these two tunnels to communicate with client B

3.1.4 Address unbinding

The address unbinding procedure is happened when an MRSIP client returns the early requested session-ticket to MRSIP gateway when the correspondent communication is terminated. The MRSIP gateway drops all the tunnels between the source and the destination host correspondent to the specific session-ticket. The

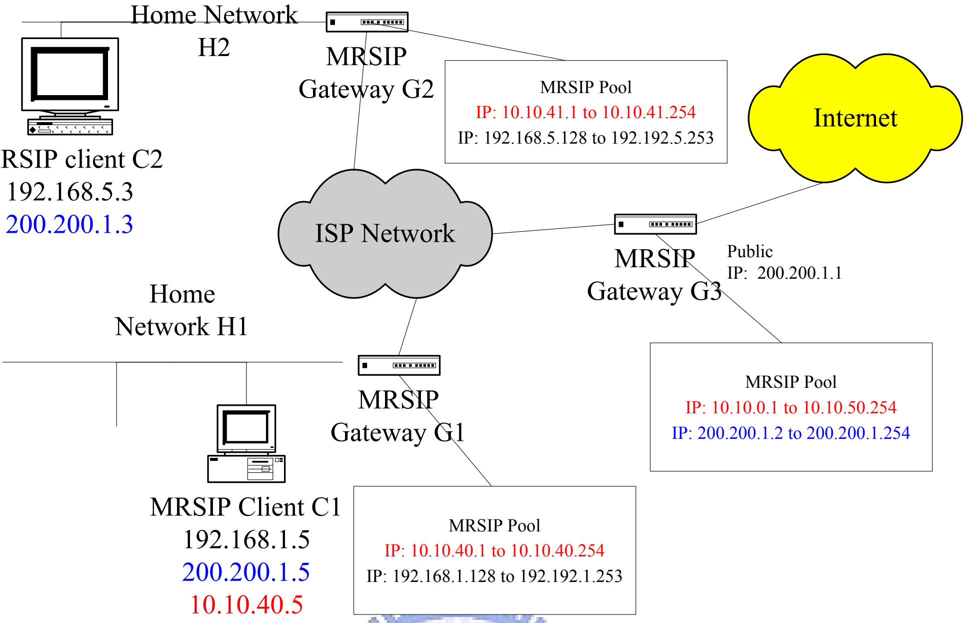


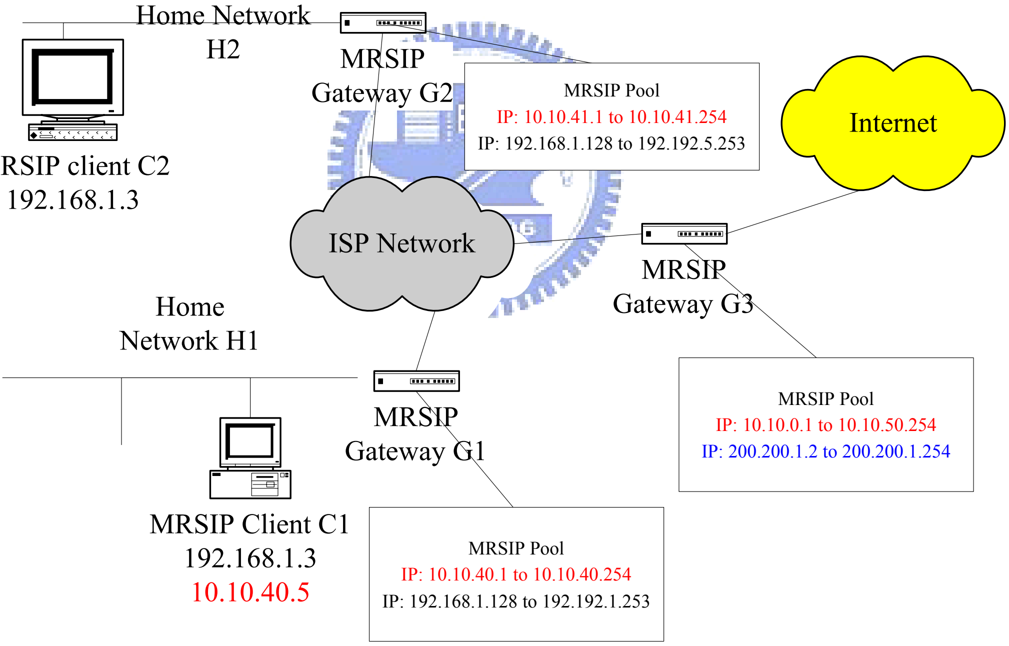
Figure 3.1: Address binding in MRSIP framework

released resources are put back to the MRSIP gateway's resource pool that can be use for future requests.

3.1.5 Address collision avoidance

Assigning private address of the destination network to the source MRSIP client reduces the necessary of public IP address, but induces the probability of address collision. Most of the private networks use IP ranges 10.0.0.0/24, 172.16.0.0/16, or 192.168.0.0/8 defined in [46]. Considering Figure 3.2 and 3.3, if the RSIP client A with private address 192.168.1.5 wants to connect to host B and host F, whereby the addresses are all 192.168.1.3

To avoid the occurrence of address collision, when the MRSIP gateways detect an address collision during the address-binding step, the MRSIP gateways negotiate with other MRSIP gateways to replace the IP address one-side or both-side to prevent the collision. For example, consider when the MRSIP gateway C in Figure 3.2 receives a connection request to Host B from MRSIP client A, gateway C first discover host B is inside its neighbor MRSIP network. It attempts to request a private address from MRSIP gateway E, assume it is 192.168.1.6. Gateway C then detects there is an address collision and then returns an alternate IP address 10.10.40.5 to host A from its own address pool to avoid the address collision.



Home Network
H2
MRSIP
Gateway G2
MRSIP Pool
IP: 10.10.41.1 to 10.10.41.254
IP: 192.168.1.128 to 192.192.5.253
Internet
MRSIP client C2
192.168.1.3
ISP Network
MRSIP
Gateway G3
Home
Network H1
MRSIP
Gateway G1
MRSIP Pool
IP: 10.10.0.1 to 10.10.50.254
IP: 200.200.1.2 to 200.200.1.254
MRSIP Client C1
192.168.1.3
10.10.40.5
MRSIP Pool
IP: 10.10.40.1 to 10.10.40.254
IP: 192.168.1.128 to 192.192.1.253

Figure 3.2: The address collision of two private networks (I)

In figure 3.3, when gateway C receives a connection request to Host B from MRSIP client A, gateway C first discover host B is resided in another MRSIP network partition by a public network. It attempts to request a private address from MRSIP gateway E, assume it is 192.168.1.6. Gateway C then detects there is an address collision and then returns an alternate public IP address 200.200.1.5 to host A from its parent RSIP gateway's address pool to avoid the address collision.

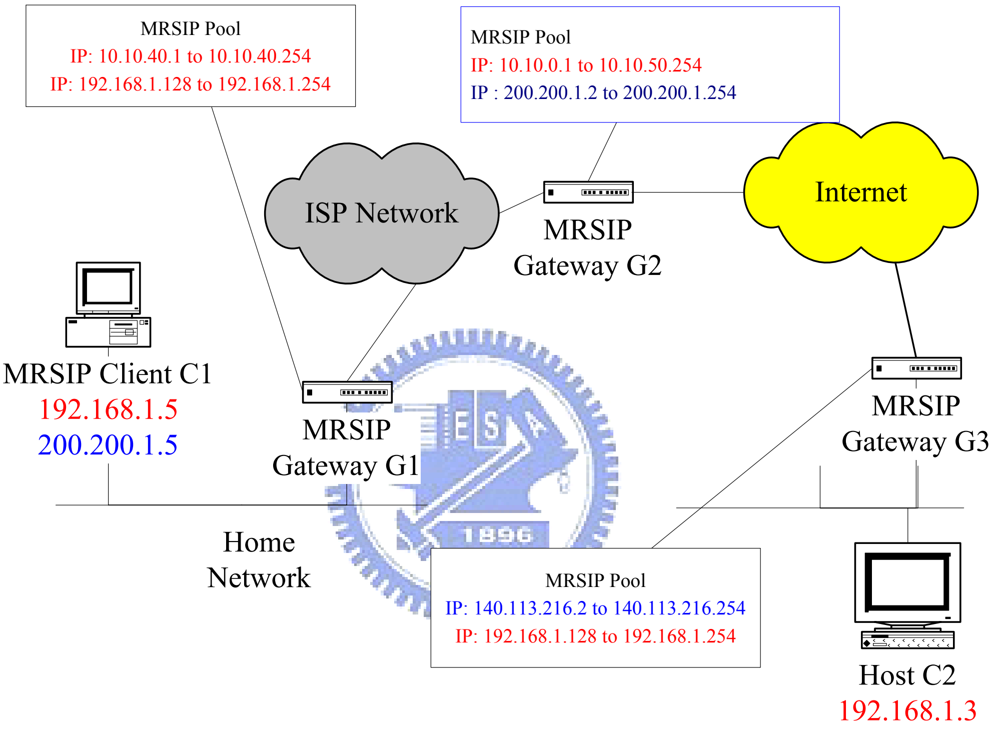

MRSIP Pool
IP: 10.10.40.1 to 10.10.40.254
IP: 192.168.1.128 to 192.168.1.254
MRSIP Pool
IP: 10.10.0.1 to 10.10.50.254
IP : 200.200.1.2 to 200.200.1.254
ISP Network
MRSIP
Gateway G2
Internet
MRSIP Client C1
192.168.1.5
200.200.1.5
MRSIP
Gateway G1
MRSIP
Gateway G3
Home
Network
MRSIP Pool
IP: 140.113.216.2 to 140.113.216.254
IP: 192.168.1.128 to 192.168.1.254
Host C2
192.168.1.3

Figure 3.3: The address collision of two private networks (II) caption

3.1.6 Security aspects

The original RSIP framework does not discuss about the security issue. The RSIP server does not identify its clients and assumes they should use IPsec or other encryption protocols to protect the packets. Also, the original RSIP client cannot access a server resides in a private network if the server does not have a public IP address. Furthermore, the RSIP server cannot resist from denial of service attack if an attacker get all the public addresses from server's IP pool.

In our proposed MRSIP framework, both MRSIP client and MRSIP server should prove there identify to the MRSIP gateway. The MRSIP gateway only provides services for trust clients and servers. An MRSIP gateway controls whether a client can get a private address to access a specific private server. Even the MRSIP gateway inhibits a client to use private address to access the specific server directly; the client may try to use public IP address to access the server.

3.2 Protocol specification

In this section, we define the parameters and the control message types that uses in our MRSIP framework. We provide a series protocol examples in section 3.3 to demonstrate how the MRSIP works.

3.2.1 Parameter specification and formats

In the original RSIP protocol specification [3] describes the parameters and control messages. Our MRSIP framework extends the specification to provide the ability of authenticate clients and gateways. The extended parameters are described as follows:

- Client-ID:

  A client-ID specifies an MRSIP client's identity. The client-ID data structure contains a unique 32-bit integers and a string that specifies client's information. The string can be a private IP address, user's e-mail address, or a certificate issued by a particular certificate-authority.

- Gateway-ID:

  A Gateway-ID is a string that specifies an MRSIP gateway's identity. The Gateway-ID data structure contains a unique 32-bit integer and a string that specifies an MRSIP gateway's identity information.

- Session-ID:

  A Session-ID is a unique 32-bit integer that used by MRSIP clients and gateways to differentiate an MRSIP client's bindings.

- Signature:

  The signature is appended in the rest of each request or response message to authenticate the message.

- Client-Ticket:

  A client-ticket is issued by a particular MRSIP gateway for a specific client after the registration procedure is succeeded. The client-to-gateway tunnel information and other server information are stored in the client-ticket.

- Session-Ticket:

  A session-ticket is issued by a particular MRSIP gateway for a specific client after the address binding procedure is succeeded. The session-ticket contains the assigned IP resources and other parameters given by the gateway.

- Gateway-Ticket:

  A gateway-ticket is issued by a particular MRSIP gateway for the other specific MRSIP gateway after the registration procedure is succeeded. The gateway-ticket contains the MRSIP gateway information.

- Tunnel-Ticket:

  A tunnel-ticket is issued by a particular MRSIP gateway for the other specific MRSIP gateway after the tunnel-binding procedure is succeeded. The tunnel-ticket contains the gateway-to-gateway tunnel information, and the IP addresses that can be provided by the remote gateways.

3.2.2 Control message types

In this section we describe the control message types that is used in our MRSIP protocol. The MRSIP control messages are based on the ”request-response” model. These control messages contains the register procedures, de-register procedures, tunnel-establishment procedures, address-binding procedures, and host query procedures.

- Registration request and response:

  An MRSIP client sends a registration request to its home MRSIP gateway to register itself before requests any resources. An MRSIP gateway should register itself to neighbor gateway before requests any resources or establish tunnels. Both MRSIP client and gateway should not register more than once before it has de-registered. An MRSIP client or gateway should provide his Client-ID or Gateway-ID and signature to the specific gateway, respectively. The registration response message is used by an MRSIP gateway to confirm the registration of an MRSIP client or the other MRSIP gateway. A Client-Ticket or a Gateway-Ticket is returned for future operations.

- De-registration request and response:

  An MRSIP client or gateway de-registers itself to an MRSIP gateway when the connection is no longer required. If an MRSIP client de-registers itself, all of the client's address-bindings are revoked. If an MRSIP gateway de-registers itself to the other MRSIP gateway, all of the address binding and tunnels are revoked. The de-registration response message is used by an MRSIP gateway to confirm the request.

- Tunnel-binding request and response:

  The tunnel-binding request and response messages are used by an MRSIP gateway to establish a gate-way-to-gateway tunnel with the other MRSIP gateway. An MRSIP gateway should register itself to the specific MRSIP gateway to get one Gateway Ticket before establishing a tunnel between them.

- Free-tunnel request and response:

  The free-tunnel request and response are used by an MRSIP gateway to free a tunnel. A tunnel is freed when all the address binding inside the tunnel are all freed.

- Address-query request and response:

  An MRSIP client or an MRSIP gateway uses the address-query request message to ask an MRSIP gateway whether or not a particular address or network is local or remote. The MRSIP client uses this information to determine whether to contact the host directly or via MRSIP gateway. When an MRSIP gateway receives the query-request message, the gateway performs the following procedures if the queried address is not access directly by itself: first, it forwards the query request message to its neighbor MRSIP gateways and wait for response. Second, a tunnel-binding request message will be sent to a specific MRSIP gateway to establish a tunnel between them. Finally, it returns a response message to the client or gateway that sent the query message.

- Address-binding request and response:

  An MRSIP client sends the address-binding request message to its home MRSIP gateway to bind an outside IP address. If the MRSIP gateway cannot allocate the resource requested by the client, it forwards the request to his neighbor gateway. A Session-Ticket is returned to the client and a client-to-gateway tunnel is established between the MRSIP client and its home MRSIP gateway.

- Free-Binding request and response:

  When an address binding is no longer required by an MRSIP client, it sends the free-binding request message with a Session-Ticket to the MRSIP gateway. MRSIP gateway frees the specific resources. If the resource is not own by the gateway, the gateway forwards the request to other MRSIP gateways. All the unused tunnels between client and gateway will be released.

3.3 Two protocol examples

In this section we describe two protocol examples of the MRSIP framework. An MRSIP client is denote by Cn, and an MRSIP gateway is denote by Gn, where n is a number to identify each entity. All MRSIP client-to-gateway traffic, gateway-to-client traffic and gateway-to-gateway traffic is denote by ’Cn→ Gn’, ’Gn→ Cn’, and ’Gn→ Gn’, respectively.

1. Client communicates with host resides in public network

   $C_1 \rightarrow G_1$: REGISTER_REQUEST

   $G_1 \rightarrow C_1$: REGISTER_RESPONSE

   The MRSIP client attempts to register with the gateway, the gateway responds and assigning a client-ticket to the client.

   $C_1 \rightarrow G_1$: QUERY_REQUEST

   $G_1 \rightarrow C_1$: QUERY_RESPONSE

   When the client C_1 attempts to connect to other host C_2, C_1 sends a query message to G_1 to retrieve C_2 ’s address information. G_1 responds if C_2 is in the foreign network or not.

   $C_1 \rightarrow G_1$: ADDRESS-BINDING_REQUEST

   $G_1 \rightarrow C_1$: ADDRESS-BINDING_RESPONSE

   C_1 determines that C_2 is located in the public network; C_1 attempts to request a public IP from the gateway G_1 and establishes a tunnel between itself and the gateway.

$C_1 \rightarrow G_1 \rightarrow C_2$: Data-Packets

C_1 uses the tunnel to communicate with C_2.

$C_1 \rightarrow G_1$: FREE-BINDING_REQUEST

$G_1 \rightarrow C_1$: FREE-BINDING_RESPONSE

C_1 ends the connection with C_2 and releases the binding IP address to the gateway.

$C_1 \rightarrow G_1$: DE-REGISTER_REQUEST

$G_1 \rightarrow C_1$: DE-REGISTER_RESPONSE

C_1 de-registers itself with the gateway.

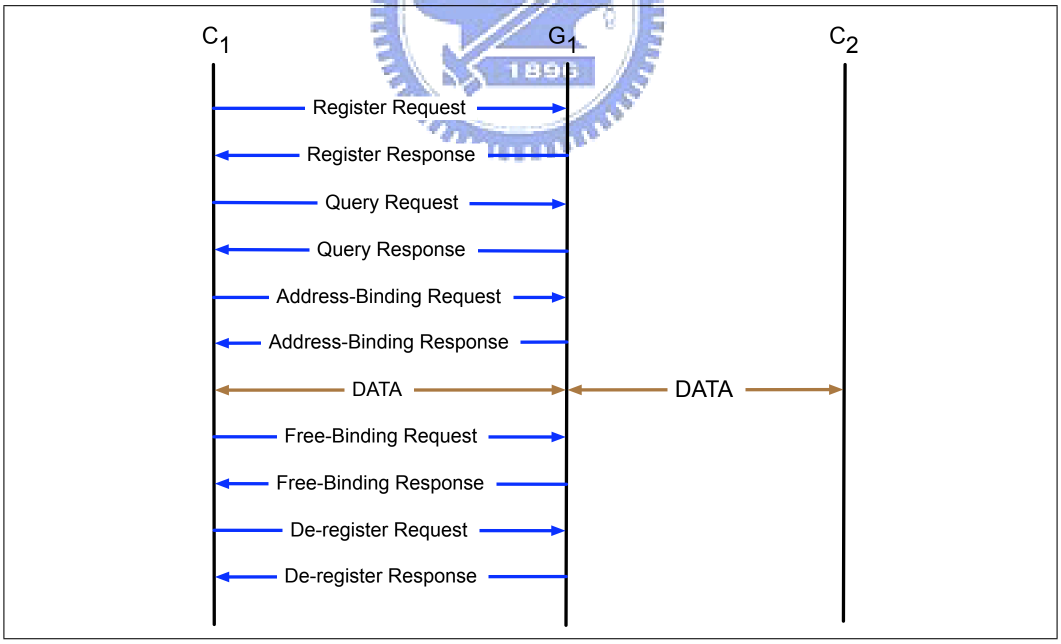


Figure 3.4: Client communicates with host resides in public network

2. Client communicates with hosts resides in another private network that partition with the public network between them

   When the client C_1 attempts to connect to other host C_2, C_1 registers with it's nearest gateway G_1 and sends a query message to G_1 to retrieve C_2's address information.

   $G_1 \rightarrow G_2$: REGISTER_REQUEST

   $G_2 \rightarrow G_1$: REGISTER_RESPONSE

   $G_1 \rightarrow G_2$: QUERY_REQUEST

   G_1 recognizes that C_2 is resides in another private network that partitions with the public network. G_1 tries to register itself to the remote gateway G_2 to retrieve the C_2's information and send the result to C_1. Fr

   $G_2 \rightarrow G_1$: QUERY_RESPONSE

   $G_1 \rightarrow C_1$: QUERY_RESPONSE

   $C_1 \rightarrow G_1$: ADDRESS-BINDING_REQUEST

   C_1 determines that C_2 is located in the foreign network; C_1 attempts to request a public IP from the gateway G_1 and establishes a tunnel between itself and the gateway.

   $G_1 \rightarrow G_2$: TUNNEL-BINDING_REQUEST

   $G_2 \rightarrow G_1$: TUNNEL-BINDING_RESPONSE

   $G_1 \rightarrow C_1$: ADDRESS-BINDING_RESPONSE

The gateway G_1 forwards the address-binding request to G_2 and return the result to C_1. A gateway-to-gateway tunnel from G_1 to G_2 is established between the address request intervals.

$C_1 \rightarrow G_1 \rightarrow G_2 \rightarrow C_2$: Data-Packets

C_1 uses the tunnel to communicate with C_2.

$C_1 \rightarrow G_1$: FREE-BINDING_REQUEST

C_1 ends the connection with C_2 and releases the bound IP address to G_1.

$G_1 \rightarrow G_2$: FREE-TUNNEL_REQUEST

$G_2 \rightarrow G_1$: FREE-TUNNEL_RESPONSE

$G_1 \rightarrow C_1$: FREE-BINDING_RESPONSE

G_1 destroys the tunnel between G_1 and G_2.

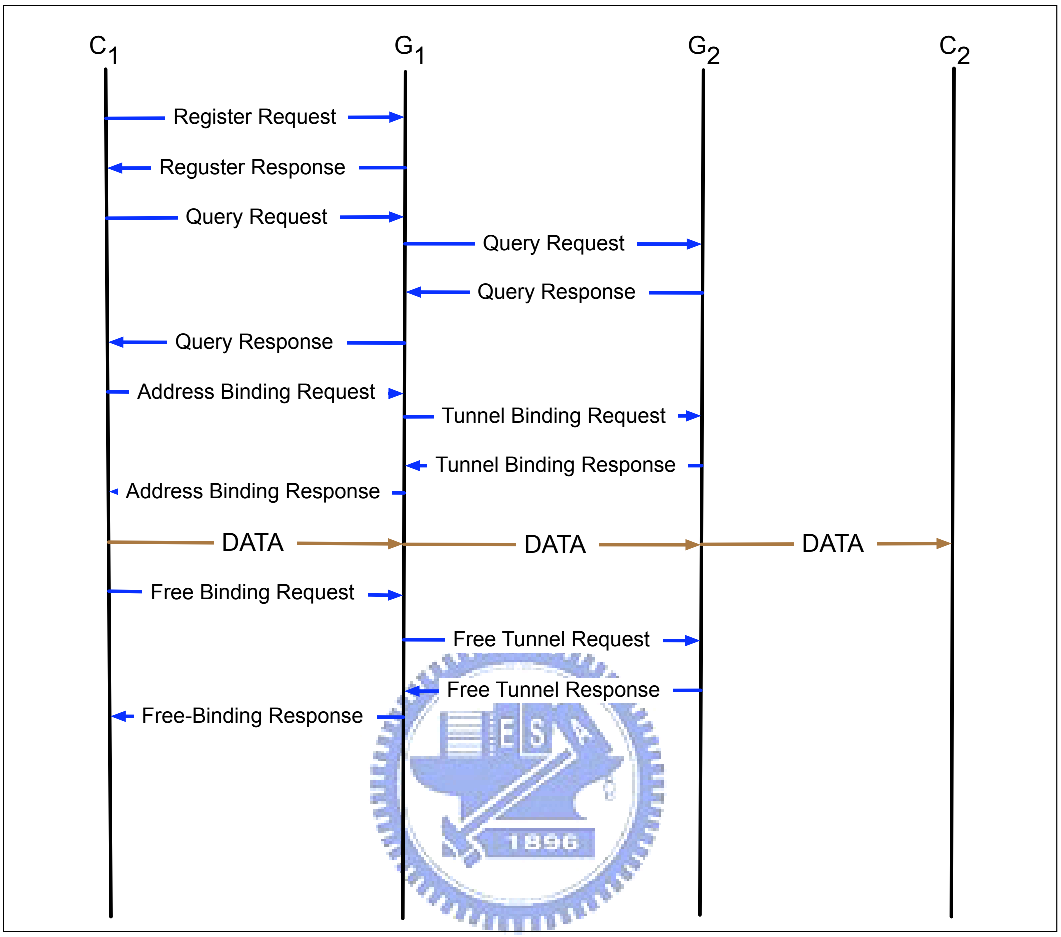


Figure 3.5: Client communicates with hosts resides in another private network

3.4 Comparison

In this section, we compare the functionality of NAT, RSIP and MRSIP, as shown in table 3.1. According to table 3.1, the NAT has only limited function to connect to or from public network (an application level gateway maybe required). Both the RSIP and MRSIP network, which can get one IP address from the gateway, can do bidirectional communication with one-level public network.

If the host is resides in more than one level private network, both NAT and RSIP client cannot be reached by public networks, but MRSIP client can get a public IP address from the top level gateway and establish a tunnel between the client and gateway.

If both communication party are resides in a second-level private network, whereby the top-level private network is providing by two different Internet service provider, the NAT and RSIP client are all unreachable.

Furthermore, since both NAT and RSIP cannot work on the multi-level private networks, only the MRSIP can do redundant path detection between source and destinations nodes, and reduces the address-collision probability.

Table 3.1: Comparison of NAT, RSIP and MRSIP

Functionality	NAT	RSIP	MRSIP
Connect from one-level private network to public network	Partial	**Yes**	**Yes**
Transparent-access capabilities from public network to the private network	Partial	**Yes**	**Yes**
Cascade private network architecture	No	No	**Yes**
Redundant path detection between source and destination nodes	No	No	**Yes**
Reduce the address-collision probability	No	No	**Yes**

3.5 Summary

This chapter describes the design of MRSIP framework, a transparency routing architecture for multi-level private networks. The MRSIP framework is proposed to replace the original NAT and RSIP network architecture. Many aspects of

MRSIP framework are inherited from original RSIP architecture, which provide the end-to-end connection nature. The idea of introducing an MRSIP gateway as a resource management controller is inherited from the application-level-gateway concept of NAT.

The MRSIP framework introduced multi-level private network architecture. The concept of using multiple address pools reduces the necessary of public addresses. The address binding procedure finds a short routing path between source and destination hosts, which reduces the unnecessary tunnels. Two private networks with the same address scope may communicate with each other by using the address collision avoidance procedure. An MRSIP client may request several IP addresses to access hosts in different address realms.

Obviously a lot of work remains to be done to refine the architectures proposed in this paper. Topics for further study include the refinement and prototyping of the concepts and technologies for MRSIP framework. Another interesting and important topic for the near future is the service allocation problem, which lookup services resides in private networks. We will integrate MRSIP framework with directory services to provide efficiency address binding and optimal routing.

Chapter 4

Multi-homed Architecture in Mobile Networks

Currently mobile devices are often equipped with more than one network interfaces, such as GSM/GPRS, 802.11 b/g wireless LAN, Bluetooth, IrDA or serial lines. For example, a mobile handset will have at least one interface GSM/GPRS, but some of them have bluetooth or IrDA interface. Most of the PDA devices have IrDA and USB serial connection with host computers, but some of them are equipped with 802.11 b/g wireless interface.

Due to the network coverage problem, a mobile device may not use all the interface to communicate with others all the time.

Multi-homing network contains several problems. First, to ensure that packets can be received over several paths, the enterprise's edge-router must advertise a set of addresses that may fall outside the aggregated address range supported by the correspondent ISPs. Each ISP's router must advertise their own aggregated address space and subscriber-specific addresses to other routers, which means the routing table entries will grow exponentially. Even if the routing protocol such as OSPF supports multi-path routing, it still needs time to construct a new routing path during network failure. During this re-convergence period, network traffic will be dropped within the network. Second, changing of an access network behind an interface may result in a situation where already established connections should be moved from one interface to another. If the access networks do not agree a same handover mechanism, the established connection will be terminated since the mobile device cannot migrate between those heterogeneous networks.

To overcome the above problems, we propose two path selection algorithms and two failover algorithms for SCTP to choose a new path by application type and network condition. This schemes reduces the failover cost.

In this section, we first describe the terminology that to be used in the following algorithms, as shown in table 4.1. We assume each end-host has more than one interface, which is addressing by i. The round trip time RTT_i, the retransmission counter when a failure occurs $Retrans_i$, the maximum retransmission count before mark one link as failure $Retrans_{MAX}$, and the congestion window size $Cwnd$ are defined in the original SCTP protocol.

4.1 Problems in SCTP protocol

In the SCTP, endpoints exchange a list of addresses during connection establishment. One of these addresses is called *primary address* and the others are *alternative addresses*. SCTP sends packets to the primary address, but use the alternative address to send non-delivery packets to improve the probability of reaching the remote endpoint. When transmission to primary address fails, packets are transmit-

Table 4.1: Terminology

i	Network interface
C	Network class
RTT_i	Round trip time of interface i
RTO_i	Re-transmission timeout of interface i
RTT_{MAX}	MAXimim value of round trip time
$Retrans_i$	The retransmission counter of each path when a failure occurs
$Retrans_{MAX}$	MAX retransmission count before mark one link as failure
$tCounter$	A threshold counter to determinate a path is temporary disabled
R_i	Link speed
$Cost_i$	Link cost
MSS	Segment size
$Cwnd$	Congestion window size
$Cwnd_{th}$	The threshold value of congestion window size
HB	Heartbeat message
HB_{ACK}	Acknowledgment of a heartbeat message

ted to one of the other alternative addresses until the protocol becomes confirmed that primary address is reachable. Unlike Mobile IP, no home agent and foreign agent are required in the SCTP protocol during the re-convergence period.

However, current SCTP protocol has several drawbacks. First, it can only send data packets to one primary address, not all the available paths.

Second, SCTP will only switch to an alternative address during network failure after quite a long time. If the failover time is too long, the session will be terminated before failover.

4.1.1 Path-selection and multi-homing

The SCTP protocol supports multi-homing. Therefore, each SCTP endpoint could have multiple IP addresses. During the protocol initiation state, SCTP endpoints exchange a static set of IP addresses from the available address pool. Each SCTP endpoint selects one of these addresses as the *primary address* and the others are *alternative addresses*.

After the protocol initiation state, the SCTP endpoint sends all data packets to the primary address, but STCP can reroute data packets to one alternate address to improve the probability of reaching the remote endpoint. When transmission to primary address fails, packets are transmitted to other alternative addresses until the protocol becomes confirmed that primary address is reachable by using extra heartbeat messages.

In SCTP, the policy for selecting the new destination address for sending data is undefined. The endpoint should monitor the reachability of these alternative addresses by regularly sending the heartbeat message to all alternative addresses. Upon reception of a hearbeat message (HB), the SCTP endpoint should reply with a *heartbeat acknowledgment* message (HB_{ACK}, whereby each peer always knows which addresses are available for the failover.

For each path, the endpoint keeps an error counter that is incremented, should the endpoint not receive an acknowledgement within a certain time. If the error counter exceeds a threshold value (a maxretransmition counter), the state of the considered path will be set to unreachable. Even if the path is set to unreachable, the endpoint will then continue to send heartbeats to this address, allowing the reinstatement of the path status to reachable at a later stage.

As described in RFC2960, the set of IP addresses is fixed during the initiation of an association and cannot be changed during the lifetime of an association. R. Stewart et al. extended the SCTP to support dynamic address reconfiguration [60] which means that each endpoint's available IP addresses can be dynamic update during the association lifetime by sending one Address Configuration Change (ASCONF) control chunk to its peer.

4.1.2 SCTP failover control

The SCTP supports multi-homed endpoints with more than one IP address. SCTP has a built in failure detection and recovery scheme, known as failover, which allows associations to dynamically send traffic to an alternative address when needed without losing the end-to-end association or requiring the application to intervene. This failover occurs after a threshold number of consecutive timeouts to the primary destination address have occurred, as shown in Figure 4.1 [26]. SCTP also exploits this path redundancy in its retransmission policy.

In figure 4.1, the packet number 1 to 8 will be sent to a destination interface B_1 and wait for the acknowledgment packet to be arrived. If endpoint A_1 does not receive the acknowledgment after a period of time RTO_1, these packets will be sent to an alternative interface B_2. The congestion window (Cwnd) C_1 will be reduce to 1, and the RTO will be doubled until a maximum value is reached (RTO_{MAX}). That is, $RTO_2 = min(2 * RTO_1, RTO_{MAX}$. After sending out packets 1 to 8, the packet 9 will still be sent to the original primary interface B_1. The congestion window C_1 will be increased by 1 if the selective-ACK of packet 9 is received.

In figure 4.2 [26], if the acknowledgment of packet 9 does not received from B_1 after a period of time $2 * RTO_1$, the packet 9 will be sent to the alternative interface B_2 and the RTO value will be increased to $4 * RTO_1$. The packet 10 will wait for $4 * RTO_1$ after timeout, and vice versa.

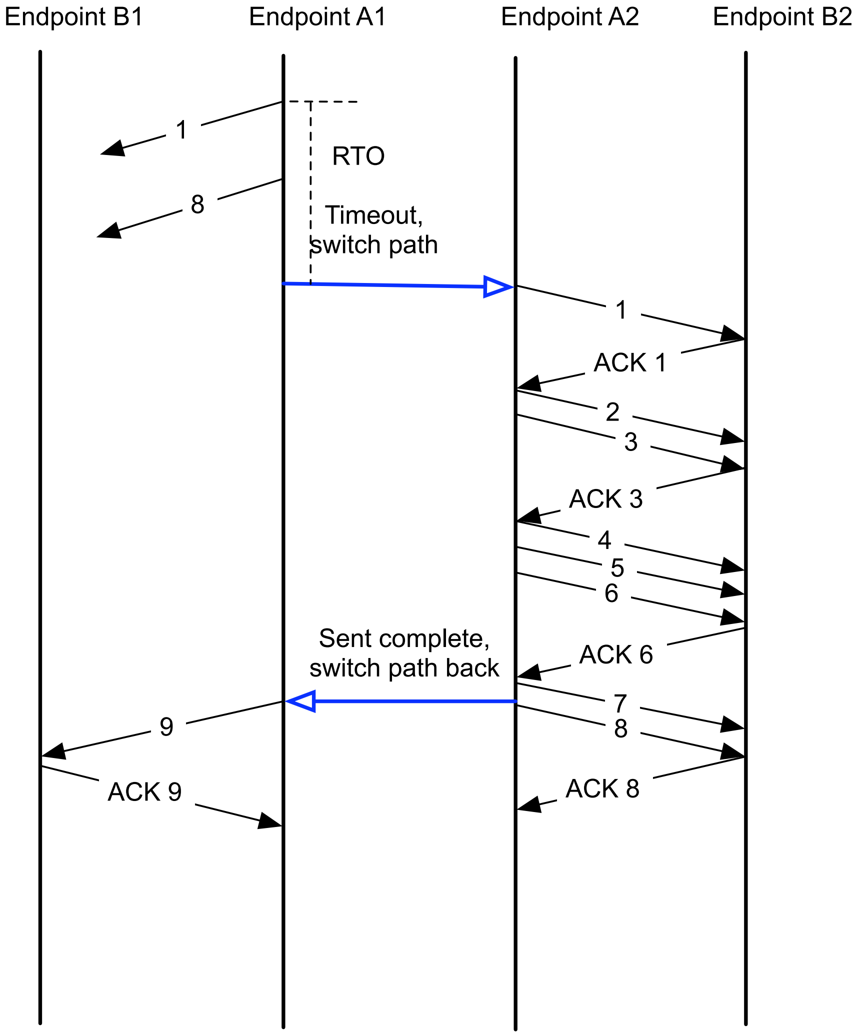


Figure 4.1: SCTP retransmit state diagram with congestion

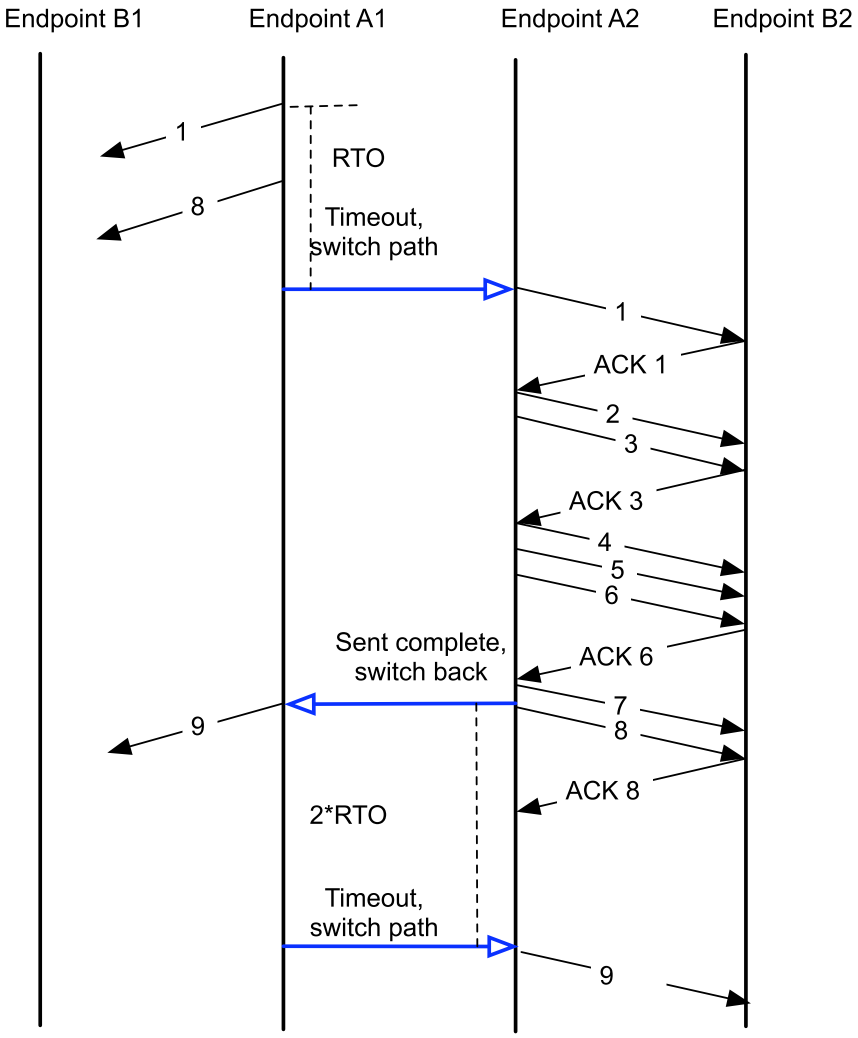


Figure 4.2: SCTP retransmit state diagram with failure

A $Retrans_i$ is used to counter the errors of each path. If the $Retrans_i$ reaches $Retrans_{MAX}$, SCTP will mark the destination address B_1 as failed and the primary destination address is changed to B_2.

The original SCTP packet-transferring algorithm is shown in Algorithm 1.

Algorithm 1 Original SCTP Packet-Transferring

```
Send packets to $PrimaryPath$;
Start timer;
if $timer$ reaches $RTO_i$ then
    Send lost packets to $AlternativePath$;
    $Retrans_i$ ++;
    $RTO_i$ = $RTO_i$ * 2;
    if $RTO_i > RTO_{MAX}$ then
        $RTO_i = RTO_{MAX}$;
    end if
    if $Retrans_i >= Retrans_{MAX}$ then
        Change primary destination path;
    end if
end if
```

4.2 Proposed enhanced SCTP protocol

To make a more flexibility decision on path selection, we should classified all the available interfaces into several *classes*. We define each class by the service type, the link cost, and the reachable destination networks. Table 4.2 is an example.

In table 4.2, the interfaces LAN_0, LAN_1, $WLAN_0$ and $WLAN_1$ allow all connections from all applications to all destination networks. Since the network speed of LAN_0 and $WLAN_0$ aenr not the same, they are in different service class. The $WLAN_2$ is only used to connect to a private network ”10.0.0.0/8.” A dial-up link $DialUp_0$ can only allow the web, telnet, and mail service to connect to the

Table 4.2: Service Classes

0	LAN_0 LAN_1	All	0.0.0.0
1	$WLAN_0$ $WLAN_1$	All	0.0.0.0
2	$WLAN_2$	All	10.0.0.0/8
3	$DialUp_0$	HTTP, Telnet, SMTP	140.113.0.0/16
4	$GPRS_0$ $GPRS_1$	SMTP	0.0.0.0

network ”140.113.0.0/16”. The $GPRS_0$ and $GPRS_1$ links are only allowed mail connections.

The lower class number has the higher priority to be select as candidate network interfaces. Each interface can be appeared in more than one service classes.

4.2.1 Primary path selection algorithm

Before an SCTP association starts to send packet, the primary and alternative interface should be chosen. This task was done by the following two algorithms: the Primary Path Selection Algorithm and the AlternativePath Selection Algorithm.

To select the primary path, we should first get all the available paths according to the destination addresses and the service type, by using the $getPathSet(dest, service)$ function.

After getting all the available paths, the priority of each path is calculated and compared to select the maximum. The priority of a specific path is defined by $\gamma \cdot min(Cwin_{th}/RTT, R_i) \cdot (1 - ErrorRate_i)/Cost_i$. The value $Cwin - threshold/RTT$ is the available sending-speed of a particular network interface. We make the link cost $Cost_i$ and the error rate $ErrorRate_i$ into consideration.

A path i that was previous marked as failed will never be take into consideration. However, after receiving a heartbeat message from this path will mark it as an active path again.

The primary path selection algorithm is shown in Algorithm 2.

Algorithm 2 Primary Path Selection

```
$PathSet = getPathSet(dest, service)$;
$Candidate = 0$;
for all $i$ in PathSet do
  if $i$ is mark as failed then
    $Priority_i$ = 0
  else
    $Priority_i = \gamma \cdot min(Cwin_{th}/RTT, R_i) \cdot (1 - ErrorRate_i)/Cost_i$;
  end if
  if $Priority_i > Priority_{candidate}$ then
    $Candidate = i$;
  end if
end for
$ActivePath$ = $i$;
return $Candidate$;
```

4.2.2 Alternative path selection algorithm

After selecting the primary path, the alternative path is selected with a lower priority than the primary path, as shown in Algorithm 3. The alternative path will not be the same as the current $ActivePath$ (the path that is used to sending packet).

The alternative path selection algorithm is to select a backup link of the current active path. The active path may not be the primary path.

Algorithm 3 Alternative Path Selection

```
$PathSet = getPathSet(dest, service)$;
if $ActivePath == 0$ then
    $Candidate = 1$
else
    $Candidate = 0$
end if
for all $i$ in $PathSet$ do
    if $i$ is marked as failed then
        $Priority_i = 0$;
    else
        $Priority_i = \gamma \cdot min(Cwnd_{th}/RTT, R_i) \cdot (1 - ErrorRate_i)/Cost_i$;
    end if
    if $(Priority_i > Priority_{candidate})$ and $(i! = ActivePath)$ then
        $Candidate = i$;
    end if
end for
return $i$;
```

4.2.3 Temporary failover algorithm

If a packet is sent to the destination address but the acknowledgment packet is not received before time RTO, the traffic should be in the following conditions:

Congestion There are network congestion between two endpoints. The transmission speed is reduced but the endpoint is still available.

Temporary failure The network was down due to unknown reasons for a short time. The link may be up again before the connection is terminated.

Permanent failure The network was down due to unknown reasons, but the link cannot be recovered before the connection is terminated.

SCTP uses same congestion control algorithm in TCP. If a network congestion is occurred, which means the acknowledgment packet does not arrived on time, the congestion windows size $Cwnd$ will be reduced. That is, if we do not received the acknowledgment packet at the first time, we will treat this condition as a congestion.

At this time, the packets without acknowledgment will be sent to an alternative path and the timeout value RTO_i of the primary path will be doubled, whereby the transmission speed between two endpoints will be reduced. The following packets, will be still send to the primary path instead of the alternative path.

In the worst case, if the timeout value RTO_i of the primary destination address is doubled several times until reach the maximum value RTO_{MAX} but the acknowledgment packets are still not received, we mark the link as temporary failure.

In this following algorithm, we use a threshold counter $tCounter$ to determinate where a path is temporary disabled if the $Retrans_i$ of the primary path reaches the threshold value of $tCounter$. The $tCounter$ is adjustable from 2 to $Retrans_{MAX}$.

The $ActivePath$ will be changed from the primary path to an alternative path temporary if any error occurred. Another path used for backup will be selected by the algorithm $AlternativePathSelection$, as shown in figure 3. In the same time, a heartbeat message will be sent to the primary path and wait for the response. If the heartbeat acknowledgment (HB_{ACK}) packet is received, the $ActivePath$ will be changed back to the primary path.

The temporary failover algorithm is described in Algorithm 4.

Algorithm 4 Temporary Failover

```
$PrimaryPath$ = $ActivePath$ = $PrimaryPathSelection(dest, service)$;
$AlternativePath$ = $AlternativePathSelection(dest, service, ActivePath)$;
Send packets to $ActivePath$;
start $timer$;
**if** $timer$ reaches $RTO_i$ **then**
  $RTO_i = 2 \cdot RTO_i$;
  **if** $RTO_i > RTO_{MAX}$ **then**
    $RTO_i$ = $RTO_{MAX}$;
  **end if**
  Send non ACK packets to $AlternativePath$;
  Send $HB$ packet to $PrimaryPath$;
  $Retrans_i$ ++;
  **if** $Retrans_i >= tCounter$ **then**
    $ActivePath$ = $AlternativePath$;
    $AlternativePath$ = $AlternativePathSelection(dest, service, ActivePath)$;
  **end if**
**end if**
**if** get $HB_{ACK}$ from $PrimaryPath$ **then**
  $AlternativePath$ = $ActivePath$;
  $ActivePath$ = $PrimaryPath$;
**end if**
```

4.2.4 Long-term failover algorithm

In the temporary failover algorithm, the data packets will be sent to the alternative address, but the condition of primary destination address is still monitored by sending the heartbeat message continuously and wait for the response.

If the HB_{ACK} is still not received in time, the retransmission counter $Retrans_i$ of the primary destination address will be increased and the RTO_i will be doubled each time until a maximum value (RTO_{MAX}) is reached. The default value of RTO_{MAX} is 60 seconds in SCTP protocol. After a specific period, where the re-transmission counter reaches $Retrans_{MAX}$, the primary path will be marked as failed. In this condition, the alternative path becomes the new primary path and another candidate alternative path is selected from the available PathSet. The long-term failover algorithm is shown in Algorithm 5.

Algorithm 5 Long-term Failover

```
if $timer$ reaches $RTO_i$ then
    $RTO_i$ = 2 * $RTO_i$;
    if $RTO_i > RTO_{MAX}$ then
        $RTO_i$ = $RTO_{MAX}$;
    end if
    Send non-ACK packets to $AlternativePath$;
    Send $HB$ packet to $PrimaryPath$;
    $Retrans_i$ ++;
    if $Retrans_i >= tCounter$ then
        $ActivePath$ = $AlternativePath$;
        $AlternativePath$ = $AlternativePathSelection(dest, service, ActivePath)$;
    end if
    if $Retrans_i >= Retrans_{MAX}$ and get no $HB_{ACK}$ from $PrimaryPath$ then
        Mark $PrimaryPath$ as failed;
        $PrimaryPath$ = $ActivePath$;
    end if
end if
if get $HB_{ACK}$ from $PrimaryPath$ then
    $AlternativePath$ = $ActivePath$;
    $ActivePath$ = $PrimaryPath$;
    $Retrans_i$ –;
end if
```

4.3 Performance evaluation

In this section, we use the $ns2$ network simulator [31] to evaluate the performance of our scheme. In the test framework, there are two SCTP endpoints, where each endpoint contains several interfaces with different network speed and network quality. During the simulation, a particular link will be marked as down for a specific time periods (15 seconds to 120 seconds) and up again.

In the following subsections, we compare the performance of original SCTP protocol and our modified SCTP protocol. In our modified SCTP protocol, the adjustable failover parameter ($tCount$) which is used to determinate a link is temporary disable will be set from 2 to 3. Setting this value too small (for example, 1) will cause unnecessary failover and setting this value too large (equal or larger than $Retrans_{MAX}$, the default value is 5) will cause the association to be terminated.

4.3.1 Two paths with same network speed between endpoints

In the first case, each SCTP endpoints has two interfaces to connect to Internet, both of them are wired ethernet, as shown in Figure 4.3. The endpoint 0 uses link 0 as primary path, and the link1 is used as an alternative path. The data rate of both link 0 and link 1 are set to 10M bps to simulate a real network environment. The failover parameter $tCount$ is set from 2 to 3, and the simulation time is set to 400 seconds. We will compare their performance with original SCTP protocol.

In Figure 4.4, link 0 is down alternatively for every 15 seconds and up for 15 seconds. Since this is a short period, we found the performance and the recovery time are similar. The short period of 15 seconds can be treat as congestion, which means the SCTP endpoints does not need to do temporary failover because the primary path will be recover quickly.

In Figure 4.5, link 0 is down alternatively for every 30 seconds and up for 30 seconds. We will find the network recovery quickly when $tCounter$ = 2 because the endpoint 0 will use the alternative path (link 1) to transfer packets. By using

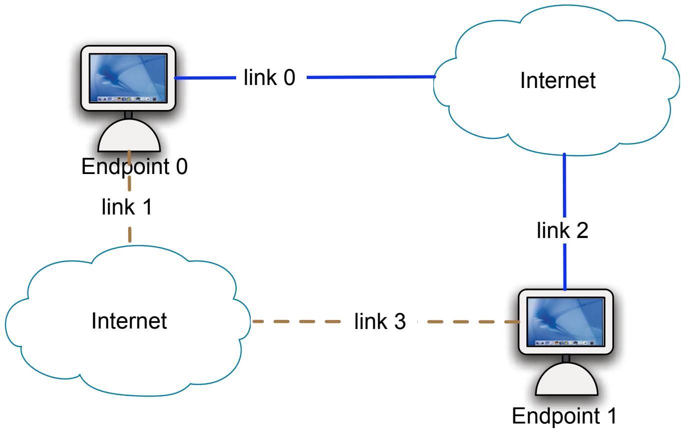


Figure 4.3: Two paths with same network speed between endpoints

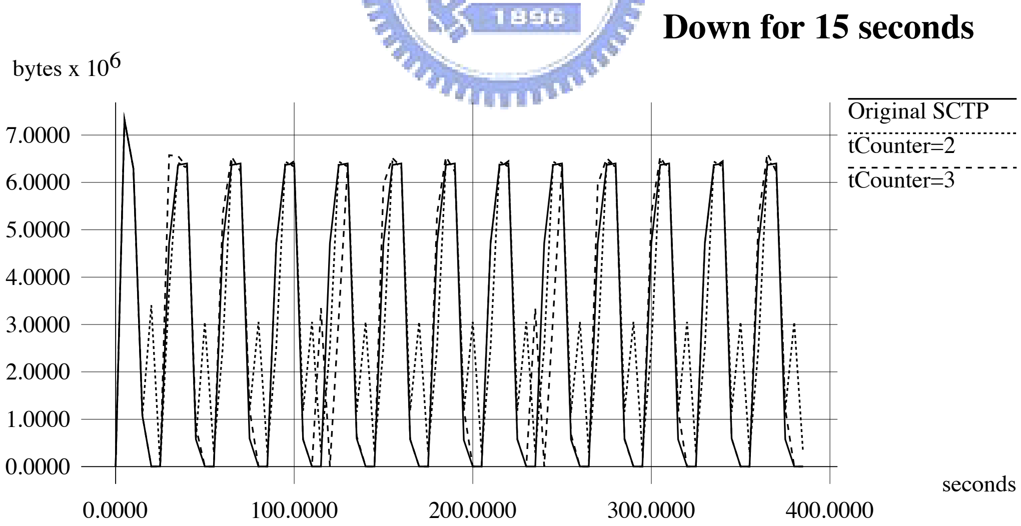


Figure 4.4: Two paths, link 0 down for 15 seconds

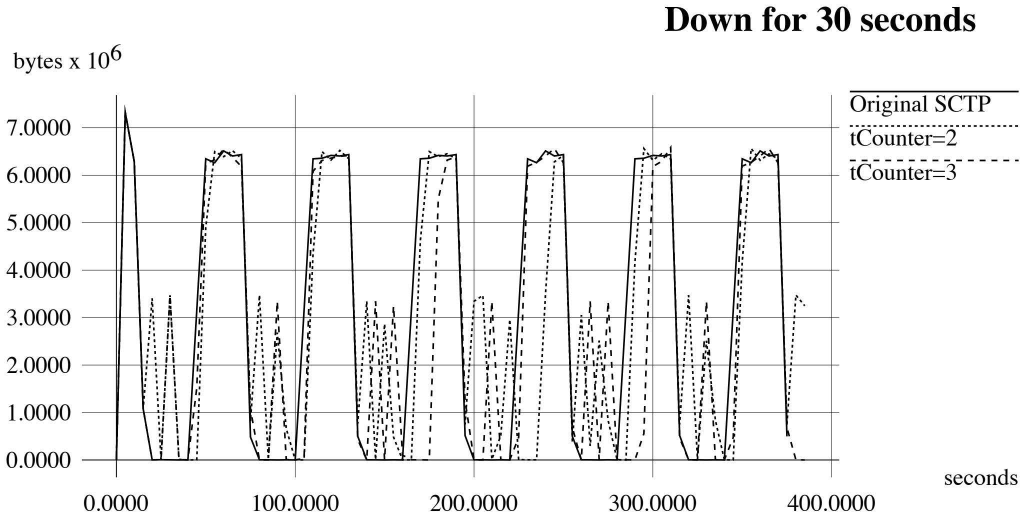


Figure 4.5: Two paths, link 0 down for 30 seconds

the original SCTP protocol, the endpoint 0 will wait for the link to be up again, which means the performance is the worst.

In Figure 4.6 and Figure 4.7, link 0 is down alternatively for every 45 and 90 seconds, respectively. In these situation, our enhanced SCTP performance significantly improve the network performance. We will find the network recovery quickly when $tCounter$ = 2. When $tCounter$ = 3, the performance is still better than original SCTP protocol.

4.3.2 Three paths with different network speed between endpoints

In the second case, each SCTP endpoints has three different links, two of them are wired network with network speed 2M bps and 1M bps, respectively. The other one is wireless connection, with network speed 0.5M bps. During the simulation

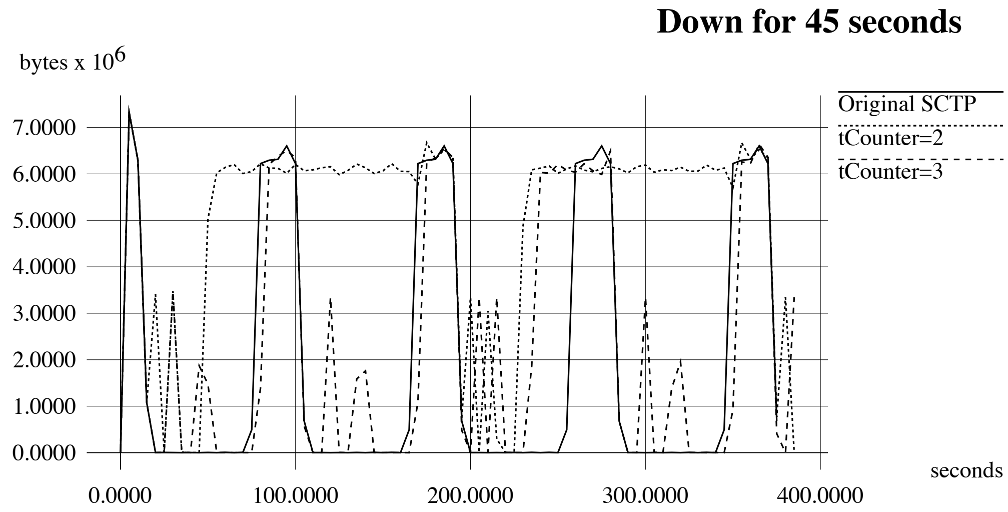


Figure 4.6: Two paths, link 0 down for 45 seconds

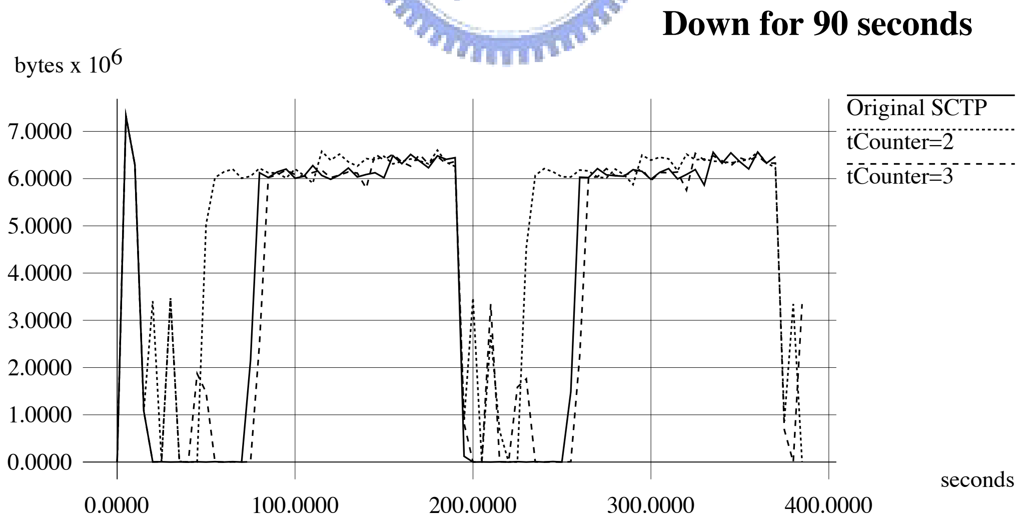


Figure 4.7: Two paths, link 0 down for 90 seconds

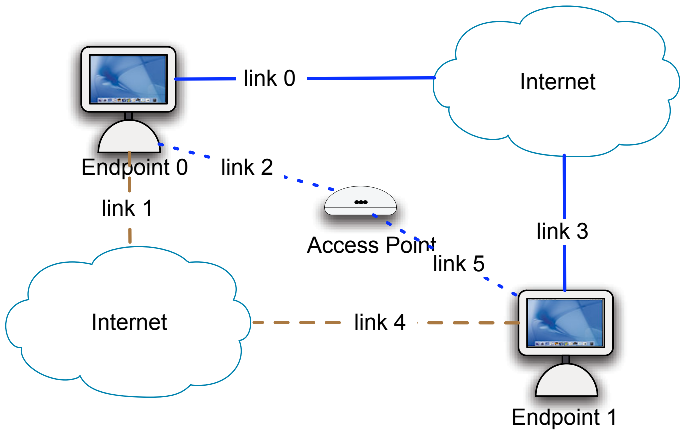


Figure 4.8: Three paths with different network speed between endpoints

time, both link 0 and link 1 will down alternatively in a specific period. The simulation time is 400 seconds.

In Figure 4.9, the network performance are not significantly better than original SCTP protocol when $tCounter = 2$ and $tCounter = 3$ because the link speed of the two backup paths are smaller than the primary path.

If the network down for a longer period, as shown in Figure 4.10, 4.11 and 4.12, we will find the network performance increase significantly. When $tCounter = 3$, the network performance are better than $tCounter = 2$ and the original SCTP protocol.

4.4 Summary

Ihe original SCTP protocol does not have a failure detection and recovery scheme to handle packet retransmission. In this chapter, we provide the following enhancement. First, each available interface will be classified into several *classes*. The operating system will select one optimal primary path and another alternative

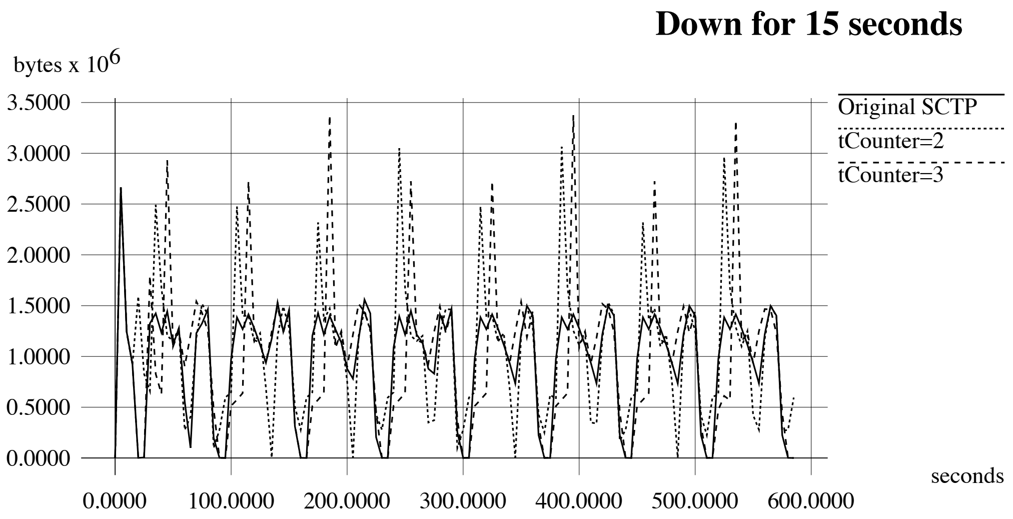


Figure 4.9: Three paths, link 0 and link 1 down for 15 seconds alternatively

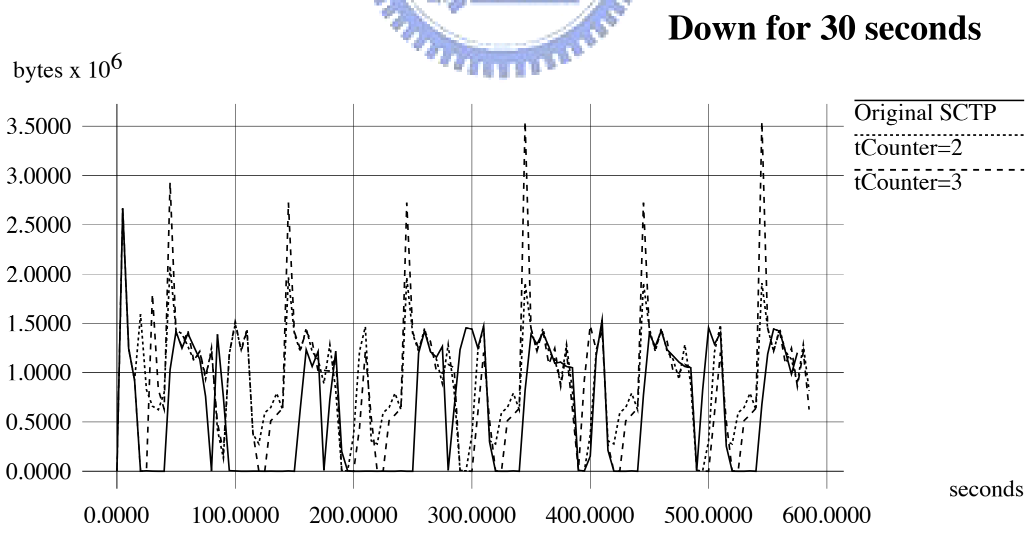


Figure 4.10: Three paths, link 0 and link 1 down for 30 seconds alternatively

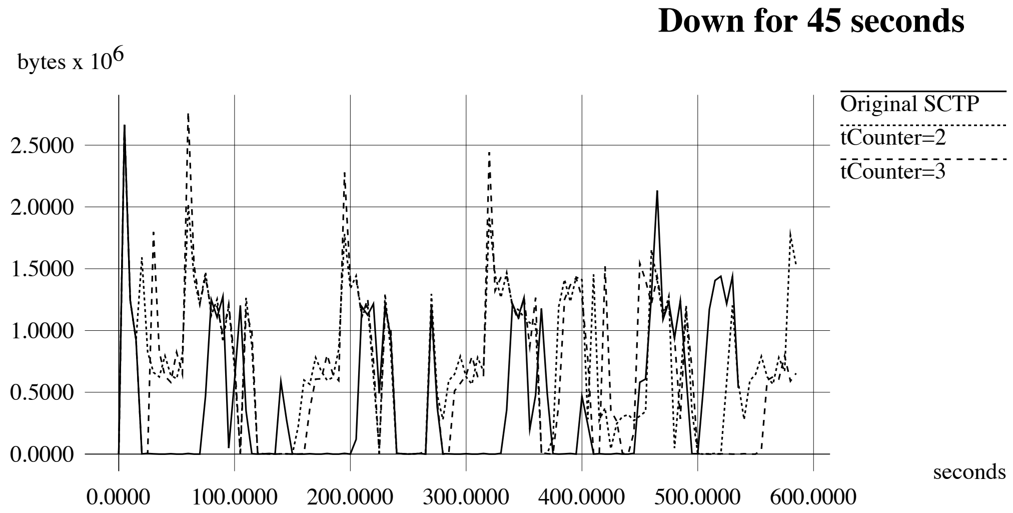


Figure 4.11: Three paths, link 0 and link 1 down for 45 seconds alternatively

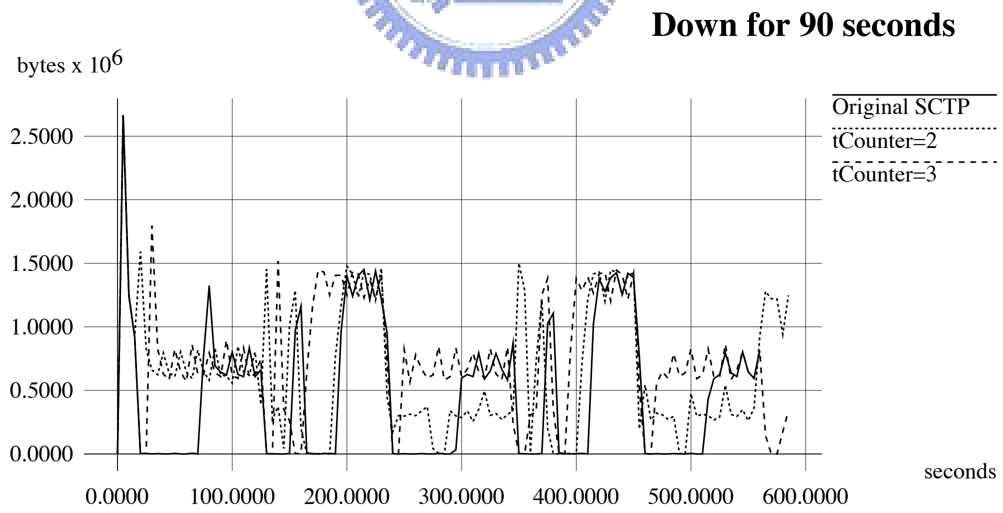


Figure 4.12: Three paths, link 0 and link 1 down for 90 seconds alternatively

path from the available classes. Second, two failover schemes are proposed to detect the network failure and make a decision to change the active path to transmit data packet. These modifications increases the transmission performance and decreases the error recovery time.

Chapter 5

Secure Communication in Multi-homed Networks

To build a secure communication environment in multi-homed networks, an authentication mechanisms is required. Most current authentication mechanisms are based on client-server architecture, which required a trusted third party (key information center). In a large-scaled multi-homing network, those authentication servers become a bottleneck because the mobile host may not reach it's home network all the time.

In this chapter, we propose a secure authentication protocol that is useful in the large-scaled multi-homed network in the Internet. In our protocol, only two communication peers are needed to participate in the authentication procedure, no trusted third party is required. The propose protocol has the property of using fewer messages to provide mutual authentication and key agreement between two communication peers. In addition, a subsequent authentication protocol is designed to minimize the overhead of the key exchange. Furthermore, a security analysis is presented to verify the strength and efficiency of the proposed protocol.

5.1 Multi-homed networks and peer-to-peer applications

Peer-to-Peer (P2P) applications offers an alternative approach to traditional client-server authentication schemes in the Internet-scale distributed environment. Comparing to client-server systems, every P2P node acts as both client and server, and provides part of the overall information available from the system. The P2P approach circumvents many problems of client-server systems such as peer identification, service allocation, node organization, security, and so on. To provide a secure P2P communication in the multi-homing network, an authentication protocol is required to offer the peers to authenticate each other's true identity and to share a session key which is only known by the participate parties.

In a multi-homing environment, there are no central servers to deal with authentication and encryption, only two peers are involved in the communication. To provide security in the multi-homing systems, it is better to use the decentralized authentication schemes because communicating peers may come from different networks and communication with a central server is difficult. Two of the famous decentralized authentication schemes are the RSA scheme and the Diffie et al's public key distribution system [13, 35]. Both of them still need a server to keep user's public information, and the correctness and security of that information must be guaranteed. In the Internet-scale P2P environment, the central server becomes a bottleneck because the server should have enough computation power, network bandwidth and data storage to deal with large amount of users. When it is unable to process requests or vulnerable to attacks, millions people may be influence.

To avoid the central server bottleneck to be used in the P2P application of the large-scale multi-homing networks, identity-based (ID-based) authentication schemes are developed [9, 42, 48]. In an ID-based authentication scheme, the public information of a user, such as name and address, are used as user's public key, which need not be stored in a central server. The first ID-based scheme proposed by Shamir [48], which rely on the existence of a trusted central authority, supports only digital signature rather than message encryption. In 1989, Okamoto and Tanaka extended Shamir's idea and combined digital signature and key distribution in a sample ID-based scheme, which supports message encryption and withstands the conspiracy problem [42]. Okamoto and Tanaka's scheme has the following problems: user identifications may be forged, user secret information may be disclosed, and the high overhead of exponential computations is needs. Tsujii proposed another ID-based cryptosystem, which suffers from the conspiracy problem, still needs high overhead of exponential computations.

In 1997, another ID-based authentication scheme are proposed by Shieh [49], in which mutual authentication and key distribution can be achieved with two messages merely between two parties involved. Then in 1999, Yen presented the cryptanalysis of Shieh's scheme, and claimed that the protocol suffers from two kinds of attacks: one is replay attack and the other is called unknown key share attack [67].

In this chapter, we enhance Shieh's scheme and propose a new authenticated key agreement protocol to be used in multi-homing. In the propose protocol, the key information center is needed only when the secure network system is being set up or when new users request to register. A new subsequent authentication phase is used to reduce the computation overhead and network traffic. Not only does our protocol need fewer exponential computations but it also resolves the security problems that appeared in the Okamoto, Tanaka, and Shieh's scheme.

5.2 Proposed secure authenticated key agreement protocol

Both the ID-based authentication scheme and key agreement technique are used in the new secure authentication protocol. If authentication and key agreement are independent, an intruder could allow two parties to carry out authentication unhindered, and could take over one party's role in key exchange [8]. In the propose protocol, the ID-based scheme is used for system setup and authentication, while the symmetric cryptographic is used for key agreement and subsequent message encryption to obtain the security and privacy.

The new authentication protocol contains three phases: the initial phase, the authenticated key agreement phase, and the subsequent authentication phase. The initial phase is completed at the key information center to set up the system, and the authenticated key agreement phase is executed between the two communication peers to achieve mutual authentication and exchange the common session key. Finally, a subsequent authentication phase is used for subsequent communications.

A. Initial phase

The information center is responsible neither for mutual authentication nor for the generation of common keys. The role of this center is to simply generate public and secret information for newly registered users. When the secure network system is setting up, the key information center will execute the following steps:

1. Choose two large prime numbers p and q, and let $n = p \cdot q$

2. Obtain the center's private key d from the following computation, which only known by the center.

$$3 \cdot q(mod\ \phi(n)) = 1 \tag{5.1}$$

   where $\phi(n) = (p-1) \cdot (q-1)$

3. Find an integer g which is a primitive element in both $GF(p)$ and $GF(q)$. We use g as the center's public information.

4. Let ID_a denote the identity of Alice. ID_a could be composed of clear-text form such as name, address, $\cdots$, and so on.

5. Choose a one-way hash function $f(x)$ to compute the extended identity (EID_a) of Alice as follows:

$$EID_a \equiv f(ID_a)(mod\ 2^N) \tag{5.2}$$

6. After computing Alice's extended identity EID_a, calculate the user's secret information S_a by the follow equation.

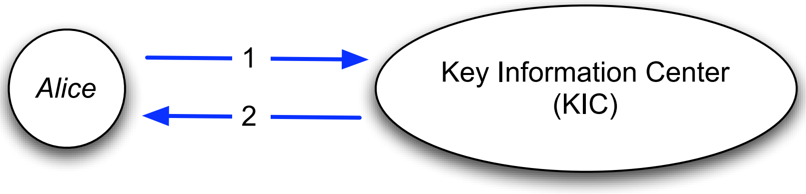
$$S_a \equiv EID_a^d (mod\, n) \tag{5.3}$$

From the relations above, the following equation would be obtained.

$$EID_a \equiv S_a^3 (mod\, n) \tag{5.4}$$

7. Send back $(n, g, f(x), S_a)$ to Alice over a secure channel. Upon receipt of the message, Alice must keep S_a secret. Alice's public information is $(n, g, f(x))$.

Once the secure network system is set up, the key information center is not needed except when new users join. The center's secret information d must be stored secretly for subsequent use. However, the two integers p and q will be no longer used and should be thrown away secretly. When a new user requests to join, he sends his ID to the center. Upon receipt of the ID, the center repeats steps 5 - 7.



1. $U_a \rightarrow KIC : ID_a$
2. $KIC \rightarrow U_a : ID_a, n, g, f(x), S_a$

Figure 5.1: Message flow in the initial phase

B. Authenticated key agreement phase

The new authenticated key agreement protocol needs three messages to complete the mutual authentication and key agreement. Upon receipt of the first message from Alice, Bob generates a session key, uses it to encrypt a nonce N_b and send back Alice. Alice tries to generate another session key, uses it to decrypt the nonce, and verifies the identity of Bob. If the verification succeeds, he believes that the message is sent by Bob, and they are using the same session key to communicate with each other. In the next step, Alice encrypts the other nonce N_a and sends the encrypted variable back to Bob. Bob then derived the nonce and uses it to verify the identity of Alice. The execution steps for mutual authentication and key agreement for a session are listed as follows.

1. If Alice wishes to communicate with Bob, she generates two random numbers r_a and N_a, and calculates the following two integers:

$$X_a \equiv g^{3 \cdot r_a} (mod\, n) \tag{5.5}$$

$$Y_a \equiv S_a \cdot N_a \cdot g^{2 \cdot r_a} (mod\, n) \tag{5.6}$$

   where N_a is used for challenge.

2. Alice sends these three integers X_a, Y_a, and N_a together with ID_a to Bob.

3. Upon receipt the message, Bob checks whether the following equation holds:

$$EID_a \cdot N_a^3 = Y_a^3 / X_a^2 \tag{5.7}$$

4. If it is true, Bob generates two random numbers r_b and N_b, and calculates

the following two integers:

$$X_b \equiv g^{3 \cdot r_b} (mod\, n) \tag{5.8}$$

$$Y_b \equiv S_b \cdot N_b \cdot g^{2 \cdot r_b} (mod\, n) \tag{5.9}$$

5. Bob calculates a session key $K_b a$ from X_a, and uses the key to encrypt the integer N_a:

$$K_{ba} \equiv X_a^{r_b} \equiv g^{3 \cdot r_a \cdot r_b} (mod\, n) \tag{5.10}$$

$$Z_b = \{N_b\} \cdot K_{ba} \tag{5.11}$$

6. Bob sends these four integer X_b, Y_b, N_b, and Z_b along with ID_b to Alice.

7. In the same way, upon receipt of the message, calculates $EID_b = f(ID_b)$ and checks whether the following equation holds:

$$EID_b \cdot N_b^3 = Y_b^3 / X_b^2 \tag{5.12}$$

8. If it is true, Alice calculates the session key $K_a b$ from X_b:

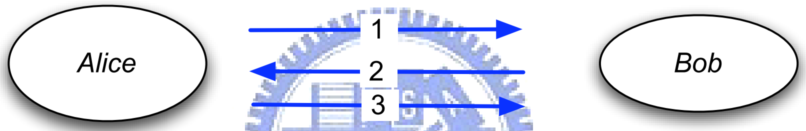
$$K_{ab} \equiv X_b^{r_a} \equiv g^{3 \cdot r_a \cdot r_b} (mod\, n) \tag{5.13}$$

9. Alice tries to derive N_a from K_{ba} and verifies if it's the same as she sent to Bob. If it is true, Alice believes the message is sent by Bob, and they are using the same session key.

10. Alice uses the session key K_{ab} to encrypt N_b and sends it to Bob:

$$Z_a = \{N_b\} \cdot K_{ab} \tag{5.14}$$

11. Upon receipt the message, Bob tries to decrypt Z_a by the generated session key K_{ba}. If he gets the correct N_b, he believes he is communicating with Alice, and agrees to use same session key to encrypt the future communicating messages. The third message (Z_b) can be sent with normal packets to reduce traffic overhead.



1. $Alice \rightarrow Bob : X_a, Y_a, N_a, ID_a$
2. $Bob \rightarrow Alice : X_b, Y_b, N_b, ID_b, Z_b$
3. $Alice \rightarrow Bob : Z_a$

Figure 5.2: The authenticated key agreement phase

C. Subsequent authentication phase

After the initial authentication, the nonce variables N_a and N_b can be used for subsequent authentication. The new nonce variables N_a' and N_b' can be derived from $f(N_a)$ and $f(N_b)$, respectively, where $f(x)$ is the same one-way function used in initial phase. Since both parties know the value of N_a and N_b from a previous session, the delivery of N_a and N_b are not required, which means only two messages is needed in the subsequent authentication phase. A new session key K_{ab}’ will be generated for each subsequent authentication, and only used in a session. Since the session keys will not be reused, replay attacks can be prevented. Both the communication parties can determine the freshness of the reply by checking the value of N_a and N_b. At the end of subsequent authentication, the client encrypts the succeeding data message in this session with the new session key. The execution steps for subsequent authentication of a session are listed as follows.

1. If Alice wishes to communicate with Bob again, she generates one random numbers r_a’, and calculates the following two integers:

$$X'_a \equiv g^{3 \cdot r'_a} (mod\, n) \tag{5.15}$$

$$Y'_a \equiv S_a \cdot f(N_a) \cdot g^{2 \cdot r'_a} (mod\, n) \tag{5.16}$$

2. Alice sends these two integers together with ID_a to Bob.

3. Upon receipt the message, Bob generates another random numbers r_b, and calculates the following two integers:

$$X'_b \equiv g^{3 \cdot r'_b} (mod\, n) \tag{5.17}$$

$$Y'_b \equiv S_b \cdot f(N_b) \cdot g^{2 \cdot r'_b} (mod\, n) \tag{5.18}$$

4. Bob calculates a session key K_{ba}’ from X_b’:

$$K'_{ba} \equiv (X'_a)^{r'_b} \equiv g^{3 \cdot r'_a \cdot r'_b} (mod\, n) \tag{5.19}$$

5. Bob checks whether the following equation holds:

$$EID_a \cdot (f(N_a))^3 = (Y_a')^3/(X_a')^2 \tag{5.20}$$

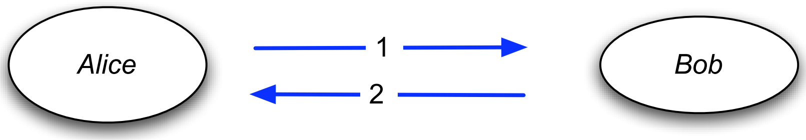
6. If it's true, Bob believes the message was sent from Alice, he will send these two integers X_b' and Y_b, along with ID_b to Alice.

7. Upon receipt of the message, Alice calculates and checks whether the following equation holds:

$$EID_b \cdot (f(N_b))^3 = (Y_b')^3/(X_b')^2 \tag{5.21}$$

8. If the equation holds, Alice believes the message is sent by Bob. Now Alice calculates the session key K_{ab} from X_b':

$$K_{ab}' \equiv (X_b')^{r_a'} \equiv g^{3\cdot r_a' \cdot r_b'}(mod\, n) \tag{5.22}$$



1. $Alice \rightarrow Bob : X_a', Y_a', ID_a$
2. $Bob \rightarrow Alice : X_b', Y_b', ID_b$

Figure 5.3: The subsequent authentication phase

Our secure authentication protocol is quite suitable for group communications. In group communicating environments, the key information center may be a computer station that can be accessed in local networks. When Alice joins the secure group, KIC stores the information (n, g, $f(x)$, S_a) in the IC card or the smart card, and sends the newly issued card to Alice. When Alice wants to authenticate with Bob on a network host, she simply inserts the card into the host computer. The card will generate or calculate the necessary information and send them to Bob just as done in the authenticated key agreement phase. Upon receiving the message, Bob follows the same procedure as Alice to establish the connection. The card must be kept with care because Alice cannot be authenticated without it.

In order to protect the secret information stored in the card and verify the person, who uses the card, the secret information stored in the card must be encrypted by the cardholder's password. In the initial phase, the KIC secretly generates a password to encrypt the secret information stored in the card. The symmetric cryptographic technique can be used to simplify the computation of encryption (decryption). When a user wants to use the card, he must provide his password to decrypt the information. Consequently, even if the card is stolen by a malicious user, he cannot masquerade as the cardholder. Both the card and the password will be sent to the cardholder in two separate certified mails, or alternatively, the password may be chosen by the cardholder when he requested for the card. These two possible ways are currently used in many commercial sectors.

5.3 Computation overhead

In the Okamoto and Tanaka's scheme, each party needs five exponential computations to complete mutual authentication and exchange a common key for each session (one for X_a, one for Y_a, two for equation check, and one for the common key calculation). Our protocol reduces the number of exponential computations for each communication session from five to two. In the authenticated key agreement phase, we can first compute g^{r_a}, then calculate X_a and Y_a as follows:

$$X_a \equiv g^{r_a} \equiv g^{r_a} \equiv g^{r_a} (mod\, n) \tag{5.23}$$

$$Y_a \equiv S_a \cdot N_a \cdot \equiv g^{r_a} \equiv g^{r_a} (mod\, n) \tag{5.24}$$

No exponential computation but multiplication is needs in these two equations. The verification of sender's identity (see equation 5.7) can also be accomplished without exponential computation in the same way. Therefore, our protocol needs only two exponential computations (one for g^{r_a}, and one for common key $(X_a)^{r_a}$).

5.4 Security analysis

Our protocol provides session key agreement and the authenticity of communicating parties to guarantee the privacy and security of network communication. The security relies on the difficulty of computing the discrete logarithm problem, which does not have the conspiracy problem existing in the Tsujii's scheme. If a forger wants to masquerade Alice to communicate with others, he must find two integers x and y satisfying the following equation:

$$y^3 = EID_a \cdot N_a^3 \cdot x^2 \tag{5.25}$$

The use of low public exponents in this equation does not lower the difficulty to crack (y, x). Although the forger can get a pair of integers (y^3, x^2) that makes the equation hold, the pair (y, x) is unattainable because computing (y, x) pair from (y^3, x^2) is a discrete logarithm problem.

In our protocol, a small integer 3 is used as the public exponent in equation (5.1). The selection of small integer is for the purpose of reducing computation overhead. For a general public exponent e, our protocol still works fine. The use of a low public exponent does not lower the difficulty to crack the user secret information S_a, because the computation of S_a in equation (5.4) is a discrete logarithm problem [1]. Though some low exponent attacks [1, 7] are proposed, both of them do not work in our protocol. Hastad proposed an attack on using RSA with low exponents in a public key network [13]. To illustrate this attack, suppose that a message m is broadcasted to three parties in which the public exponents are e_1 = e_2 = e_3 = 3, and in which the modulus are n_1, n_2, and n_3. The encrypted messages are

$$m^3 \; mod \; n_1, m^3 \; mod \; n_2, m^3 \; mod \; n_3 \tag{5.26}$$

Using the Chinese remainder theorem, one can find $m3modn_1 \cdot n_2 \cdot n_3$. However, $m3 < n_1 \cdot n_2 \cdot n_3$ because $m < n1, n2, n3$. Therefore, m^3 is not affected by being reduced modulo $n_1 \cdot n_2 \cdot n_3$. This attack will not succeed in our protocol, because the same modulus n is used for all parties. This attack will not succeed, since our protocol uses the same modulus n for all parties.

In 1996, Coppersmith and et al presented another low exponent attack [7] in which encrypted messages may be recovered under RSA with a public exponent of 3, if a cracker can get α and β such that two messages m_1 and m_2 satisfy $m_2 = \alpha m_1 + \beta$. This attack will not work in our protocol, since an outsider has no idea of the relation of users' secret information S_a to crack equation (5.4).

In 1999, Yen [67] presented a replay attack on our previous authentication protocol [49]. That is, an intruder may derive the forged Y_a' and N_a' to pass the verification of equation (5.7) by performing the following computation on previous intercepted X_a, Y_a and N_a.

$$Y_a' \equiv Y_a' \cdot (N_a)^{-1} \cdot N_a' \; (mod\, n) \tag{5.27}$$

The intruder then sends the forged pair (X_a, Y_a') and N_a' to Bob. Bob may consider the message fresh, then choose a new random number r_b' and reply the following message to the intruder.

$$X_b' = g^{3 \cdot r_b'} \; (mod\, n) \tag{5.28}$$

$$Y_b' \equiv S_b \cdot N_a' \cdot g^{2 \cdot r_b'} \; (mod\, n) \tag{5.29}$$

As a result, the session key of this communication session is $K' = g^{3 \cdot r_a \cdot r_b'}$, rather than the old session key $K' = g^{3 \cdot r_a \cdot r_b}$. The intruder cannot compute the new common key K' without knowing the random number r_a. Since the old messages are all encrypted with the old common key K, the intruder cannot successfully replay the following messages he eavesdropped. Furthermore, the intruder cannot compute the response message Z_a without knowing the new session key K', because N_b should be encrypted by the new session key and sends it back to Bob. Consequently, the replay attack proposed by Yen does not succeed in our authenticated key agreement protocol.

Another attack proposed by Yen is called unknown key share attack [67], but the attack does not succeed in our authenticated key agreement protocol. Suppose an intruder has intercepted a pair (Y_A, X_A) of the authentication procedure between Alice and Bob, he then chooses a random number R and computes a new pair($Y'_A = Y_A R^2, X'_A = X_A R^3$). If the intruder sends the new pair (Y'_A, X'_A) to Bob, the verification of equation (5.14) will succeed because

$$Y'^{3}_{A}/X'^{2}_{A} = {Y_A}^3 R^6/{X_A}^2 R^6 = {Y_A}^3/{X_A}^2 = EID_A \tag{5.30}$$

This implies that Bob may derive the wrong session key $K'_{AB} = (X'_A)^{r_B}$. This attack will be detected in step 9 and 11, when user A and B try to use their session key to decrypt the nonce N_A and N_B from the response message Z_B and Z_A. In step 5 of authentication phase, Bob should first compute Z_B by using the new session key K'_{BA} to encrypt the challenge variable N_A. In step 10, Alice should also compute Z_A by using her generated session key K'_{AB} to encrypt the challenge variable N_B. Alice and Bob exchange Z_A and Z_B in step 6 and step 10. Since K_{AB} is not equal to K'_{AB}, the decryption of N_A and N_B will be failed. Bob can notice the conflict of session keys and detects the attack in step 11. With the verification of N_A and N_B, the unknown key share attack can be early detected during authenticate key agreement phase. Consequently, the unknown key share attack does not work in our protocol.

5.5 Summary

In this chapter, we proposed an ID-based authenticated key agreement protocol to be used in the multi-homing network. In the proposed protocol, both the key information center and files for the storage of public information are not required. Once the secure network system is set up, the authentication and key agreement can be handled solely by the two peers involved, instead of the key information center. This protocol resolves the problems appeared in the Okamoto, Tanaka's, and Shieh's scheme. In contrast to five exponential computations needed in the Okamoto and Tanaka's scheme, our protocol needs only two exponential computations for mutual authentication and key exchange, thereby greatly reducing the load on communication devices. Though Yen claimed that our previous protocol suffers from two possible attacks, this paper presents that the two attacks do not succeed. To avoid confusion, we revise the protocol accordingly. We also explain that the low exponent attacks do not work in our protocol.

Chapter 6

Conclusion

In current scenario, use of mobile and Internet has been increasing and the increasing number of users are coming forward to use new services like mobility and multihoming. Wide variety of wireless networks will be merged into the Internet and allow users to continue their application with higher degree of mobility. In such environment, multimedia applications, which require smooth rate transmission, will become more popular.

Roaming users are interested to stay connected with network while moving from one network to another network with multiple network interfaces e.g. WLAN and GPRS. Problems which might arise when mobile host moves from one network to another network, especially the resides network is a private network.

To solve these problems, we proposed an cascade private network architecture to be used in the Internet, an multi-homing protocol to solve the roaming problem and an authentication protocol to solve the authentication issues. The following summaries our works.

Chapter 2 reviewed the network address translation (NAT) and its variants.

NAT is considered a solution to the insufficiency of IPv4 address space. Regardless of address space consideration, NAT can still be considered a solution in IPv6 for security or load-sharing concerns. In the near future, Internet service providers may only provide NAT solutions to small enterprise networks due to the shortage of address space. In such trends, existing Internet security protocols must be re-examined together with this new network environment.

In chapter 3 we designed an transparency routing architecture for multi-level private networks called MRSIP. The MRSIP framework is proposed to replace the original NAT and RSIP network architecture. The MRSIP framework introduced multi-level private network architecture. The concept of using multiple address pools reduces the necessary of public addresses. An MRSIP client may request several IP addresses to access hosts in different address realms.

In chapter 4 we discuss the problems in the multi-homing network: IP allocation, handover procedure, and the compatibility with NAT boxes. We propose an enhanced SCTP mobility scheme, including two path-selection algorithms and two failover algorithms. The enhanced SCTP uses these failover algorithms to choose a new path by application type and network status. This scheme reduces the handover cost.

Finally in chapter 5 we proposed an ID-based authenticated key agreement protocol to be used in the multi-homing network. In the proposed protocol, both the key information center and files for the storage of public information are not required. Once the secure network system is set up, the authentication and key agreement can be handled solely by the two peers involved, instead of the key information center.

Bibliography

[1] L. Adleman, “A Subexponential Algorithm for the Discrete Logarithm Problem with Applications to Cryptography,” in *20th IEEE symposium foundations of computer science*, 1979, pp. 55–60.

[2] P. Bellavista, A. Corradi, R. Montanari, and C. Stefanelli, “Dynamic Binding in Mobile Applications A Middleware Approach,” *IEEE Internet Computing*, pp. 34–42, April 2003.

[3] M. Borella, D. Grabelsky, J. Lo, and K. Tuniguchi, “Realm Specific IP: Protocol Specification,” *Internet Draft*, January 2000. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-nat-rsip-protocol-07.txt

[4] B. Callaghan, M. Eisler, R. Thurlow, D. Robinson, S. Shepler, D. Noveck, and C. Beame, “NFS Version 4 Protocol,” *Internet-Draft*, February 2000. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-nfsv4-06.txt

[5] A. L. Caro, J. R. Iyengar, P. D. Amer, G. J. Heinz, and R. R. Stewart, “Using SCTP Multihoming for Fault Tolerance and Load Balancing,” *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 3, p. 23, July 2002.

[6] L. Coene, "Multihoming Issues in the Stream Control Transmission Protocol," *IETF Internet Draft, draft-coene-sctp-multihome-03.txt*, February 2002.

[7] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, "Low-exponent RSA with Related Messages," in *Advances in Cryptology-Eurocrypt 96*, 1996, pp. 1–9.

[8] W. Diffie, "Authentication and Authenticated Key Exchanges," *Design, Codes, and Cryptography*, vol. 2, pp. 107–125, 1992.

[9] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[10] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merwe, "Resource Management With Hoses: Point-to-Cloud Services for Virtual Private Networks," *IEEE/ACM Transaction on Networking*, vol. 10, no. 5, pp. 679–692, October 2002.

[11] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," *RFC-2827*, May 2000.

[12] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)," *RFC 2409*, November 1998.

[13] J. Hastad, "On Using RSA with Low Exponent in a Public Key Network," in *Proceedings of the Advances in Cryptology (CRYPTO'85)*, 1985, pp. 403–408.

[14] M. Holdrege and P. Srisuresh, “Protocol Complications with the IP Network Address Translator (NAT),” *Internet Draft*, March 2000. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-nat-protocol-complications-02.txt

[15] H.-Y. Hsieh and R. Sivakuma, “Transport Layer Issues: A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts,” in *Proceedings of the 8th annual international conference on Mobile computing and networking*, 2002, pp. 83–94.

[16] C.-Y. Huang, J.-C. Liu, J. cheng Chen, M.-C. Jiang, and H.-W. Lin, “Integration of NAT with Mobile IP.”

[17] Z. J. Hui Luo, B.-J. Kim, N. Shankaranarayanan, and P. Henry, “Integrating Wireless LAN and Cellular Data for the Enterprise,” *IEEE Internet Computing*, pp. 25–33, March 2003.

[18] G. J. H. II, “Priorities in Stream Transmission Control Protocol (SCTP) multistreaming,” Master’s thesis.

[19] ITU-T, “Visual Telephone Systems and Equipment for Local Area Networks which Provide a Non-Guaranteed Quality of Service,” *ITU-T Recommendation H.323*, November 1996.

[20] J. R. Iyengar, A. L. Caro, P. D. A. Jr., G. J. Heinz, and R. R. Stewart, “Making SCTP More Robust to Changeover.”

[21] ——, “Retransmission Policies With Transport Layer Multihoming.”

[22] ——, "SCTP and TCP Variants: Congestion Control Under Multiple Losses."

[23] J. R. Iyengar, K. C. Shah, P. D. Amer, and R. Stewart, "Concurrent Multipath Transfer Using SCTP Multihoming."

[24] S.-E. Jeon, R. T. Alber, J. A. Copeland, and Y. Pan, "Path Selection With Class Distribution Information in the Integrated Network," *IEEE COMMUNICATIONS LETTERS*, vol. 6, no. 2, February 2002.

[25] D. B. Johnson and C. E. Perkins, "Mobility Support in IPv6," *Internet-Draft*, November 2001.

[26] A. L. C. Jr., J. R. Iyengar, P. D. Amer, G. J. Heinz, and R. R. Stewart, "A Two-level Threshold Recovery Mechanism for SCTP," 2002. [Online]. Available: www.cis.udel.edu/~amer/PEL/poc/pdf/sci2002-acaro.pdfhttp://citeseer.ist.psu.edu/616657.html

[27] K. Kar, M. Kodialam, and T. V. Lakshman, "Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 18, no. 12, December 2000.

[28] S. Kent and R. Atkinson, "IP Authentication Header," *IETF RFC 2402*, November 1998.

[29] ——, "IP Encapsulating Security Payload (ESP)," *IETF RFC 2406*, November 1998.

[30] ——, “Security Architecture for the Internet Protocol,” *RFC 2401*, November 1998.

[31] E. Kevin Fall and K. Varadhan, “The network simulator - ns2.” [Online]. Available: http://www.isi.edu/nsnam/ns/

[32] E. Korach and R. Ohayon, “Path Selection in Networks with Non-Deterministic Information,” *IEEE*, pp. 2227–2231, 2002.

[33] S. Ladha and P. D. Amer, “Improving Multiple File Transfers Using SCTP Multistreaming.”

[34] S. Ladha, P. D. Amer, A. L. C. Jr., and J. R. Iyengar, “Improving file transfer in FCS networks.”

[35] E. Lee, H. Chae, B. Park, and M. Choi, “An Expanded NAT with Server Connection Ability,” in *IEEE Region 10 Conference TENCON 99*, 1999.

[36] Y.-K. Lee and S. Lee, “Path Selection Algorithms for Real-time Communication,” 2001, pp. 398–404.

[37] R. E. Lennon, “Cryptography Architecture for Information Security,” *IBM Systems Journal*, vol. 17, no. 2, pp. 138–150, 1978.

[38] Y.-B. Lin, Y.-C. Lo, , and H. C.-H. Rao, “A Push Mechanism for GPRS Supporting Private IP Addresses,” *IEEE Communication Lettors*, vol. 7, no. 1, pp. 24–26, January 2003.

[39] H. Matsuoka, T. Yoshimura, and T. Ohya, “A Robust Method for Soft IP Handover,” *IEEE Internet Computing*, pp. 18–24, March 2003.

[40] C. Metz, “The Latest in Virtual Private Networks: Part I,” *IEEE Internet Computing*, pp. 87–91, January 2003.

[41] G. Montenegro and M. Borella, “RSIP Support for End-to-End IPsec,” *IETF Internet Draft*, February 2000.

[42] B. C. Neuman, “Kerberos: An Authentication Service for Computer Networks,” *IEEE Communications*, September 1994.

[43] J. M. Ng and P. K. Ng, “Cost-Delay Path Selection Function for Real-Time Multicast Routing.”

[44] E. Okamoto and K. Tanaka, “Identity-based Information Security Management System for Personal Computer Networks,” *IEEE Journal on Selected Areas In Communications*, vol. 7, pp. 290–294, 1989.

[45] Y. Pan, M. Lee, J. B. Kim, and T. Suda, “Roaming and Handoff Management: An End-to-end Multi-path Smooth Handoff Scheme for Stream Media,” in *Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, Sept. 2003.

[46] A. Pasztor and D. Veitch, “On the Scope of End-to-End Probing Methods,” *IEEE Communication Lettors*, vol. 6, no. 11, pp. 509–511, November 2002.

[47] C. E. Perkins, “IP Mobility Support for IPv4,” *IETF RFC 3220*, January 2002.

[48] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, “Address Allocation for Private Internets,” *RFC 1918*, February 1996.

[49] A. Shamir, “Identity-based Cryptosystems and Signature Schemes,” in *In proceedings of Crypto-84, Santa Barbara, CA*, 1984, pp. 47–53.

[50] S. Shieh, W.-H. Yang, and H.-M. Sun, “An Authentication Protocol Without Trusted Third Party,” *IEEE Communication Letters*, vol. 1, no. 3, pp. 87–89, May 1997.

[51] R. Srinivasan, “RPC: Remote Procedure Call Protocol Specification Version 2,” *RFC 1831*, August 1995. [Online]. Available: http://www.ietf.org/rfc/rfc1831.txt

[52] P. Srisuresh, “Security Model with Tunnel-mode IPsec for NAT Domains,” *RFC-2709*, October 1999.

[53] P. Srisuresh and K. Egevang, “Traditional IP Network Address Translator (Traditional NAT),” *Internet-Draft*, September 1999. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-nat-traditional-03.txt

[54] P. Srisuresh and D. Gan, “Load Sharing using IP Network Address Translation (LSNAT),” *RFC-2391*, August 1998.

[55] P. Srisuresh and M. Holdrege, “IP Network Address Translator (NAT) Terminology and Considerations,” *RFC 2663*, August 1999.

[56] P. Srisuresh, G. Tsirtsis, P. Akkiraju, and A. Heffernan, “DNS extensions to Network Address Translators DNS_ALG,” *RFC 2664*, September 1999.

[57] J. G. Steiner, B. C. Neuman, and J. I. Schiller, “Kerberos: an authentication service for open network systems,” *Proceedings of the winter 1988 USENIX Conference*, pp. 191–201, February 1988.

[58] T. Stergiou, R. J. Green, and M. S. Leeson, “Protocol Stack Design for 3rd Generation Mobile Systems: UMTS Core Network.”

[59] R. Stewart, M. Ramalho, M. T. Q. Xie, I. Rytina, M. Belinchon, and P. Conrad, “Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration,” *IETF Internat-Draft, draft-ietf-tsvwg-addip-sctp-07.txt*, February 2003.

[60] R. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, “Stream Control Transmission Protocol,” *RFC 2960*, October 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2960.txt

[61] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications,” *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 11, no. 1, pp. 17–32, February 2003.

[62] C. System, *Cisco IOS Solutions for Network Protocols, Volume I: IP*. Macmillan Technical Publishing, 1998.

[63] D. Tipper, J. L.Hammond, S. Sharma, A. Khetan, K. Balakrishnan, and S. Menon, “An analysis of the congestion effects of link failures in wide area networks,” *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 1, pp. 179–192, January 1994.

[64] G. Tsirtsis and P. Srisuresh, “Network Address Translation - Protocol Translation (NAT-PT),” *IETF RFC-2766*, February 2000.

[65] S. Tsujui, T. Itho, and K. Kurosawa, "ID-based Cryptosystem Using Discrete Logarithm Problem," *IEEE Electronic Letter*, vol. 23, pp. 1318–1320, November 1987.

[66] B. Wellington, "Secure domain name system (dns) dynamic update (rfc 3007)," November 2000. [Online]. Available: http://www.ietf.org/rfc/rfc3007.txt

[67] S.-M. Yen, "Cryptanalysis of an Authentication and Key Distribution Protocol," *IEEE Communication Letters*, vol. 3, no. 1, pp. 7–8, January 1998.

[68] M. Ylianttila, R. Pichna, J. Vallstrom, J. Makela, A. Zahedi, P. Krishnamurthy, and K. Pahlavan, "Handoff Procedure for Heterogeneous Wireless Networks," in *Proceedings of Global Telecommunications Conference 1999*, 1999, pp. 2783–2787.

[69] X. Yuan and A. Saifee, "Path Selection Methods for Localized Quality of Service Routing."

Curriculum Vitae

Jia-Ning Luo received his B.S. degree in Electrical Engineering and M.S. degree in Computer Science Engineering from Tatung University. He is currently a Ph.D. candidate in the department of Computer Science and Information Engineering, National Chiao Tung University. Jia-Ning Luo is interested in distributed systems and network security.

Publication List

- Journal Paper

    1. Jia-Ning Luo, Shiuhpyng Shieh, and Ji-Chiang Shen, ”Secure Authentication Protocols Resistant to Guessing Attacks,” *accepted by Journal of Information Science and Engineering*.
    2. Shu-Ming Cheng, Shiuhpyng Shieh, Weng-Her Yang, Fu-Yuan Lee and Jia-Ning Luo, ”Designing Authentication Protocols for Third Generation Mobile Communication Systems,” *Journal of Information Science and Engineering*, Vol 21, 2005, pp.361-378
    3. Shiehpyng Shieh, Y.L. Huang, F.S. Ho, Jia-Ning Luo, ”Network Address Translators: Effects on Security Protocols and Applications in the TCP/IP Stack,” *IEEE Internet Computing*, Nov./Dec. 2000.

- Conference Paper

    1. Chang-Han Jong, Shiuhpyng Shieh and Jia-Ning Luo, ”Detecting Distributed DoS/Scanning by Anomally Distribution of Packet Fields,” *ICS'2002*, Hua-Lien, Taiwan, Dec. 2002.

2. Shiuhpyng Shieh and Jia-Ning Luo, ”An ID-Based Authenticated Key Agreement Protocol,” *ACM. International Conference On Information Security*, Shanghai, China, July 2002

3. Jia-Ning Luo and Shiuhpyng Shieh, ”The Multi-Layer RSIP Framework,” *Proceeding of Ninth IEEE International Conference on Networks*, Bangkok, Thailand, July 2001, pp.166-171

4. Jia-Ning Luo, Kuo-Cheng Lee and Shiuhpyng Shieh, ”A Transparent and Efficient Mechanism for File System Protection,” *Information Security Conference*, Taiwan, May 2000.

5. Jia-Ning Luo and Shiuhpyng Shieh, ”A Secure Mail System for Mobile Users in the Internet,” *Proceeding of the Third Mobile Computing Workshop*, HsinChu, Taiwan, March 25-26, 1997