SEAMLESS HANDOVER AMONG HETEROGENEOUS MOBILE NETWORKS USING STREAM CONTROL TRANSMISSION PROTOCOL (SCTP)

ENG SE-HSIENG

(B.Eng (Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

Acknowledgments

The research work presented herein has benefited from the generous support accorded to it by my supervisor, A/Prof Hari K Garg and the Department of Electrical and Computer of Engineering of NUS.

Mr. Cyrille Colin, a fellow MEng candidate, has taken part in countless brainstorming sessions and has been an invaluable partner throughout the course of this work.

The developers of the Stream Control Transmission Protocol, Mobile Stream Control Transmission Protocol and the Linux kernel development project have been extremely patient and responsive to my queries. Even though we have never met, the discussions we have exchanged have given me a greater understanding of the issues surrounding transport protocol development.

I am grateful to Mr. Eric Siow Hong Lin of the ECE-I^2R Laboratory for Wireless Communications for his assistance in procuring the necessary equipment for this project.

Last but not least, I thank my family, especially my late father, for their enduring love and understanding.

Table of Contents

Summary

This dissertation focuses on the development of a transport-layer mobility management solution using Stream Control Transport Protocol (SCTP), a transport protocol that is being adopted for general data transport by the Internet Engineering Task Force (IETF). SCTP supports multi-homing and allows endpoints to use multiple heterogeneous links in an association. An enhancement known as the dynamic address reconfiguration extension (ASCONF) defines procedures to gracefully add and remove network addresses from an active connection without the need to restart the connection. These advantages over traditional TCP make SCTP a promising candidate for transport layer mobility management in networks.

This work establishes key weaknesses of SCTP impeding its deployment. Network Address Translator (NAT) traversal issues have yet to be adequately resolved and SCTP suffers from poor fault resilience in asymmetric multi-homed topologies. To address these issues, modifications to the protocol were designed, implemented and extensively tested in the course of the research work reported herein.

The key benefits of the proposed modifications lie in the fact that SCTP would be fully compatible with IPv4 networks. Any alterations are kept to a minimum and do not contradict protocol specifications. Most importantly, mobile hosts may make use of the modified SCTP to seamlessly roam between heterogeneous networks.

Seamless handover between WLAN and GPRS based on SCTP is demonstrated using an application that is developed and tested on existing commercial networks. This establishes that with the proposed modifications in place, SCTP may be easily supported by existing heterogeneous wireless network architectures.

List of Abbreviations

ADDIP	Dynamic addition of IP address to a SCTP association
ALG	Application Layer Gateway
ASCONF	Dynamic address reconfiguration extension for SCTP
ASCONF-ACK	ASCONF acknowledgment
DELIP	Dynamic deletion of IP address to a SCTP association
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
I-D	Internet-Draft
IETF	Internet Engineering Task Force
IESG	Internet Engineering Steering Group
INIT	Initialization of a SCTP association
INIT-ACK	Initialization acknowledgment of a SCTP association
IP	Internet Protocol
LAN	Local Area Network
MTP	Message Transfer Part
NAT	Network Address Translator
OOTB	Out-of-the-Blue
PCMCIA	Personal Computer Memory Card International Association
PDA	Personal Digital Assistant
RFC	Request For Comments
SCTP	Stream Control Transmission Protocol
SETREMPRI	Setting of remote primary IP address in a SCTP association

SIGTRAN	Signaling Transport Working Group
TCP	Transmission Control Protocol
TSVWG	Transport Area Working Group
UDP	User Datagram Protocol
USB	Universal Serial Bus
WCDMA	Wideband Code Division Multiple Access
WLAN	IEEE 802.11 Wireless Local Area Network

List of Figures

Chapter 1 Introduction

1.1 Background

With the increasing trend of wireless network technologies, end-users are looking to move easily from place to place while retaining access to network services. While cellular technologies meet this demand, they traditionally offer low data rates, such as 21.4kbps or less in GPRS networks [1]. Although 3G cellular networks can support an aggregated high speed up to 2Mbps for indoor/small cell environment or 384kbps for wide area [2], voice remains the major revenue-generating source and most bandwidth will be reserved for voice users. For high-speed wireless connectivity, users are now looking to the IEEE 802.11 Wireless Local Area Network (WLAN).

WLAN is designed as an extension to wired Local Area Networks (LAN) and offers a data rate of up to 11Mbps. Since its release in 1999, 802.11b [3] WLAN has been widely deployed in offices, educational institutions, homes and public hotspots such as cafes, hotels and airports. WLAN network adaptors are now available as Personal Computer Memory Card International Association (PCMCIA) cards and Universal Serial Bus (USB) adaptors for laptops as well as CompactFlash cards for Personal Digital Assistants (PDA). However, a serious disadvantage of WLAN is its limited coverage. For example, in the National University of Singapore (NUS), users need to be within 60 meters of a WLAN access point.

As mobile end-users enter the coverage area of each network, they will be looking to access network services anywhere, anytime. With cellular technologies as an always-on backup, users may wish to pop into the nearest hotspot or use the waiting time in an airport to perform data-intensive tasks on the WLAN backbone.

Personal mobility [4] enables a person to access services irrespective of his location and the terminal he is using. For example, a user may possess both a GPRS cellular phone and a WLAN-enabled PDA. However, the convergence of PDAs and cellular phones, as well as the increasing portability of tablet PCs and laptops, imply that users will be able to access both cellular and WLAN technologies on the same device. Terminal mobility enables devices to receive continued access to services, independently of their location and while moving. This thesis focuses on enabling terminal mobility.

Devices which have multiple network interfaces and IP addresses are termed "multi-homed". The Stream Control Transmission Protocol (SCTP) [5][6] from the Internet Engineering Task Force (IETF) supports multi-homing and allows hosts to include several IP addresses in a connection. SCTP was originally designed by IETF for signalling networks. However, it has since been elevated to stand beside UDP and TCP as a general-purpose transport protocol. It is a secure, connection-oriented protocol that supports multi-homing and multiple streams, making it particularly suited for wireless environments [7] [8].

Multi-homing allows a network session to be more resistant to network failure, which would cause a singly-homed host to be temporarily unreachable. SCTP further allows the endpoint to securely and efficiently migrate sessions from one network link to another, providing a means for reliable failover recovery. With the Dynamic Address Reconfiguration (ASCONF) extension [9], SCTP endpoints can even modify the list of network addresses and select the primary network interface for receiving packets. Unlike traditional TCP, the connection does not need to be restarted to switch from one link to another. SCTP thus provides the potential for mobile hosts to enjoy seamless handover between heterogeneous networks.

Mobility management may be defined in broad terms as the ability to keep track of a mobile host's movements [10]. This may be achieved in several ways. Cellular technologies and WLAN support mobility within their respective technologies and this is known as link-layer mobility. Network-layer mobility solutions, such as Mobile IP [11], are routing-based approaches that allow devices to be reached at their original network addresses even when they are in foreign networks. However, the deployment of Mobile IP would require substantial changes to IP architecture. Transport layer mobility management [12] allows mobile hosts to roam seamlessly between heterogeneous networks, using link-layer information to track the presence of links. This requires support from end-hosts and applications but requires no change to the IP substrate and may be easier to deploy. With SCTP's support for multi-homing, SCTP has been identified as a transport protocol that can lead to a transport-layer mobility management solution.

The objective of this work is to examine the feasibility of SCTP as a transport-layer mobility management solution in mobile networks. The critical weakness that may hamper the adoption of SCTP and ASCONF in wireless networks is identified as the lack of a viable Network Address Translators (NAT) traversal solution. A complete NAT traversal solution is defined and implemented.

It has previously been identified that SCTP provides poor fault resilience in asymmetric multi-homing topologies [5]. Asymmetric multi-homing topologies arise when an endpoint has fewer network interfaces than its counterpart, a likely scenario when mobile clients with several low-bandwidth links connect to a server with one high-speed link. Enhancing SCTP with source address selection would circumvent association breakdown in asymmetric topologies [13]. In the course of developing the

NAT traversal solution, a novel and simple method of source address selection for SCTP has been devised.

Modifications to the protocol are kept to a minimum. Original protocol definitions of SCTP in Request for Comments (RFC) documents are adhered to, with modifications being proposed only for procedures that are outlined in Internet-Drafts (I-D).

The proposed enhancements to SCTP and ASCONF are implemented and extensively tested. To demonstrate seamless handover and mobility management, an application is developed to transparently switch between GPRS and WLAN during an active file transfer. The wireless access technologies used as examples in this thesis are GPRS and WLAN, but the results apply to IP-based networks in general.

1.2 Main contributions

The main contributions of this work are as follows:

- Establishes critical weaknesses impeding the deployment of the Stream Control Transmission Protocol (SCTP) and its Dynamic Address Reconfiguration (ASCONF) extension in IPv4 networks
- Proposes modifications, which have been implemented and tested, to SCTP and ASCONF procedures to allow for Network Address Translator (NAT) traversal
- Enhances SCTP with a source address selection feature that may resolve the single point of failure in asymmetric network topologies.
- Demonstrates internetworking between existing commercial GPRS and WLAN networks using the modified SCTP.

1.3 Organization of the thesis

Chapter 2 provides background information on SCTP, with an emphasis on its multi-homing feature and how it manages network links during a connection. Chapter 3 discusses existing transport-layer mobility management solutions and in particular, describes the concept of Mobile SCTP as it has been introduced in existing literature.

The challenges facing the deployment of Mobile SCTP are identified in Chapter 4. Chapter 5 describes the experimental setup that was subsequently used to implement and test modifications to SCTP. Chapter 6 presents the proposed enhancements to SCTP, including a complete NAT traversal solution for SCTP that covers aspects from association initialization to dynamic address reconfiguration. The introduction of a novel feature in SCTP known as source address selection, which would improve fault resilience in asymmetric topologies, is also discussed. The implementations of a mobility management solution based on the modified SCTP, as well as the results of the field trials, are presented in Chapter 7. Chapter 8 extends the source address selection feature to SCTP acknowledgment chunks, eliminating a critical weakness of SCTP known as single point of failure in multi-homing networks. Finally, Chapter 9 concludes this dissertation, with a discussion on future steps to maximise the mobility experience for the end-user. The program codes and references are provided in the accompanying CD.

Chapter 2 A brief review of SCTP

2.1 Birth of SCTP

The Stream Control Transmission Protocol (SCTP) was designed by the Internet Engineering Task Force (IETF) Signalling Transport Working Group (SIGTRAN) to replace the lower layers, Message Transfer Part (MTP) 1-3, of the SS7 signalling protocol. However, the Internet Engineering Steering Group (IESG) has since decided that the resulting protocol is robust enough to be elevated from a specialized transport for telephony signalling to a new general-purpose transport protocol to stand alongside the User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP). To this end, SCTP has been moved from SIGTRAN to the general Transport Area Working Group (TSVWG). A detailed description of SCTP can be found in [5] or RFC 2960 [6].

2.2 SCTP in wireless environments

Reliable transport protocols such as TCP have been tuned for traditional networks made up of wired links and stationary hosts. It is well known that TCP performs well in such networks by adapting to end-to-end delays and packet losses caused by congestion [14]. It provides reliability by maintaining a running average of estimated round-trip delay and mean deviation, and by re-transmitting any packet whose acknowledgment is not received within four times the deviation from the average. Due to the relatively low bit-error rates over wired networks, all packet losses are assumed to be due to congestion alone.

In the presence of high error rates and intermittent connectivity characteristic of wireless links however, TCP reacts to packet losses as it would in the wired environment: it drops its transmission window size before retransmitting packets, initiates congestion control or avoidance mechanisms. These measures result in an unnecessary reduction in the link's bandwidth utilization, thereby causing a significant degradation in performance in the form of poor throughput and very high interactive delays [15][16]. Simulation results presented in [17] have shown that the main cause of TCP performance degradation on a lossy link is indeed the congestion algorithm.

SCTP distinguishes itself from TCP by its capacity to maintain multiple streams of messages inside a single association. Multi-streaming allows the sequence of messages to be maintained within each stream independently of the others. In single-stream transport, if a packet is lost, packets following the lost one will be stored in the receiver's buffer until the lost packet is retransmitted from the source. This is known as head-of-line blocking in TCP where only one stream carries data. In SCTP, multiple-stream transport ensures that data from the alternative streams can still be passed to upper layer applications. According to an article by Sun Microsystems on 4G wireless systems [7], this makes SCTP ideally suited for connecting and monitoring wireless cell phone and Internet appliances. SCTP's multi-streaming feature has already been evaluated over satellite links in work supported by NASA with encouraging results [8].

SCTP additionally supports multi-homing. Hosts are able to take advantage of multiple network layer addresses belonging to different network interfaces during an association. This means that peers are able to make use of logically distinct paths to reach the host. Furthermore, the validity of each address is actively monitored by endpoints which regularly send out message chunks known as HEARTBEATs.

Failures or losses of sessions during an active association can thus be instantly detected.

SCTP multi-homing capabilities were primarily designed for failover error resilience. In the case where the primary network address fails, or when the upper layer application explicitly requests the use of the backup, data is channelled to other available network addresses. This is in contrast to a TCP connection where the endpoints may not change their IP addresses without restarting the connection.

The authors of SCTP have further proposed an extension to SCTP known as Dynamic Address Reconfiguration (ASCONF) [9]. With this enhancement, hosts are able to constantly update their list of available addresses while on the move. This makes SCTP particularly suitable for mobile clients that roam between heterogeneous IP networks, e.g. WLAN and GPRS. The above advantages of SCTP make it a promising candidate for transport-layer mobility management.

2.3 Association Initialization

An established connection between two endpoints is known as an association in SCTP terminology. This association is established after a four-way handshake between the two endpoints which exchange the following chunks:

1. INIT

The host sends an INIT chunk requesting an association to its peer. In this chunk, the host includes a list of its IP addresses that it wishes to include in the association. The basic structure of an INIT chunk is shown in Figure 2.1.

Chunk Code INIT	Chunk Flags = 0x00	Chunk Length
Initiation Tag		
Advertised receive window credit		
Outbound streams	Maximum inbound streams	
Initial Transmission Sequence Number		
Optional Parameter #1 (e.g. IP addresses)		
…		
Optional Parameter #N		

Figure 2.1 Basic structure of an INIT chunk

2. INIT-ACK

The peer stores the INIT chunk received but does not process it. It responds with an INIT-ACK chunk, including a list of its own available IP addresses for the association. A unique cookie is also embedded in this chunk. The basic structure of an INIT-ACK chunk is shown in Figure 2.2.

Chunk Code INIT-ACK	Chunk Flags = 0x00	Chunk Length
Initiation Tag		
Advertised receive window credit		
Outbound streams	Maximum inbound streams	
Initial Transmission Sequence Number		
State cookie		
Optional Parameter #1 (e.g. IP addresses)		
…		
Optional Parameter #N		

Figure 2.2 Basic structure of an INIT-ACK chunk

3. COOKIE ECHO

The requesting host echoes the cookie to the peer. This chunk may be bundled with user data.

4. COOKIE ECHO-ACK

If the received echoed cookie is correct, the peer processes the parameters found in the initial INIT chunk. When the association is successfully configured, the peer returns a COOKIE ECHO-ACK chunk to the requesting host indicating that the association is up.

2.4 Managing multiple addresses within an association

When the association is established, each endpoint has a list of destination network layer addresses belonging to its peer. When a host receives an SCTP packet, it identifies the association by the destination IP address as well as the destination port number.

One of the IP addresses of the peer is known as the primary destination or primary path, i.e. packets sent to the peer are sent to this destination address by default.

During retransmission, however, alternative destination addresses are used. The retransmission algorithm is explained in [6]. When the host's retransmission timer is expired, the host resends the same packet through the alternative path. The host proceeds to send packets with the next sequence number through the primary path and sets the retransmission timer to double the previous value. This is known as binary backoff. Retransmissions are repeated a pre-defined number of times before the host completely switches from the primary path to the alternative path. The default value of this predefined value is five, and the initial timeout is one second. Thus the time required for an endpoint to completely switch path, or destination address, is 63s. This is shown in [13].

To monitor the validity of inactive destination addresses, HEARTBEAT chunks are sent at regular intervals. If a HEARTBEAT-ACK chunk is not received within a specified timeout, the destination address in question will be marked 'unreachable' and will not be used by the host, even during retransmission.

2.4.1 Dynamic Address Reconfiguration (ASCONF)

The ASCONF extension provides SCTP with the ability to reconfigure IP address information on an existing association and set the remote primary path. This allows the graceful addition or removal of network interfaces to an existing association, and a means for an endpoint to indicate to its peer which of its interfaces it prefers for receiving SCTP packets.

A host requesting an address reconfiguration typically constructs and sends a chunk as follows:

Chunk Code ASCONF	Chunk Flags = 0x00	Chunk Length
Serial Number		
Address Parameter (any EXISTING address in association)		
ASCONF Request Parameter #1		
….		
ASCONF Request Parameter #N		

Figure 2.3 Basic Structure of an ASCONF chunk

Upon receiving an ASCONF chunk, the peer uses the source address in the IP header and the port number in the transport header to identify the association that the chunk belongs to. If it is unable to find an association, it uses the Address Parameter

which contains an existing address in the association. If it still fails to find the association, it regards this as an Out Of the Blue chunk and replies with an ABORT chunk, as explained in Section 2.5.

If it finds an association, it processes the ASCONF request parameters, which may specify the:

1. Addition of an address (ADDIP),
2. Deletion of an address (DELIP), or
3. Setting of an address as the preferred address for receiving packets. If this address already exists in the association, the peer will use this as the primary destination address when sending packets to the requesting host. (SETREMPRI)

These parameters take the general form as shown in Figure 2.4.

Parameter Type ADDIP / DELIP / SETREMPRI	Parameter Length
Address to be added / deleted / set as primary destination	

Figure 2.4 Parameter embedded within ASCONF chunk

If the peer is able to process all the ASCONF parameters without error, it replies with an acknowledgment ASCONF-ACK chunk, as shown in Figure 2.5.

Chunk Code ASCONF-ACK	Chunk Flags = 0x00	Chunk Length
Serial Number		
ASCONF Parameter Response #1		
….		
ASCONF Parameter Response #N		

Figure 2.5 Basic structure of an ASCONF-ACK chunk

A complete success is indicated by returning only the chunk header without any parameter responses. The peer can also indicate individual successes and errors in each parameter response.

ASCONF request and acknowledgment procedures take place via an existing address in the association. When an addition is requested, the receiving peer may use the address being added as a destination for its packets once it has processed the request parameter. However, the sender may only use the address as a source when it has received a successful ASCONF-ACK.

Once the address has been deleted from the association, the peer would be unable to identify the association to which a packet sent from the deleted address belongs. Thus, the requesting host does not send out packets from interfaces that are being deleted from the association if delete requests are pending.

The ASCONF extension would enable mobile hosts to dynamically switch between network interfaces and/or network addresses while on the move.

2.5 Termination of an association

A SCTP association may be terminated because of several reasons.

1. SHUTDOWN

The host sends a SHUTDOWN chunk to its peer, indicating that it would like to bring the association to a close. The peer typically responds with a SHUTDOWN-ACK chunk. The host that requested the shutdown then responds with a SHUTDOWN COMPLETE chunk to indicate completion of the shutdown process.

2. ABORT

A host classifies a packet as Out-of-the-Blue (OOTB) when the source IP address and the source port number of the packet does not match any of its known peers. Several rules are defined in [6] for the handling of special OOTB packets (e.g. those containing INIT or SHUTDOWN chunks) but in general, a receiver of an OOTB packet will respond to the sender with an ABORT chunk. When the ABORT chunk is received, the host moves the association into the closed state.

Chapter 3 Literature Review

3.1 Introduction

Mobility may be defined as the ability to change points of attachment to the network during an active connection. This may be provided at the link layer as in GPRS and WLAN or at the network layer as in Mobile IP. In this chapter, we review existing literature on mobility management and in particular, ongoing developments of SCTP as an end-to-end transport-layer solution.

3.2 Link-layer mobility

Link-layer mobility allows the user to change its points of attachment to a network. For example, cellular phones may roam between base stations and a WLAN-enabled device may move from one access point to another. Link-layer mobility offers a user uninterrupted access to network services as he moves from one point of attachment to another. However, solutions are technology-specific and are not standardized across heterogeneous wireless networks. Furthermore, WLAN access points have limited coverage areas (e.g. 60 meters within NUS) that seldom overlap. This means that a user that starts a data transfer on WLAN has to remain with the coverage area until the transfer is complete. This restricts the user to a single network interface and a single network operator and its partners.

3.3 Network layer mobility

Mobile IP [11] provides mobility support by a system of packet forwarding. A mobile host's home address serves as a unique endpoint identifier. Mobile IP ensures the delivery of packets destined to a mobile host's home address, independent of the host's physical point of attachment, or care-of address. This is enabled by a routing tunnel between a mobile host's home network and foreign networks. However, it has been recognized that in this case, hosts undergo triangular routing, which is often longer than the optimal unicast path [18]. Although route optimization options for Mobile IP have been designed, deployment would require modifications to both the IP layer in end-hosts and the routing infrastructure in networks.

3.4 Transport layer mobility

Transport layer mobility is defined in [11] as being link layer independent in that it can support heterogeneous technologies. However, it is also link layer-aware as it tracks available link layer connections and determines which should be used for data communications. The endpoints themselves initiate handoffs, paving the way for solutions that are adaptive to end-to-end channel characteristics. Transport layer mobility facilitates seamless roaming between heterogeneous IP networks, such as IEEE 802.11 WLAN and 2.5G/3G cellular technologies.

Software support for mobility is implemented on end-hosts. However, no changes are required to the underlying IP substrate and routing architecture, keeping the packet-switched nature of the Internet intact.

3.4.1 Datagram Congestion Control Protocol (DCCP)

The Datagram Congestion Control Protocol (DCCP) is an example of a transport protocol that supports mobility. DCCP is a basic and lightweight protocol which is intended for delay-sensitive applications, such as streaming media or Internet telephony. In [19], the authors of the protocol acknowledge the advantages of developing mobility into a transport protocol rather than relying on other layers. DCCP provides support for a moving host to transfer a connection from one address to another. A host moves to acquire a new address and subsequently informs its peer of the new address. However, DCCP does not provide for soft handover mechanism and in [20], the authors note that DCCP's primitive support for mobility is intended to solve only the simplest multi-homing and mobility problems.

3.4.2 Mobile SCTP (mSCTP)

The authors of Mobile SCTP [21], [22] present SCTP together with the Dynamic Address Reconfiguration (ASCONF) extension as a transport layer mobility management solution. Mobile SCTP is targeted towards handovers in the classic client-server model, where mobile clients initiate sessions with fixed servers. SCTP, together with the ASCONF extension, has the potential to allow mobile clients to fully accomplish seamless handovers in heterogeneous IP networks. The authors suggest that client mobility management based on mSCTP seems not to require any new protocol development. However, it is found in the course of this work that several

issues remain unresolved, such as Network Address Translators (NAT) traversal issues and poor fault resilience in asymmetric multi-homing configurations.

Chapter 4 Critical weaknesses of SCTP in wireless networks

4.1 Introduction

This chapter identifies and establishes the unresolved issues in Mobile SCTP preventing its deployment in networks. This forms the basis and motivation for the subsequent work.

4.2 SCTP and Network Address Translator (NAT) Traversal

When a mobile client connects to the Internet by some wireless technology, it is typically assigned an IP address from the local address space. This may be accomplished by any of the dynamic address assignment techniques currently being used, such as Dynamic Host Configuration Protocol (DHCP) or Point-to-Point Protocol (PPP).

In order to allow these hosts with private addresses to connect to an external realm with globally unique addresses, Network Address Translators (NAT) [23] [24] are widely deployed in networks. When a packet traverses the NAT, it replaces the source address in the IP header and the source port number in the transport header of the outgoing packet with the globally valid IP address of the NAT and a port number that it randomly assigns to the request.

The following figure shows the basic structure of a typical SCTP chunk, which is similar to that of a TCP or UDP packet.

IP header	Source address	Destination address	…
Transport header	Source port	Destination port	…
SCTP packet data	Chunk type	Chunk length	…

Figure 4.1 Basic structure of a data packet

Basic address and port replacement by a NAT is shown in Figure 4.2.

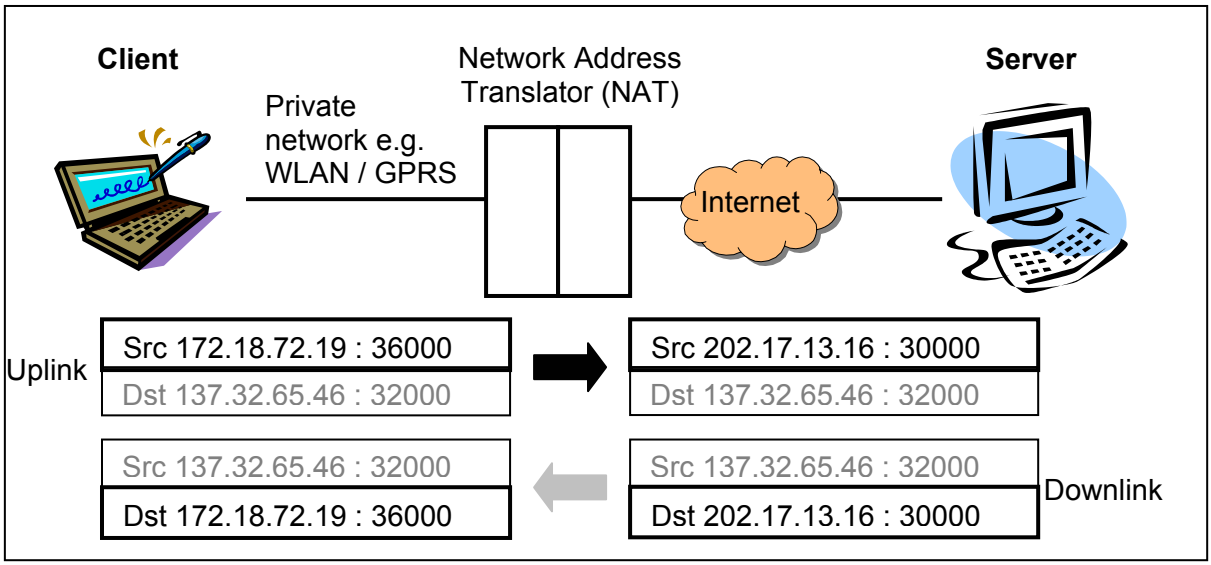

Figure 4.2 Function of Network Address Translator

When the peer responds to the host, it addresses the packet to the NAT. The NAT identifies the true destination of the packets by way of the destination port number in the transport header, which it earlier assigned to the host initiating the request. After replacing the destination address in the IP header and destination port number in the transport header with those of the intended destination host in its network, it forwards the packet accordingly. This is how a host in a private network typically accesses the Internet today, be it a computer in a wired Local Area Network (LAN) or a mobile node using WLAN or GPRS technologies.

Any mobility management solution that aims to be deployable must hence support Network Address Translators. It is the main function of a NAT to replace the source and destination IP addresses found in the IP header and port numbers in the transport header respectively. However, it is unable to modify data embedded within the protocol-specific packet data unless special support, such as Application Level Gateways (ALG) is installed. For example, a customised ALG designed for the File Transfer Protocol (FTP) allows the addresses embedded within FTP packets to be modified by NATs.

4.2.1 Network Address Translation

Unfortunately, SCTP has poor support for NAT traversal. Initialization procedures are successful as peers reply to the source address of the initialization request. The reply traverses the NAT and the IP header and transport header are appropriately translated.

However, as shown in Section 2.3, a host embeds a list of its available network addresses in SCTP initialization chunks. Private network addresses are leaked to hosts in the public realm. The server attributes both the public address of the NAT in the IP header and a private address in the INIT chunk to the host. It thinks that the client disposes of more than one valid network address, when two of them refer to the same interface. When the server attempts to reach to the host using an invalid private addresses, it may attain an internal peer if a separate private addressing system exists within the server's internal network. The internal peer would reply with an ABORT, causing the association to break down. An example is shown in Figure 4.3.

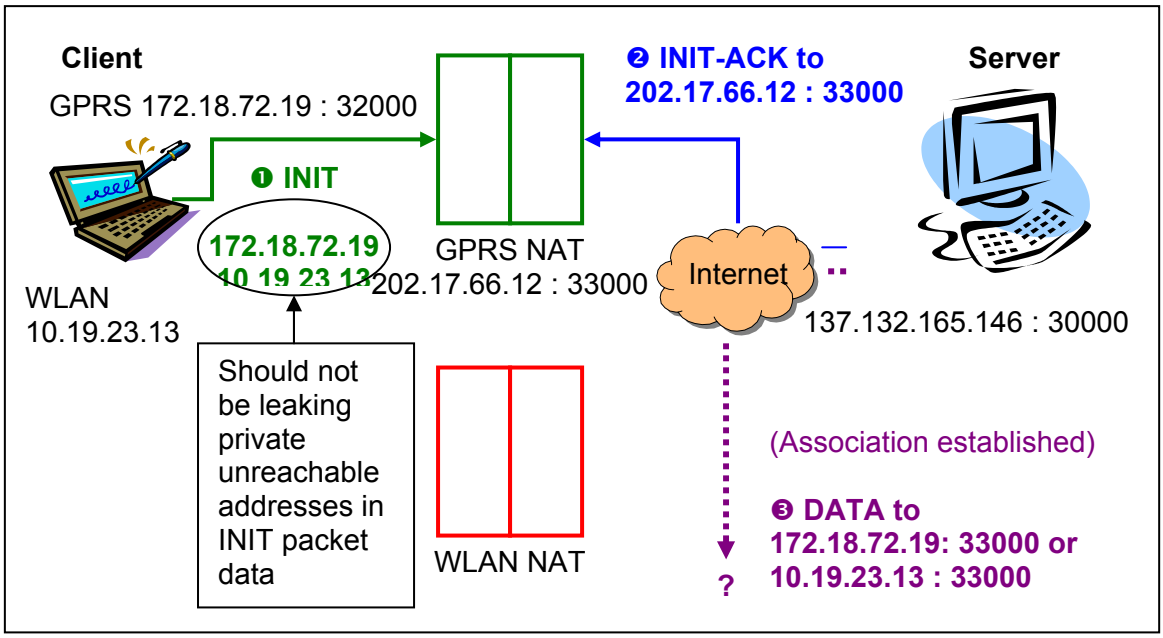

Figure 4.3 Leakage of private network address during SCTP association initialization by multi-homed host

This has been identified as a "black-hole condition" in SCTP [25]. SCTP and NAT issues have only been partially addressed in RFC3257 [26]. In the case of a single-homed host with only one IP address available, such as in Figure 4.2, the authors suggest that no transport addresses be sent in the INIT or INIT-ACK chunk. This forces the peer to use the source address in the IP header, which would have been replaced by the NAT's IP address, as the only destination address for this association. However, it is admitted that any multi-homing configuration that is behind the NAT would not be visible to the peer and thus not be taken advantage of. This is shown in Figure 4.4.

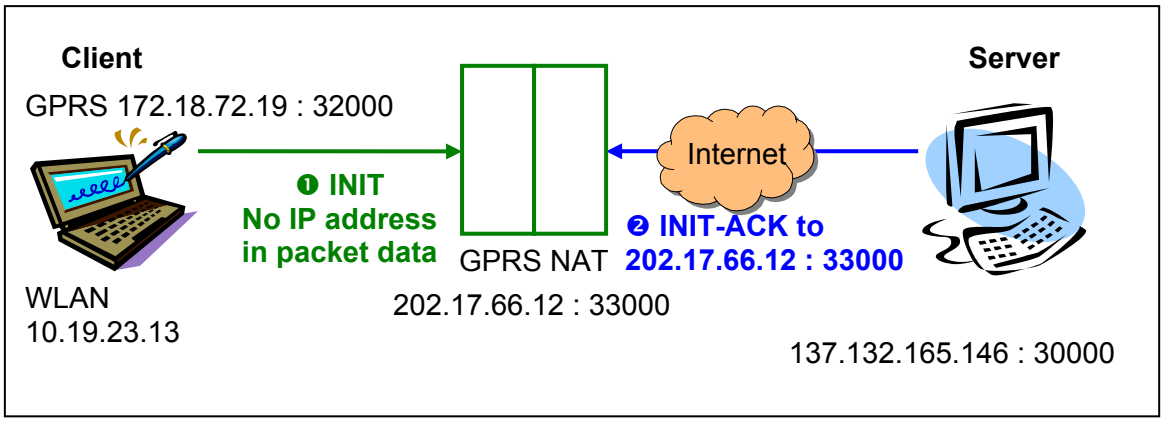

Figure 4.4 Restricting a multi-homed host to using only one interface

In order to use the multi-homing configuration, it is recommended in [26] that the NAT should have a public IP address for each represented internal IP address. However, the primary objective of a NAT is to allow numerous hosts with private network addresses to connect to the Internet using the common public address of the NAT and it would be inefficient to assign as many public IP addresses as the number of hosts behind it.

Furthermore, the NATs would require Application Layer Gateway (ALG) to intelligently translate the private IP addresses embedded within SCTP chunks. This solution would require the development of customised ALGs, such as what has been developed for FTP, and is not easily deployable.

Another alternative suggested in [26] is to use the hostname feature and Domain Name Search (DNS) to resolve the addresses. The hostname, which is included in the INIT or the INIT-ACK of the association, is resolved by DNS before the association is completely set up. However, there are outstanding issues regarding NAT and DNS [27]. Furthermore, not all hosts may have a DNS hostname.

As shown in Section 2.4.1, Dynamic Address Reconfiguration (ASCONF) procedures take place via an existing address in the association. The ASCONF

extension is defined such that a host embeds its network addresses within the request parameters in the ASCONF chunk, as shown in Figure 4.5.

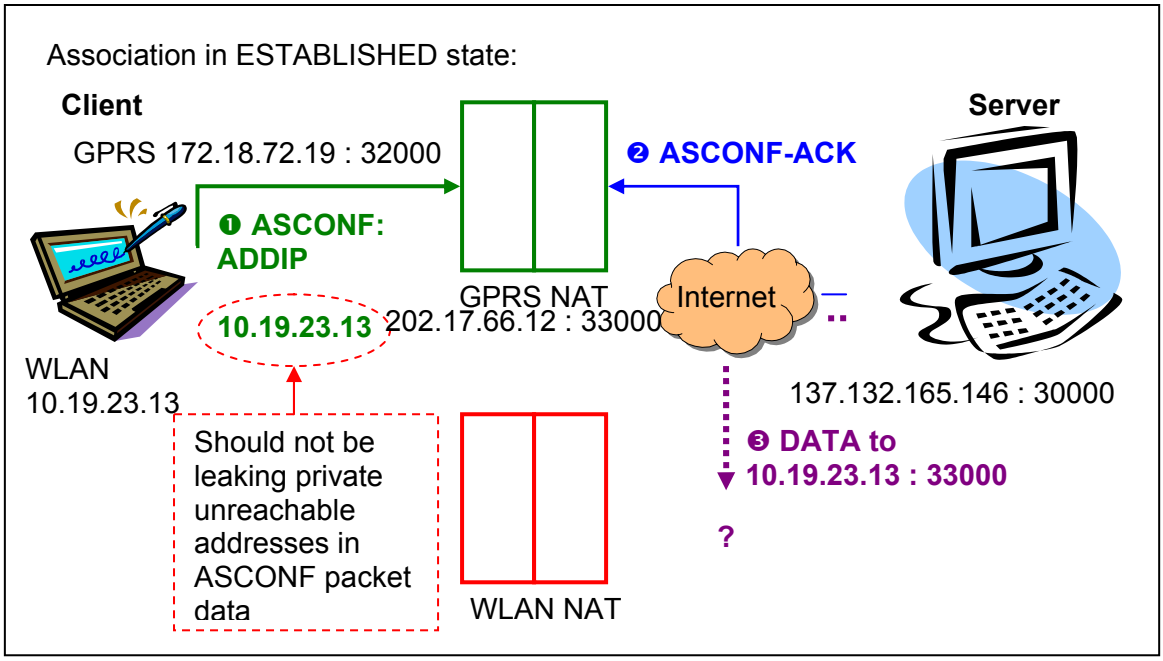

Figure 4.5 Leakage of private network addresses during address reconfiguration

In Figure 4.5, let us assume that the host initiates the association using the GPRS network and roams into a WLAN hotspot. By definition, the request to add the IP address of the WLAN interface is sent from the GPRS interface, i.e. the address that already belongs to the association. Here again, the association would break down if DATA or HEARTBEAT chunks are sent to the invalid private network address. No viable solutions have been suggested in [9], [21], [22] or [26] to allow multi-homed hosts in heterogeneous networks to reliably perform Dynamic Address Reconfiguration procedures.

4.2.2 Port Translation

When a host is behind different NATs on distinct networks, the NATs in each network work independently of each other. For example, when a host that is dual-homed in GPRS and WLAN networks sends a packet via the WLAN interface, the NAT in the WLAN network randomly assigns the request an available port number to the request. Similarly, when the host sends a packet via the GPRS interface, the NAT installed in the GPRS network randomly assigns the request one of its available port numbers. The same host would thus present two different port numbers to its peers.

An SCTP association is defined by the network addresses and a destination port number belonging to the connected peer. In [9], it is stated that this port number should not change throughout the association.

However, as there is no known protocol that would allow two NATs in different networks to communicate with one another, it would be highly unlikely that they reserve the same port number for the host. In the case of a mobile host, it is highly likely that the host initiates a connection in one wireless network and subsequently moves into the coverage area of a second network. The port number that is reserved for the host on the first NAT is unlikely to be available on the second NAT. Even it were available, it would not be easily reserved as port assignments are random on session demand. Thus there is merit in reviewing the assumption that a host in a SCTP association has a destination port number that does not change throughout the association.

4.3 Fault resilience in asymmetric multi-homing topologies

It has been identified in [1] that an "ideal" multi-homing configuration in SCTP is one that maximises path diversity, i.e. every destination address of a peer results in a distinct, separate path towards the peer. The underlying assumption is that both hosts include an equal number of IP addresses in the association. An example of this is a symmetric two-two topology shown in Figure 4.6.

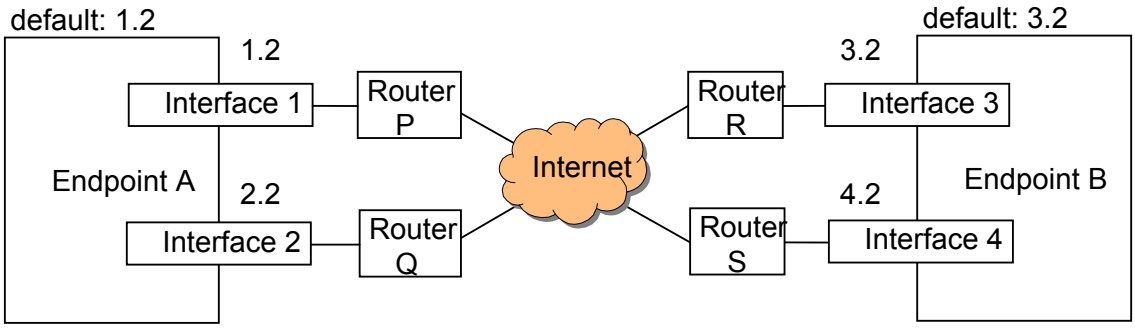

Figure 4.6 Symmetric two-two topology

Routing tables may be set up such that packets destined for 3.2 of Endpoint B are sent exclusively via Interface 1 and vice versa. Similarly, packets addressed to 4.2 of Endpoint B are sent exclusively via Interface 2 and vice versa. In a symmetric topology, this ensures that if the transport of data to one of the destination addresses fails, the sender has alternate paths to the peer, as shown in Figure 4.7.

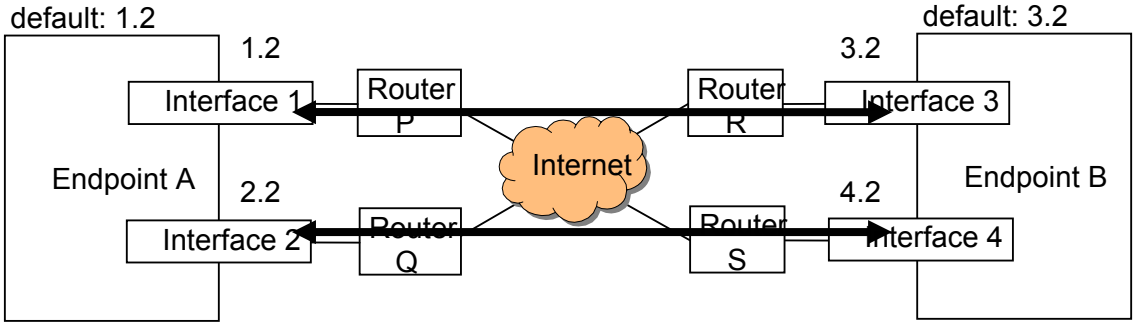

Figure 4.7 Path diversity in a symmetric two-two topology

This requirement of configuring routing tables arises because SCTP is unable to select the source address of outgoing packets. SCTP relies on IPv4 routing mechanisms that are destination-based. As such, the multiple paths that guarantee a SCTP association can only be provided if there is proper configuration at the end-hosts. The authors of [5] admit, however, that whether a symmetric topology can be achieved in practice depends on a number of factors, including path diversity, multiple connectivity and the routing protocols that glue the Internet together.

A more common network configuration is an asymmetric multi-homing configuration, such as that found in Figure 4.8. This is a likely scenario in mobile networks, where mobile clients with multiple network links and limited bandwidth connect to fixed servers that often have only one high-speed link.

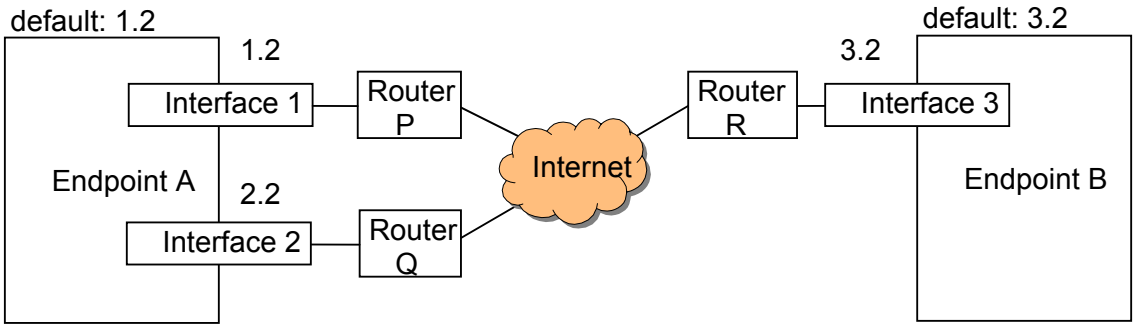

Figure 4.8 Asymmetric two-one topology

In this asymmetric topology, the association would evidently fail should Interface 3 belonging to Endpoint B experience network failure. However, due to the way routing tables are inherently configured in IPv4, the association would also collapse if the networks of either Interface 1 or 2 belonging to Endpoint A cease to function.

Endpoint A typically sends all outbound traffic to Endpoint B from one of its interfaces, which has been marked as the "default outgoing interface" in Endpoint A's routing tables. Should the default interface be unable to transport data due to network failure, the routing table would not automatically switch to sending packets via its alternative interface as the endpoint has no way of detecting the failure. The routing tables would update themselves only when the default interface ceases to exist in the system, e.g. the interface is physically removed from the system, As such, although Endpoint A has two network interfaces, it only has one possible path to its peer Endpoint B.

Unfortunately, SCTP's HEARTBEAT mechanism would not remedy the situation. In Figure 4.9, we may assume that by default, outgoing traffic from Endpoint A to Endpoint B is sent from Interface 1. Endpoint B regularly sends HEARTBEAT chunks to the inactive destination address of Endpoint A, i.e. Interface 2. To reply to Endpoint B however, Endpoint A sends acknowledgment HEARTBEAT-ACK chunks from Interface 1, as per its routing tables.

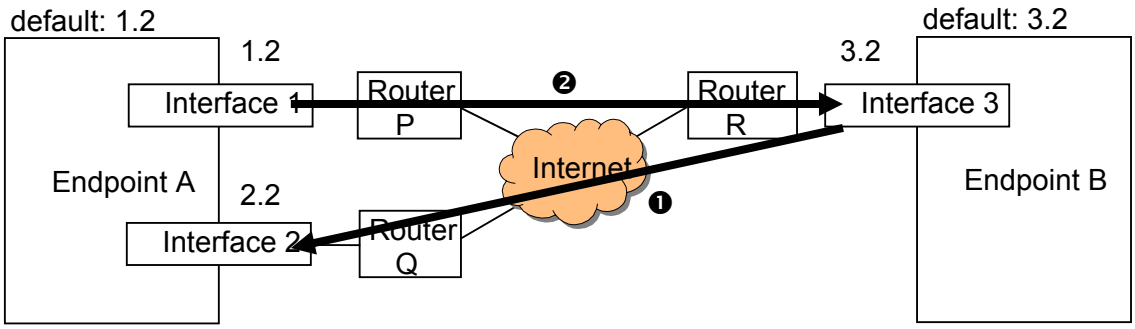

Figure 4.9 HEARTBEAT mechanism in two-one topology

Even if the network of Interface 1 fails, Endpoint A would persist in sending data chunks from Interface 1. Endpoint B would no longer receive any data from Endpoint A, including HEARTBEAT-ACK chunks. After a pre-specified timeout,

Endpoint B would mark Interface 2 of Endpoint A as 'unreachable'. As data leaving Endpoint A does not migrate to the alternative path available, the association ceases to be active.

This has been identified in [28] as a critical weakness of SCTP known as single point of failure in multi-homing networks. Because of the single point of failure, multi-homed hosts are unable to take advantage of their alternative paths for fault resilience. This is largely due to the fact that SCTP protocol designers have chosen to not to interfere with routing mechanisms in the IP network layer.

In [5], the designers of SCTP suggest that in order to avoid association collapse, a single-homed host should assign a second network address to its sole interface. With this additional address, the routing tables may be changed to provide an alternative path from Endpoint A to Endpoint B, as shown in Figure 4.10.

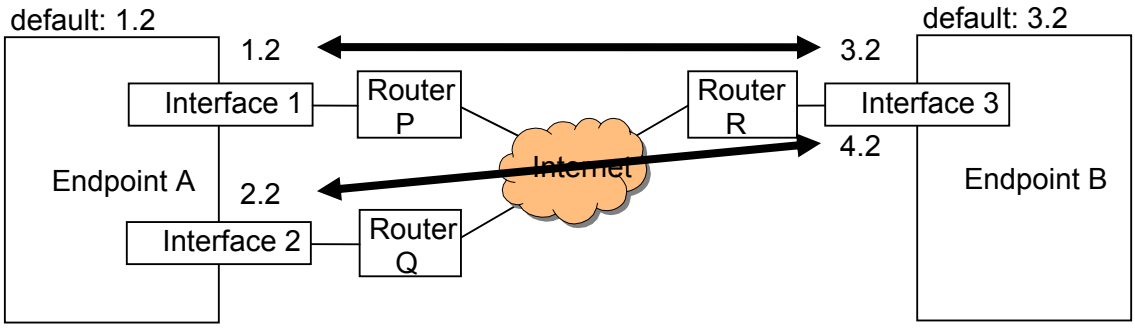

Figure 4.10 Assignment of a second network address to single-homed Endpoint B

In [22], the authors of the Mobile SCTP (mSCTP) mention this solution as a factor for consideration in mSCTP handover. It would be impractical and inefficient, however, to expect that additional network addresses from the limited address space of IPv4 may be readily assigned to hosts.

Alternatively, the authors of [13] propose a path-management solution, which employs the Source address-oriented Traffic Arrangement Router (STAR) system that

provides multiple routing tables. They also propose modifying the data structures within the SCTP transport protocol, to extend its destination-based route management system to a path-based one consisting of a source and destination pair. As shown in Section 2.4, the retransmission timeout for path switching is by default 63s in SCTP. It is shown that when one network link of a multi-homed host in an asymmetric topology is cut off, the proposed modifications allow the host to switch to its alternative path after this timeout.

However, the timeout of 63s would be too long for hosts with high mobility. In order to achieve seamless mobility, mobile clients should be able to switch to their alternative network links while on the move, and perform soft handover immediately when the quality of one of the links is deteriorating.

4.4 Summary of problem statement

In previous chapters, it has been shown that SCTP is advantageous for wireless networks because of its multi-homing and multi-streaming features. As mentioned in Section 3.4.2, the authors of Mobile SCTP feel a transport-layer mobility management solution may be fully achieved with SCTP and the Dynamic Address Reconfiguration (ASCONF) extension and that no new protocol development is needed.

In this chapter, it has been shown however that Mobile SCTP fails to address a number of key issues that would impede its deployment in mobile networks, namely

- Network Address Translator traversal, and
- Poor fault resilience in asymmetric multi-homing configurations

After establishing these weaknesses, the motivation behind subsequent work is to define a complete mobility management solution based on SCTP and the ASCONF extension that would be rapidly deployable in current network architectures.

Chapter 5 Experimental scenario

5.1 Introduction

This chapter presents the experimental scenario in which all the proposed enhancements described in the following chapter were implemented and extensively tested. First, the mobile multi-homed testbed is described. The SCTP implementations that are available for download are briefly outlined and the decision to use the SCTP Reference Implementation is justified. Next, the development of a UDP encapsulation tool, which allows SCTP packets to be transported via existing networks, is described.

5.2 Wireless multi-homed testbed

The experimental testbed consists of an asymmetric client-server configuration. The server is single-homed and the client is multi-homed across heterogeneous networks. A basic schematic of the network configuration is shown in Figure 5.1.

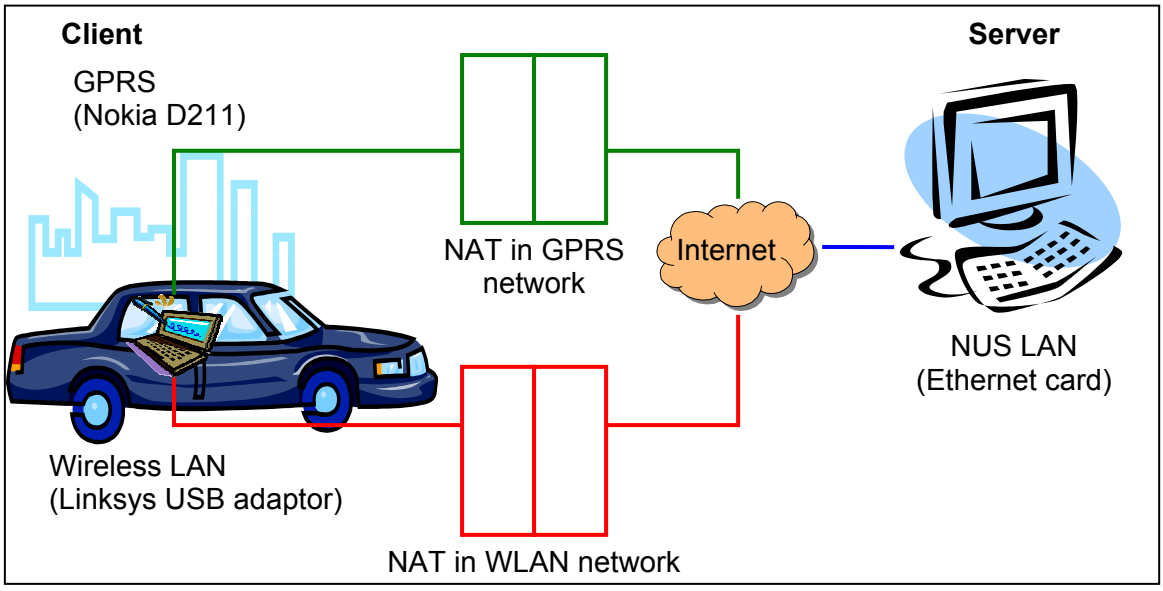

Figure 5.1. Experimental network configuration

The desktop server runs Redhat Linux 8.0 (kernel 2.4.18-20) and is connected the Local Area Network (LAN) of NUS with a fixed public IP. The client is a laptop computer running Redhat Linux 7.3 (kernel 2.4.18-3). A Nokia D211 GPRS/WLAN PCMCIA card is used to access GPRS networks. As the Nokia D211's GPRS and WLAN functionalities are unable to be activated simultaneously, a Linksys USB WLAN Network Adapter is required for WLAN connectivity, as shown in Figure 5.2.

Figure 5.2 Mobile client with Nokia D211 GPRS PCMCIA modem (left) and Linksys USB WLAN network adaptor (right)

It is desired to use existing commercial networks in our experimentation to show that the enhanced SCTP protocol can be deployed in existing networks. The GPRS network used is from MobileOne (M1). The wireless LAN network is accessed in NUS and in the vicinity of the Delifrance café in Anchorpoint Shopping Centre, which is a Singapore Telecom (Singtel) Wireless Surf Zone. According to a CNETAsia article on 14th April 2003 [29], Singapore has over 220 commercial hotspots, out of which over 150 are provided by Singapore Telecom. Under the auspices of the National University of Singapore (NUS) Wireless Information Network programme (WINZONE), there are now 350 WLAN access points installed on campus.

Figure 5.3 Existing GPRS and WLAN networks used in experiments

The client-server configuration, together with the existing GPRS and WLAN network architecture, allows for the implementation and testing of the proposed enhancements to SCTP and ASCONF described in Chapter 6. The aspect of mobility is also introduced by placing the wireless client in a vehicle. Subsequent field trials demonstrate seamless handover between heterogeneous networks without disrupting an active session.

5.3 Choice of SCTP implementation

Several SCTP implementations are available for download and testing on the Internet, and they are compatible with either the Linux or FreeBSD operating systems.

5.3.1 Linux Kernel SCTP (lksctp)

Preliminary work was carried out on the Linux Kernel SCTP Project (lksctp). The Linux kernel project aims to port SCTP into the Linux kernel and is ongoing work [30]. This required an upgrade to development versions of the kernel, namely 2.5.x, which were unstable and largely unsupported. For example, the driver distributed with the Nokia D211 GPRS/WLAN PCMCIA card is designed for 2.4.x kernels and is not yet available for 2.5.x kernels. Furthermore, the ASCONF extension is yet to be implemented in the lksctp project. Any modification to the protocol requires a recompilation of kernel sources, which is complex and less versatile than a user-space implementation.

5.3.2 SCTP Userspace Implementation by University of Essen, University of Applied Sciences (Germany) and Siemens AG

The SCTP Userspace Implementation [31] is a joint effort by Siemens AG and two German universities, University of Essen and University of Applied Sciences Münster. This implementation proved to be more efficient for testing purposes as the kernel does not have to be recompiled with every modification. However, the ASCONF modification has not been implemented, as the developers focus primarily on a variant of SCTP, the Partially Reliable SCTP (PR-SCTP) [32].

5.3.3 SCTP Reference Implementation

The SCTP Reference Implementation was designed by the designers of SCTP and is distributed together with [5]. It was decided to use the Reference Implementation as it is the only implementation that offers the ASCONF extension. This is a stable implementation that runs in userspace and includes simple commands that demonstrate the capabilities of SCTP. Regular updates are available on the www.sctp.org website but these are compatible only with the FreeBSD operating system.

5.4 UDP Encapsulation

The basic function of a NAT is to replace the source IP address in the IP header and the source port number in the transport header, regardless of the transport protocol used. This minimal support for transport protocols is largely protocol-independent, as transport packets have a basic structure as defined in Figure 4.1.

It is assumed in the course of this work that this basic support will be extended to SCTP packets when SCTP is deployed across networks. This has already been assumed in [26] where it is suggested that a single-homed host may initiate an association by not embedding any IP address within its packet data, thus leaving the peer to deduce the valid network address from its translated IP header and port number from the translated transport header respectively.

As one of the main objectives is to demonstrate mobility management based on SCTP in existent wireless networks, it is necessary for the Network Address Translators in these networks to dispose of this minimal support for SCTP. For this

purpose, the SCTP packets transmitted are encapsulated within the User Datagram Protocol (UDP). A special port number 9899 has been defined in [33] for the tunneling of SCTP packets over UDP.

As shown in Figure 5.4, SCTP packets are encapsulated within a UDP transport header and an IP header before being transmitted, as if the entire SCTP packet is UDP packet data. Network Address Translators of wireless networks readily support UDP and replace the source address in the IP header and the source port number in the UDP transport header accordingly.

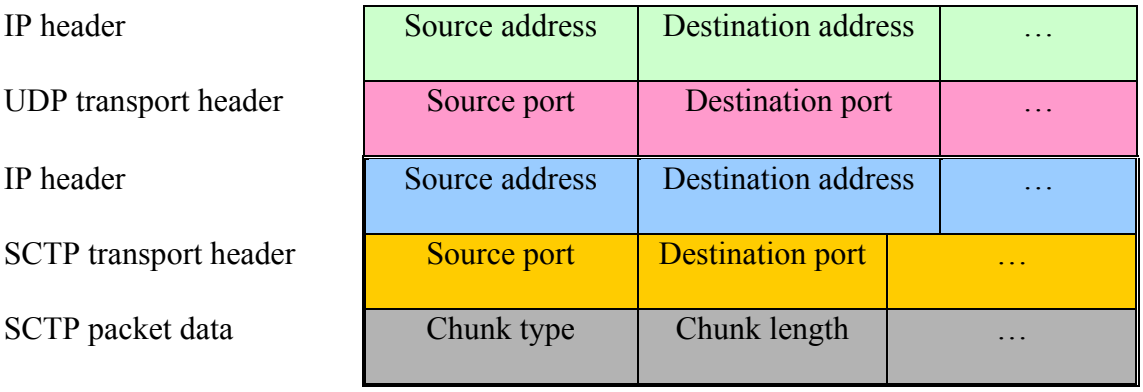

Figure 5.4 Basic structure of a UDP-encapsulated SCTP packet

Upon reception, UDP packets sent to this port number are filtered to the upper layers and the UDP encapsulation is stripped off the SCTP packet, replacing the source address in the original IP header with that found in the encapsulated IP header and the source port number in the SCTP transport header with the source port number in the UDP transport header, as shown in Figure 5.5. The packet is then sent to the SCTP implementation running in userspace.

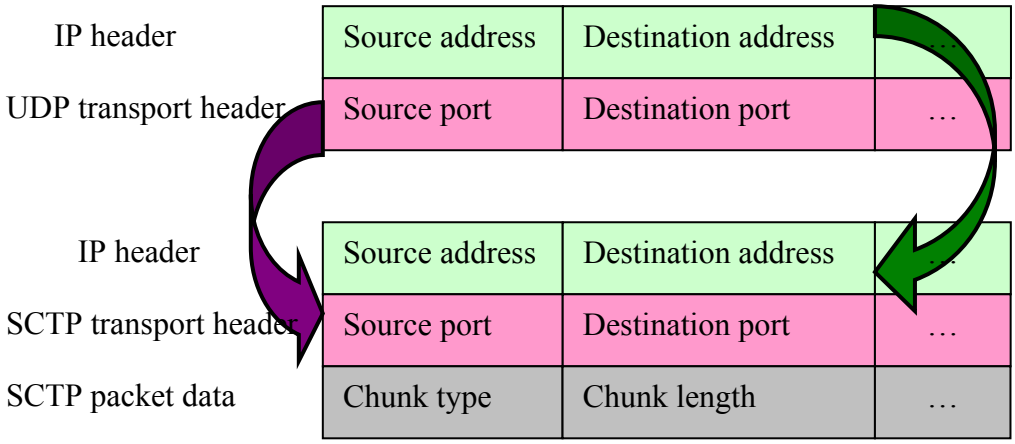

Figure 5.5 De-encapsulation of a UDP-encapsulated SCTP packet upon packet reception

This is achieved by filtering the SCTP packets before transmission and upon reception. They are then encapsulated or de-encapsulated by a program that was developed for this purpose.

With this method, the NATs in existing networks translate the IP header and the transport header of the UDP-encapsulated SCTP packets like normal UDP packets. This simulates the minimal support for address and port translation which NATs would dispose of once SCTP is widely deployed. In our field trials, the UDP encapsulation tool allows the testing of SCTP capabilities in existing networks, such as M1's GPRS network, the WLAN network within NUS and Singapore Telecom's WLAN hotspots.

5.5 Summary

In this chapter, the experimental testbed is described. It consists of a multi-homed client that has access to two wireless networks, namely M1's GPRS network and Singtel's Wireless LAN network. The server is a desktop that is assigned a fixed

IP and accesses the Internet via the LAN within NUS. This allows a trial transport-layer mobility management solution to be developed to show that SCTP, with the proposed enhancements, can be easily deployed in existing networks.

The SCTP Reference Implementation is used as it provides stable support for SCTP and Dynamic Address Reconfiguration (ASCONF) in the Linux operating system

To simulate basic support for SCTP in existing networks, a UDP encapsulation tool is developed. This allows the Network Address Translators (NAT) in all existing networks to treat the SCTP packets as UDP packets and replace the IP address in the IP header and the port number in the transport header accordingly. This simulates the minimal support for SCTP that is expected to become available once SCTP is deployed across networks.

Chapter 6 Proposed enhancements to SCTP and ASCONF

6.1 Introduction

In Chapter 4, we established the key factors impeding the deployment of SCTP in IPv4 networks. In this chapter, we discuss how with a few modifications, these weaknesses may be overcome to enable mobile multi-homed clients to use multiple network links during an active association. Firstly, the definition of an endpoint is reviewed. Next the enhancements to allow SCTP's Dynamic Address Reconfiguration (ASCONF) procedures to safely traverse Network Address Translators (NAT) are described. Finally, a novel feature that incorporates source address selection into SCTP is described. By allowing a host to select a source network interface, fault resilience in asymmetric network configurations may be improved.

6.2 Definition of an endpoint

As noted in Section 4.2, an SCTP association is an active connection between two hosts, or endpoints. These endpoints are defined by their network addresses, as well as the port numbers that have been reserved for the session. The host reaches the peer by sending packets to one of these network addresses and the associated port number.

Figure 6.1 shows a dual-homed client and a singly-homed server in an association. The client has reserved the port number 32000 for this association. The

server identifies the client and the association by the client's network addresses 137.132.165.45 and 202.165.132.25, as well as the port number 32000. Similarly, the server has reserved the port number 30000 for the association. The client recognises the server and consequently the association by the pair formed by the network address 137.132.165.146 and the port number 30000. Both client and server have globally valid public IP addresses assigned to their network interfaces.

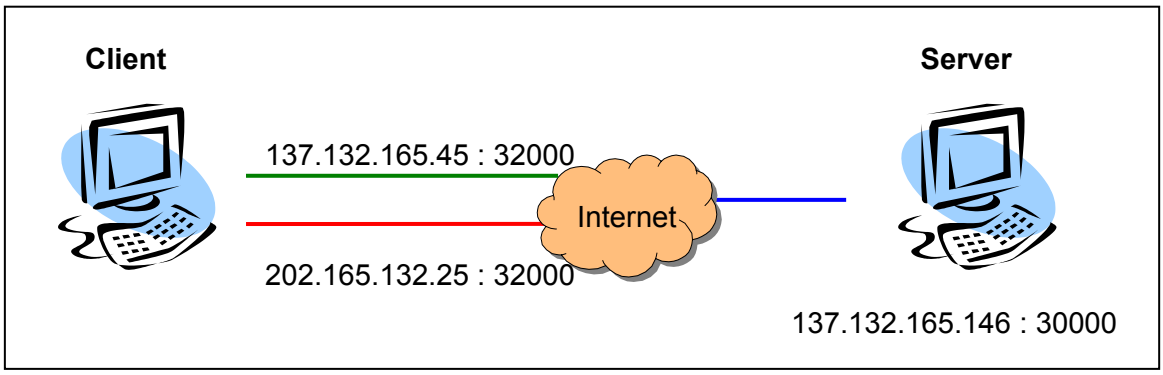

Figure 6.1 Multi-homed client and server with fixed public IP addresses

If the hosts were behind NATs, the network interfaces would be assigned private network addresses. The different NATs would assign ports to each session independently. In Figure 6.2, the client and server both fulfil the local definition of an endpoint by reserving one port number each for the association. However, the server sees the client as having multiple IP addresses and multiple ports, as the NAT in the WLAN network has reserved a different port from that assigned by the NAT in the GPRS network. This is a likely scenario in multi-homed mobile hosts.

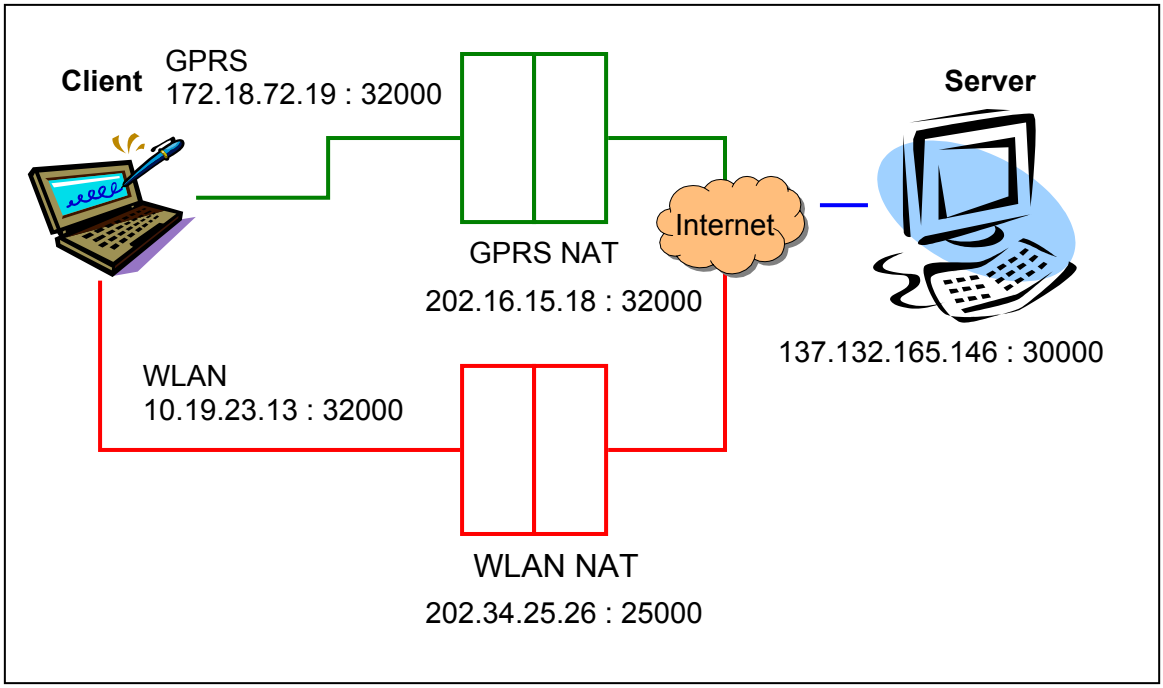

Figure 6.2 Multi-homed mobile client with private IPs

As mentioned earlier in Section 4.2.1, there is no known protocol that would allow two NATs from different networks to agree on a similar port for requests emanating from a single host. Furthermore, it is highly unlikely that both NATs randomly assign the same port number.

Allowing multi-homed mobile hosts to utilize all the links available to them implies that an endpoint may see his peer as having multiple IP-Port pairs. However, this is necessary if mobile hosts are to take advantage of the diversity of paths available to them for failover redundancy, load sharing or load balancing purposes.

6.3 Association initialization for mobile hosts behind NATs

In Section 2.3, association initialization procedures were briefly discussed. In Section 4.2, it is shown how a mobile host behind a Network Address Translator (NAT)

leaks its private network addresses during association initialization and address reconfiguration (ASCONF). As such, mobile hosts are unable to take advantage of SCTP multi-homing features.

A mobile host may only inform its peer of its valid network address by sending packets through the NAT. The User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP) are two examples of transport protocols that are supported by NATs. When a transport protocol is supported, a NAT is able to replace the source address in the IP header with its own, and replace the source port number in the transport header with one that it randomly assigns to this request. A NAT would be unable to replace any information that is found embedded within the SCTP data chunks unless special Application Level Gateways are installed.

It has previously been suggested in [26] that a host behind a NAT may carry out initialization procedures without embedding any network addresses in the INIT chunk. This allows the peer to take the source address in the IP header as the network address as well as the source port in the transport header as the sole transport address to include in the association and prevents leakage of private network addresses during the initialization of the association. However, this prevents a host from taking advantage of its multi-homed configuration, as shown in Section 4.2.1.

Rather than restricting a host that is behind a NAT to use only one of its interfaces, it is proposed that as and when other network interfaces become active, the host may add or delete the network addresses from the association, using the Dynamic Address Reconfiguration (ASCONF) extension.

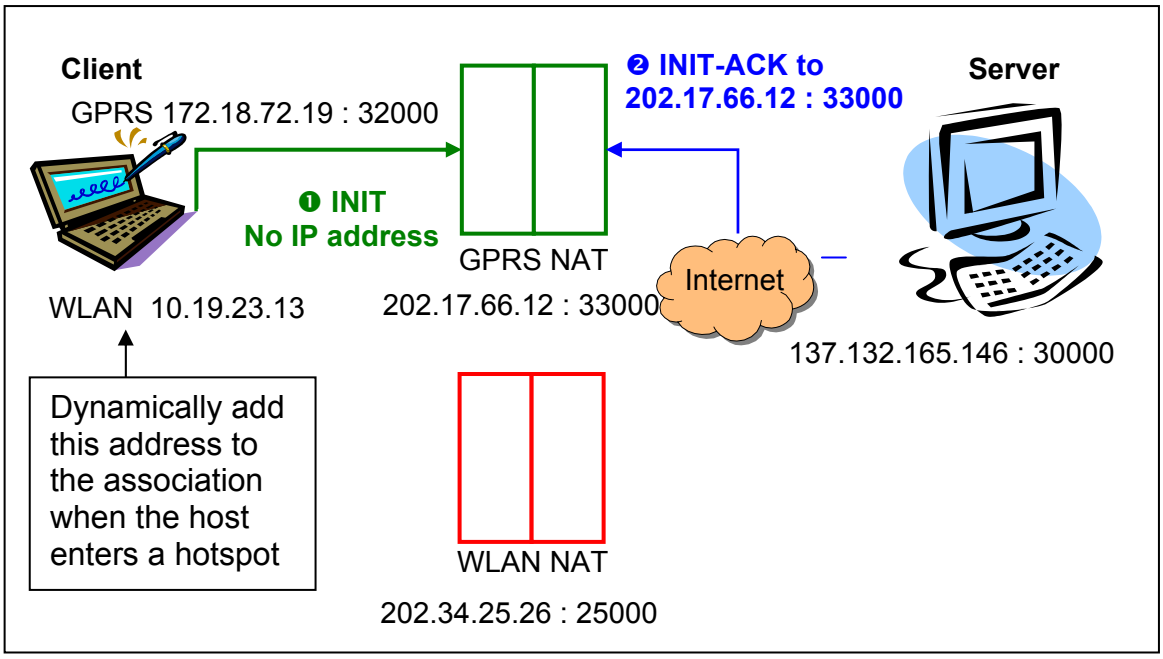

Figure 6.3 Initializing an association from behind a NAT

This assumes that the ASCONF extension may also traverse NATs successfully and is the subject of Section 6.4.

6.4 Dynamic Address Reconfiguration (ASCONF) for mobile hosts

In Section 4.2, it is shown that a multi-homed mobile host leaks private network addresses to the public address realm when it dynamically configures its network addresses during an SCTP association. This is because the dynamic address reconfiguration request is sent from a network address that already belongs to the association and the private network address is embedded in the ASCONF chunk.

A packet must traverse the NAT in order for the NAT to reserve a port number for the session and replace the private network address with its own globally valid one. For this to take place, part of the dynamic address reconfiguration request should be

sent from the address being configured. After the packet traverses the NAT, the NAT's network address and the port number reserved for the session will be transmitted to the peer as part of the IP header and the transport header respectively.

The ASCONF extension allows the host to request the

1. Dynamic addition of IP addresses to an association (ADDIP),
2. Dynamic deletion of IP addresses from an association (DELIP), and
3. Peer to use a particular IP address as the primary destination address to reach the requesting host. (SET REMOTE PRIMARY)

6.4.1 Dynamic addition of an address (ADDIP)

It is proposed that the addition of an address to an association takes place in two parts to allow for proper discovery of addresses for mobile hosts. For the addition of a valid address to the association when the host is behind a NAT, the packet requesting the addition must traverse the NAT in order for the latter to replace the source address and port with its own. Thus, it becomes necessary for part of the ASCONF to use the address being added as the source address, even if it is not part of the association as yet.

An example of how a network address may be added to the association is shown in Figure 6.4.

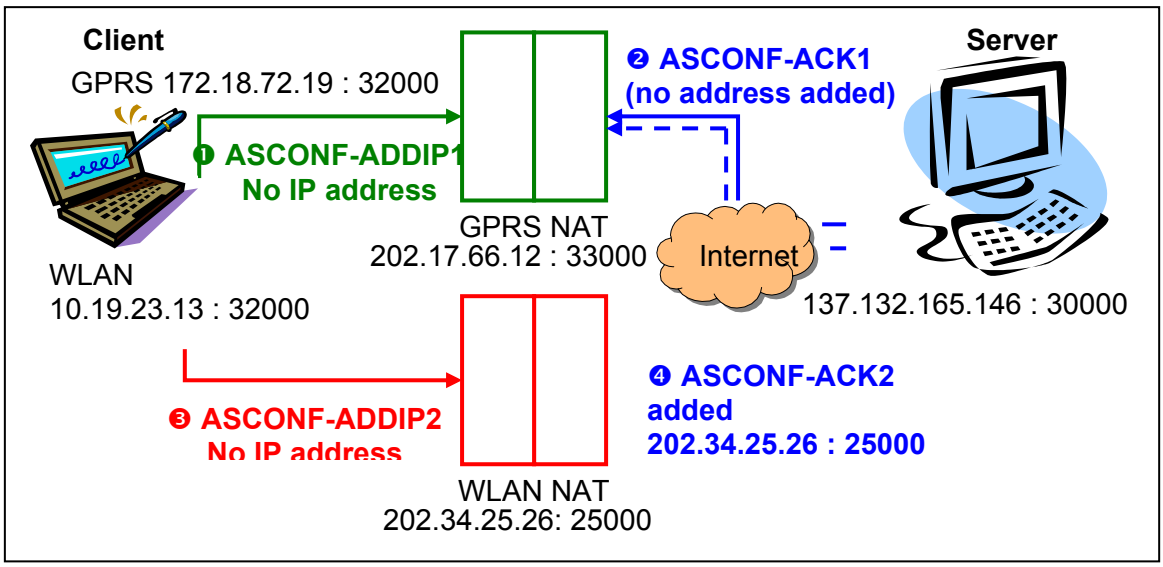

Figure 6.4 Proposed addition of a network address without leakage of private network addresses

The first ASCONF chunk in the modified ADDIP process serves to inform the peer that the sender host wishes to dynamically add an IP address to the association. This is in line with specifications in the Internet Draft on ASCONF [9] that the dynamic addition of an address takes place via an existing address in the association.

When the first ASCONF chunk arrives at the peer, it uses the source address in the IP header and the source port number in the transport header to identify the association to which the chunk belongs. As mentioned in Section 2.4.1, if it does not find the association, it refers to the network address in the Address Parameter to identify the association. In order to avoid leaking private addresses, this Address Parameter has been replaced by a random key known as the Association Key. Although the second ASCONF packet will arrive at the peer from a source address that does not as yet belong to the association, this Association key will serve to identify the association to which the chunk belongs. The amendment made to the chunk header of the first ASCONF chunk is shown in Figure 6.5.

Chunk Code ASCONF	Chunk Flags = 0x00	Chunk Length
Serial Number		
Association Key		
ASCONF Request Parameter #1		
….		
ASCONF Request Parameter #N		

Figure 6.5 Address parameter replaced by Association Key in chunk header of first ASCONF chunk during dynamic addition

The new address to be added to the association is usually embedded in the ASCONF request parameter that specifies the ADDIP request. In the first ASCONF chunk of the modified ADDIP process, no address should be sent as part of the parameter. This parameter simply indicates to the peer that an ADDIP is being requested.

Parameter Type ADDIP	Parameter Length
NULL	

Figure 6.6 Request Parameter embedded within first ASCONF chunk in dynamic addition process

If the peer has the ability to configure addresses dynamically, it should respond with a successful ASCONF-ACK. No modifications to the ASCONF-ACK chunk are required. This ASCONF-ACK chunk is sent, by default, to the primary destination of the requesting host.

Chunk Code ASCONF-ACK	Chunk Flags = 0x00	Chunk Length
Serial Number		
ASCONF Parameter Response #1		
….		
ASCONF Parameter Response #N		

Figure 6.7 ASCONF-ACK chunk

The sender, upon receiving a successful ASCONF-ACK, continues the procedure by sending the peer a second ASCONF chunk via the interface that it is requesting to be added. This ASCONF chunk would arrive bearing the source address of the NAT and the source port of the NAT in the network of the network interface being added to the association. As specified in [9], when the peer is unable to identify the association an ASCONF chunk belongs to, it refers to the Address Parameter. The Address Parameter would bear the same Association Key as previously transmitted in the first ASCONF chunk, thereby allowing the peer to identify the association. The structure of the second ASCONF chunk in the dynamic addition process is shown in Figure 6.8.

Chunk Code ASCONF	Chunk Flags = 0x01	Chunk Length
Serial Number		
Association Key		
ASCONF Parameter #1		
….		
ASCONF Parameter #N		

Figure 6.8 Second ASCONF chunk in dynamic addition process

In [9], no explicit use has been defined for the chunk flags in ASCONF chunks and they are set to 0x00. As defined in [6], these chunk flags have no meaning unless otherwise specified and in general, bear no significance for the receiver. This definition is adhered to in our implementation. The chunk flags of the ASCONF-ADDIP2 chunk are set to 0x01, in order to indicate to the sending program that the chunk should be sent from the interface being configured.

Once again, no address parameters are leaked as part of the request parameter. When the peer processes this parameter, the peer is aware that following the first ASCONF chunk requesting an ADDIP received from an address in the association, this is the second ASCONF chunk in the process. As such, the source address in the IP header and the source port number found in the transport header should be added to the association.

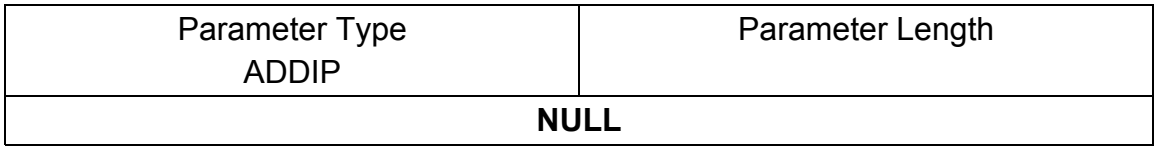

Figure 6.9 Request Parameter within second ASCONF chunk in dynamic addition process

Upon successful addition of the address and port, the peer should respond with a successful ASCONF-ACK as above. Once again, the ASCONF-ACK chunk is sent, by default, to the primary destination of the requesting host.

After the sender receives the successful ASCONF-ACK, it can proceed to fully include the address in the association and the configuration process is complete. The requesting host has included the private network address of the network interface in the association and the peer has included the public network address of the NAT as well as

the port number assigned to the session by the NAT. The peer may safely send and receive data packets from the new address that is added to the association.

6.4.2 Dynamic deletion of an address (DELIP)

As described in the previous section, a private network address may be part of an association either when the association is initialized using this address, or when it is dynamically added during the association. When the private network address is fully included in the association, the host with this network interface has added its private network address while the peer would have added the public network address and a valid port number of the NAT in the network.

When it is desired to dynamically delete a private network address from the association, an ASCONF chunk has to be sent to the peer requesting that the transport address pair consisting of the public IP address of the NAT and the corresponding port number be deleted. This packet must once again traverse the relevant NAT in order for the replacement in the IP and transport headers to take place. Unlike the ADDIP process however, only one ASCONF chunk needs to be sent. The peer would be able to identify the association as the transport address pair being deleted is already part of the association. An example is provided in Figure 6.10.

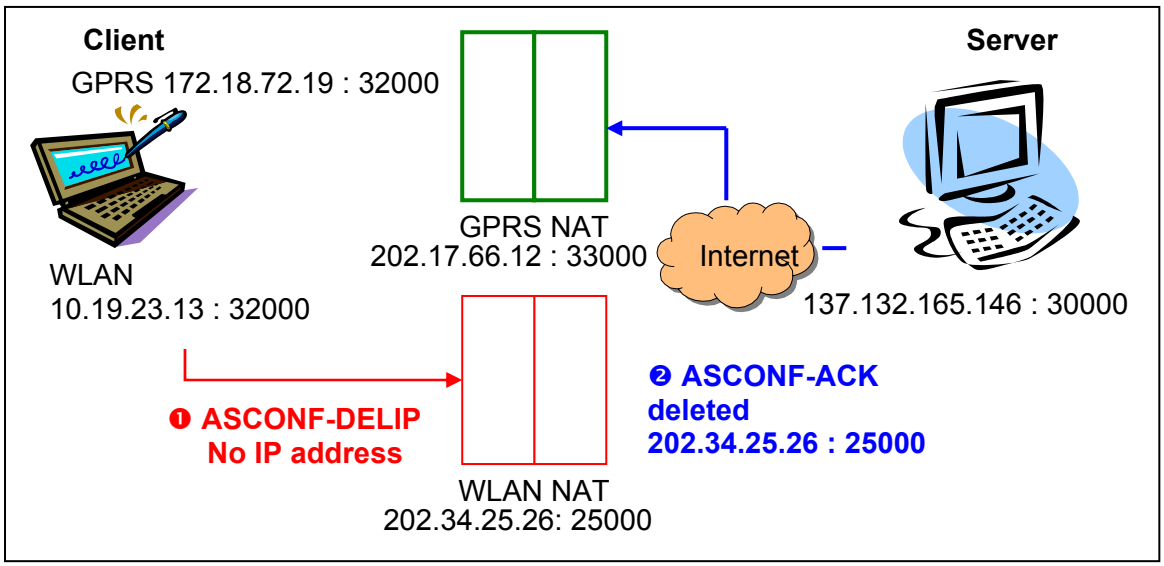

Figure 6.10 Proposed deletion of a private network address

The ASCONF chunks are similar to those defined for the dynamic addition in the previous section. An example of an ASCONF chunk with the DELIP request parameter is shown in Figure 6.11.

Chunk Code ASCONF	Chunk Flags = 0x01	Chunk Length
Serial Number		
Association Key		
Parameter Type DELIP		Parameter Length
NULL		
….		
ASCONF Request Parameter #N		

Figure 6.11 ASCONF chunk with DELIP request parameter

No private address is embedded within the Request Parameter, indicating to the server that it should delete the network address in the IP header and the port number in the transport header.

No modifications are necessary to the acknowledgment chunk, as shown in Figure 6.7. As previously shown, the acknowledgment is sent by default to the primary destination address. When the configuration is completed, the public network address of the NAT and assigned port number is deleted from the association on the server side and the client would have deleted the private network address from the association.

6.4.3 Setting of remote primary address (SET REMOTE PRIMARY)

The proposed modification to the SET REMOTE PRIMARY process is similar to that proposed for the DELIP process.

When a host requests its peer to use a particular network address, which is already included in the association as the default destination address, it usually embeds the address within the SET REMOTE PRIMARY Request Parameter of an ASCONF chunk. Instead, the host should not include any network address within the Request Parameter and send the packet from the interface being configured. In this way, the packet traverses the NAT and the appropriate address and port replacements are made. When the packet arrives at the peer, the peer knows that the address being configured is the transport address pair made up of the public IP address in the IP header and the port number in the transport header. As before, no modifications to the acknowledgment process are necessary. An example is shown in Figure 6.12.

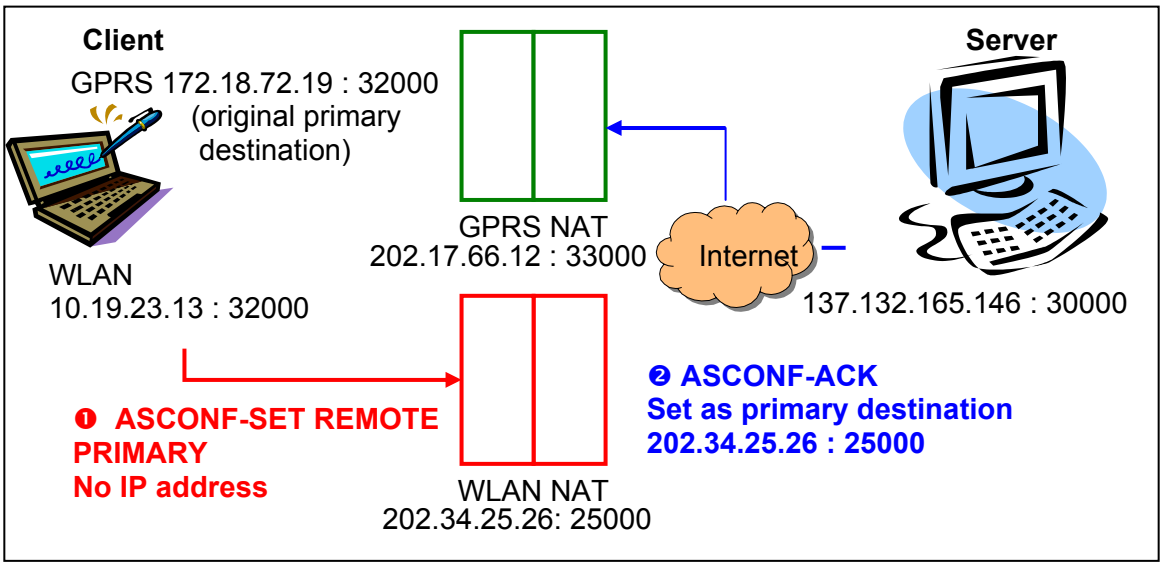

Figure 6.12 Setting a network interface with a private network address as the primary destination for receiving packets

In the above example, when the process is completed, the server will send packets to the client using the WLAN interface by default.

An example of an ASCONF chunk requesting the SET REMOTE PRIMARY chunk is shown in Figure 6.13.

Chunk Code ASCONF	Chunk Flags = 0x01	Chunk Length
Serial Number		
Association Key		
Parameter Type SET REMOTE PRIMARY		Parameter Length
NULL		
….		
ASCONF Request Parameter #N		

Figure 6.13 ASCONF chunk with SET REMOTE PRIMARY request parameter

6.5 Applicability to hosts in wired networks

It has been shown how ASCONF procedures may be modified to allow mobile hosts to take advantage of their multiple network interfaces, even when they are assigned private network addresses. This, combined with the association initialization procedures described in Section 6.2, defines a complete NAT traversal solution for SCTP and the ASCONF extension.

The proposed modifications are applicable to all hosts that desire to take advantage of a multi-homing configuration behind NATs in different networks. This scenario is most common for mobile hosts behind NATs but hosts in wired Local Area Networks (LAN) may also find these modifications useful.

The proposed modifications may even be generalized to all hosts, regardless of whether they have private or public IP addresses. For a host with globally valid IP addresses, or fixed IPs, the network address in the IP header and the port number in the transport header remain as is when the packet reaches the peer. This is a valid transport address that may be successfully configured with the proposed modifications. A small overhead is introduced, that of an additional ASCONF chunk which is required in the ADDIP procedure, but the overwhelming benefit is that a uniform transport protocol may be deployed across all networks, including IPv6 networks.

6.5.1 A special case when a network interface changes its IP address

A special case mentioned in [9] occurs when the IP address of a network interface is renumbered, e.g. certain Internet Service Providers (ISPs) rotate fixed IPs among its clients. It is recommended in [9] that the requesting host should send an

ASCONF chunk with 2 Request Parameters within the chunk: ADDIP to add the new IP and DELIP to delete the previous IP. Since the chunk will inevitably be sent using the new address, the source address and port cannot be used to identify the association. The previous IP address should be placed in the Address Parameter to enable the peer to identify the association. This has not been implemented in the SCTP Reference Implementation. However, it is desired to show that the proposed modifications would also allow dynamic address reconfiguration procedures to take place in this special case.

With the proposed modifications, the requesting host should include the fixed address being deleted as the Association Key in the first ASCONF chunk during the ADDIP procedure to allow the association to be identified. When sending the second ASCONF chunk, it includes as always the same Association Key as the first ASCONF chunk and may send both the ADDIP Request Parameter and the DELIP Request Parameter within the same chunk.

In the case of a private IP address being renumbered, the NAT's address and port in the IP header and transport header remains the same for the old and new addresses and would allow the association to be identified.

6.6 Source address selection

If a host wishes to include the network address of one of its network interfaces in an SCTP association, it has to inform the peer of the globally valid IP address of the Network Address Translator in the network, as well as the port number the NAT has reserved for the session. In order for the IP address of the NAT to be discovered and the port number to be reserved, a packet must traverse the NAT, i.e. the packet must

leave the host from the network interface that is situated behind the NAT. This would allow the NAT to fulfil its basic functions of address and port translation as earlier described and the packet would arrive at the peer with a valid source IP address and source port number.

This implies that when a sender host tries to inform its peer of a network address, the packet must leave the host from that particular network interface in order to traverse the NAT in that network. As mentioned in Section 4.3, SCTP relies on the IP layer's destination-based routing mechanism. This means that there is no mechanism within SCTP to select by which network interface a packet leaves. In order to allow hosts with private addresses to discover the IP address of the NAT and reserve a port number however, it is necessary to implement source address selection within SCTP.

In the SCTP reference implementation, packets are transmitted using the `sendto()` function. However, the Internet Draft on SCTP socket extensions [34] specifies that an application should use the `sendmsg()` and `recvmsg()` calls to transmit and receive data from its peer. This allows SCTP-specific data to be passed via ancillary data structures in the message header `struct msghdr` used in `sendmsg()` and `recvmsg()`.

```
int sendmsg (int sockfd, const struct msghdr *msg,
unsigned int options)

int recvmsg(int s, struct msghdr *message, int flags);
```

Figure 6.14 sendmsg() and recvmsg()

One additional option that may be set with `sendmsg()` is the `IP_PKTINFO` socket option in the Linux IPv4 protocol implementation. An `IP_PKTINFO` ancillary message contains the structure `in_pktinfo` that contains information about how the outgoing packet should be routed [35]. The outgoing interface is specified using its interface index. This option may be set on the raw sockets used in the SCTP Reference Implementation.

```
struct in_pktinfo
{
 unsigned int ipi_ifindex; /*Outgoing interface index*/
 struct in_addr ipi_spec_dst; /* Destination address*/
 struct in_addr ipi_addr;     /*Used in recvmsg() call*/
}
```

Figure 6.15 IP_PKTINFO ancillary data to specify the outgoing interface

The reference implementation is modified to allow the outgoing interface to be specified, if desired. This is useful for packets that need to inform the peer of a network address, such as the Dynamic Address Reconfiguration chunks. Although this may not coincide with the routing table's default network interface choice, the network interface chosen remains a valid one in the host's system.

This method of selecting the source interface uses an existing socket option within the Linux kernel IPv4 protocol implementation. By being able to select the source address, the host is able to discover the valid network address of the NAT and reserve a port number for the session. However, it is to be noted that the

implementation of the source address selection feature will be dependent on the IPv4 protocol implementation in each operating system.

In this work, source address selection using the `IP_PKTINFO` option is invoked to effect source address selection in the following scenarios:

- The host is multi-homed but not all of its addresses are in the association. The routing table might choose to send the packet via an interface that is not included in the association, which would cause the peer to abort the association. The source address selection described allows packets to be sent only via interfaces that are already part of the association.
- The host is sending a Dynamic Address Reconfiguration (ASCONF) packet, which serves to configure a network address. The packet must leave the host via the interface being configured to allow the discovery of the NAT's address and a port number to be reserved for the session.
- The application layer requests that source interface selection be applied to acknowledgment chunks to eliminate the single point of failure in asymmetric configurations. This is shown in Chapter 8.

6.7 Summary

In this chapter, it is shown how a mobile host may update its various network addresses in an active association, with a few modifications to dynamic address reconfiguration (ASCONF) procedures. These modifications are particularly important for hosts that have private network addresses, in either wired or wireless networks. The changes have been implemented and tested on a network configuration consisting of a mobile multi-homed client and a fixed singly-homed server. It is

necessary to modify SCTP to allow a source network interface to be selected. The ability to select a source address may eliminate the single point of failure in asymmetric multi-homing configurations, which is a frequent scenario when mobile clients connect to fixed servers, as it provides flexibility and improves fault resilience. This paves the way for a reliable transport layer mobility management solution based on SCTP.

Chapter 7 Achieving seamless handover among WLAN and GPRS

7.1 Introduction

The objective of this chapter is to use the modifications to SCTP that were described in Chapter 6 to demonstrate seamless handover between the WLAN network and the GPRS network.

7.2 Seamless session handover between WLAN and GPRS

The SCTP reference implementation includes the `rftp` command that performs a round-trip file transfer, i.e. the host sends a file to its peer over two streams, and the peer retransmits the file back to the host over the two streams. The `send` command sends a string of text to its peer. The mobility management application is adapted from these two commands and incorporated within the SCTP reference implementation.

The `send` command is modified to send a request for file transfer, similar to the `get` command in the File Transfer Protocol (FTP). For example, `send /home/sctp/mediumfile` requests the server to send the file `mediumfile found in the /home/sctp/` directory.

As described in Section 5.2, two network interfaces are provided to the user: GPRS and WLAN. Once a data packet is received, the application loops through all network interfaces available in the system. When a GPRS interface is available, the application automatically adds it to the association as an always-on backup using the

modified dynamic addition procedures described in Section 6.4.1. If a WLAN interface is available, the application uses the Linux Wireless Extensions [36] to obtain the signal strength on a scale of 0 to 100. If the signal strength is above a certain threshold, the WLAN interface is included in the association via the modified dynamic addition procedures. It is subsequently set as the primary network interface on which to receive packets using the modified Set Remote Primary procedures as described in Section 6.4.3. On the contrary, if the signal strength of the WLAN interface below the threshold and it is currently receiving packets on this interface, the GPRS interface is set as the primary interface. The WLAN interface is subsequently deleted from the association via the modified dynamic deletion procedures described in Section 6.4.2. This allows the user to take advantage of the higher throughput offered by WLAN whenever he is in a hotspot while maintaining an always-on backup with GPRS.

The user may initialize the association and start the file transfer on either the GPRS or the WLAN interface. If he starts the association in a hotspot, the file transfer will switch to the GPRS interface as he moves away from the hotspot. If he moves towards a hotspot during the file transfer, data transfer will switch to the WLAN once he is in the hotspot and the signal strength is above a certain threshold.

A flowchart of the application is shown in Figure 7.1.

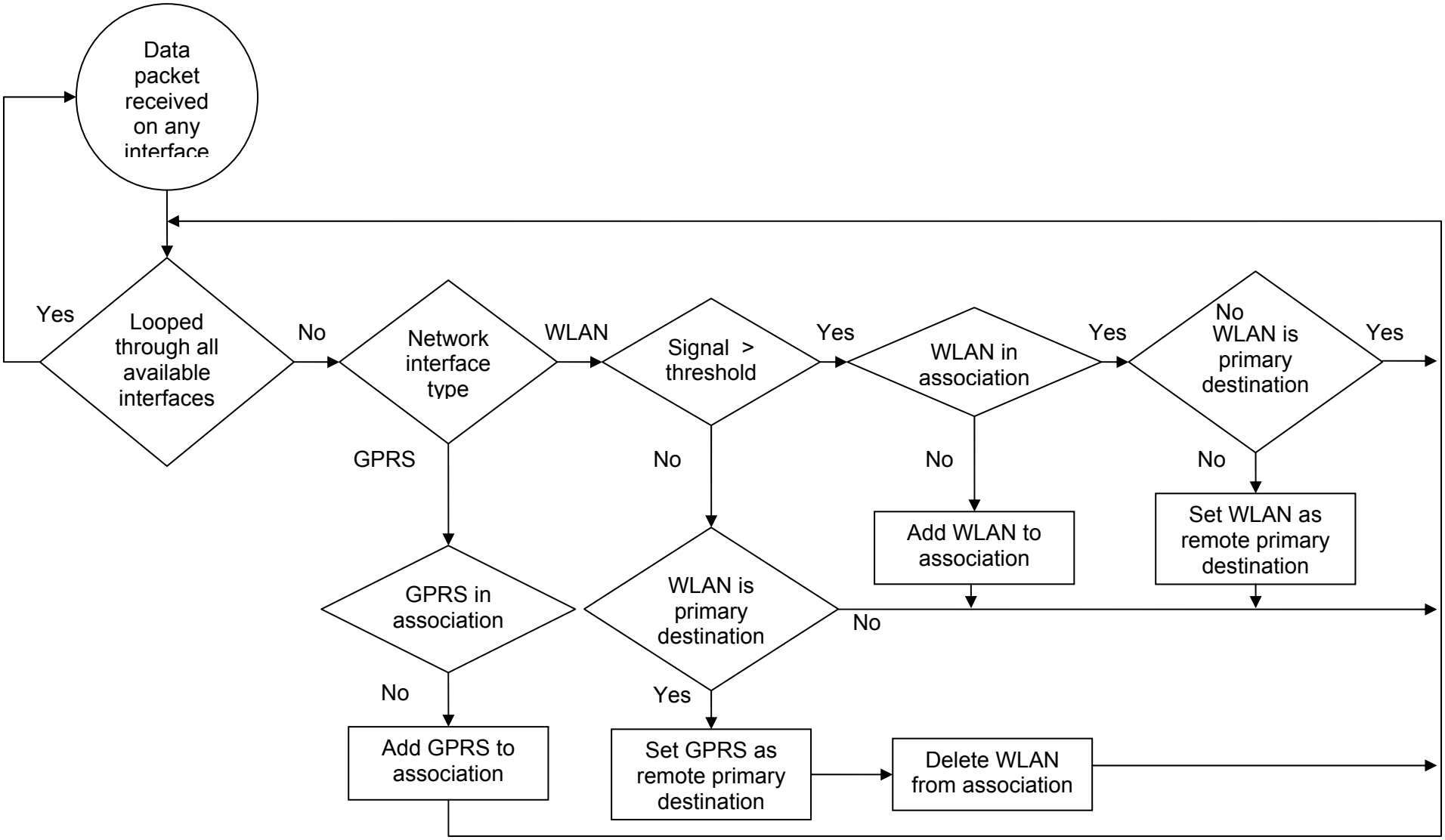

Figure 7.1 Flowchart of mobility management application

7.2.1 Test area

As described in Section 5.2, it is desired to use existing commercial networks in the field trial. The laptop was placed in a vehicle that moved in a small alley outside Delifrance café in Anchorpoint shopping centre, which is a Singtel wireless surf zone. This allows the end-user to start a file transfer using the GPRS network and switch to the WLAN interface as he approaches a hotspot. As he moves away from the hotspot, the file transfer switches back to the GPRS network, demonstrating seamless roaming between heterogeneous networks and different network operators.

Figure 7.2 Accessing M1 GPRS network and Singtel WLAN outside Delifrance

7.2.2 Threshold values

The signal strength varies from 1 to 100, where 100 is the maximum signal strength. The numerical value is not an absolute one, as different WLAN network adaptors from different manufacturers were found to give different readings at the same spot. This may be attributed to a lack of standard drivers for WLAN devices in Linux.

There is a tradeoff between the time needed for a soft handover from WLAN to GPRS when the mobile client is leaving a hotspot and the delay in switching from GPRS to WLAN when the mobile client approaches a hotspot. After surveying the signal strength values in the test area using the Linksys USB 802.11b network adaptor, signal strength of 15 was found to be a suitable threshold.

As the signal strength varies greatly even when the testbed is stationary, a time threshold had to be set to prevent oscillation and frequent switching. In this implementation, the WLAN interface is included in the association if the signal strength is found to be greater than 15 for more than 0.5s. Inversely, the GPRS interface is set as the primary destination and the WLAN interface is removed from the association if the signal strength is less than or equal to 15 for more than 2s.

7.3 Experimental Results

The user is first authenticated in the WLAN network to ensure that movement into the hotspot would initiate a seamless handover, as long as the user had not left the WLAN network within a timeout determined by the ISP. The association is first initialised in the GPRS network and a file is requested to be downloaded from the server.

Figure 7.3 shows the results of the mobility management application, with the timescale relative to the reception of the first packet by the client. Figure 7.3(a) shows the variation of signal strength with time, at regular intervals of 500ms (RSSI) as well as at the reception of each packet

(RSSI at packet reception). Figure 7.3(b) shows the signal duration against time. Uptime denotes periods when RSSI is more than the threshold of 15 while downtime denotes the reverse. As explained previously, handover from GPRS to WLAN begins when uptime exceeds 0.5s and handover from WLAN to GPRS occurs when downtime exceeds 2s in our application. The uptime and downtime counters are reset to zero if a variation contrary to the current trend is detected, i.e. when RSSI falls below 15 while uptime is increasing and if RSSI exceeds 15 when downtime is increasing. This prevents a ping-pong effect due to spurious handovers. Figure 7.3(c) shows the instantaneous throughput on the GPRS and WLAN networks, as well as the SCTP configuration procedures to accomplish handover between them. The instantaneous throughput is calculated using the formula shown below, which is commonly used in FTP applications.

$$\text{Instantaneous Throughput} = \frac{\text{Number of bytes received in data packet}}{\text{Time elapsed since last data packet received}} \qquad (7.1)$$

The file transfer begins in an area with minimal WLAN coverage, where RSSI is well below the threshold of 15 (t=0s). From t=0 to t=8s, the vehicle with the mobile client moves towards the WLAN access point at a speed of 10.9km/h until the WLAN RSSI is detected to be well above the threshold of 15. From t=8s to t=21s, the vehicle remains stationary to allow the client to take advantage of the increased throughput in the hotspot. From t=21s to t=34s, the mobile client moves away from the hotspot at a speed of 4.9km/h. It then remains stationary until the end of the file transfer at t=40s.

Procedures to include the WLAN interface in the SCTP association begin at t=4s, when the RSSI exceeds 15 for more than 500ms (uptime>0.5s). WLAN configuration requires several procedures which take a total of 13s to complete (ADDIP1-WLAN at t=4s, ADDIP1-ACK and ADDIP2-WLAN at t=9s, ADDIP2-ACK and SETREMPRI-WLAN at t=16s and SETREMPRI-ACK WLAN at t=17s).

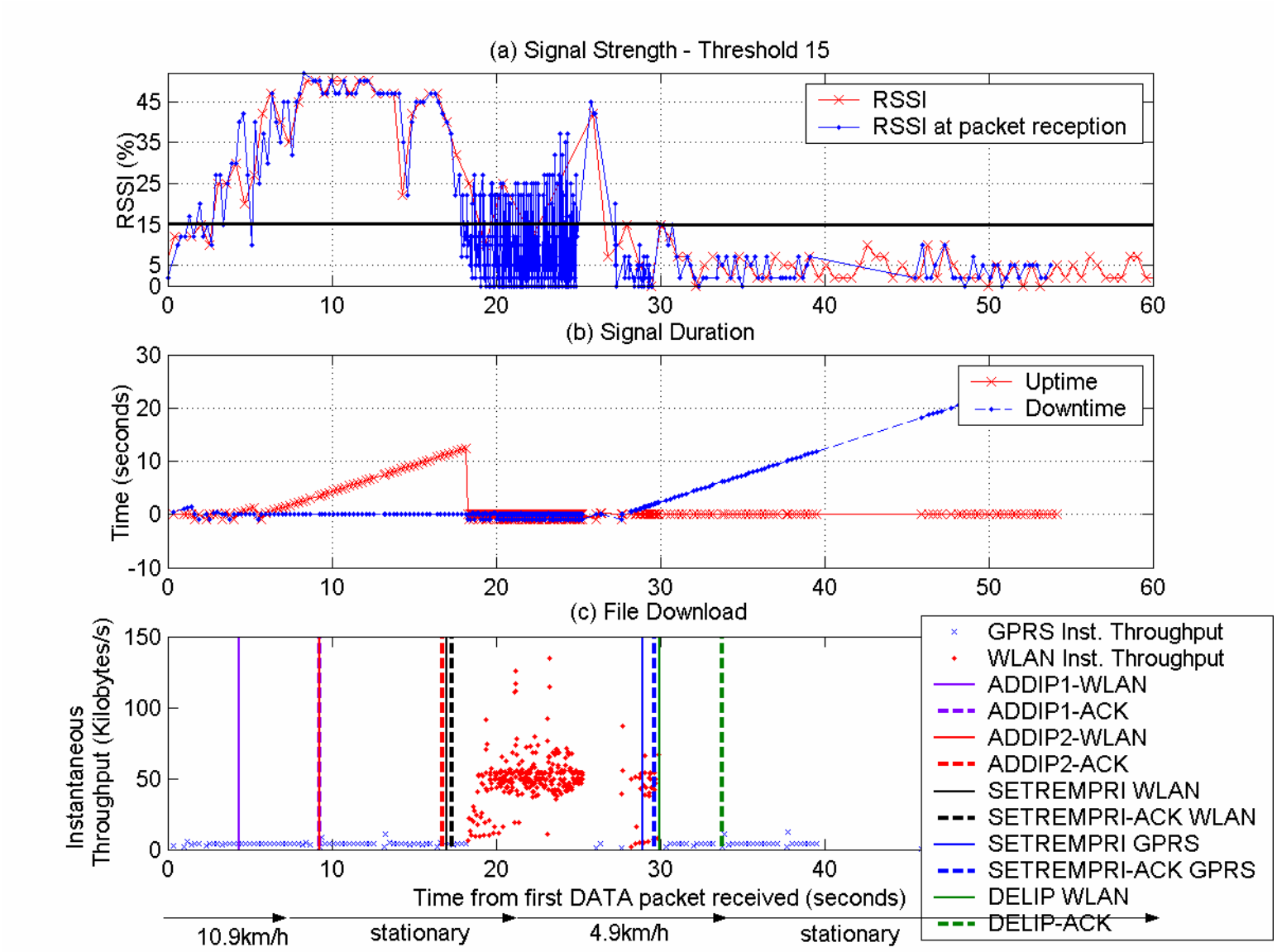

Figure 7.3 Address reconfiguration and mobility management during a file transfer

The client initiates handover to the GPRS network when downtime first exceeds 2s at t=28s. The configuration is denoted by SETREMPRI GPRS at t=28s and SETREMPRI-ACK GPRS at t=29.5s. Subsequently, the WLAN interface is removed from the association, as shown by DELIP WLAN at t=30s and the corresponding acknowledgment DELIP-ACK received at t=34s.

7.3.1 Packet loss and retransmission

As the GPRS interface is never removed from the association, it serves as a reliable always-on connection. When packets are not received on the WLAN interface, SCTP initiates data retransmission on the alternative network interface available, in this case GPRS. This may be seen by the GPRS interface receiving packets between t=27s to t=28s before handover from WLAN to GPRS is complete.

7.3.2 Variation of WLAN RSSI

WLAN RSSI was the determining factor for handover in this application. However, variation of WLAN RSSI is largely dependent on local factors such as the presence of metal objects, walls, etc in the path between receiver and access point. A sudden drop of WLAN RSSI at t=14s from 47 to 23 was observed while the vehicle is stationary. Similarly, a sudden increase of WLAN RSSI was observed from 12 to 43 at t=25s. In this implementation, uptime and downtime counters were introduced to counter the effect of spurious handovers caused by ping-pong variations in RSSI.

7.3.3 Instantaneous throughput

802.11b WLAN networks offer a theoretical maximum bitrate of 11Mbps with overheads. The instantaneous throughputs observed on the WLAN network varied

between 10 kilobytes/sec to 800 kilobytes /sec. This corresponds to a maximum instantaneous throughput of 6.4 Mbps on the 802.11b WLAN network.

On the GPRS network, average instantaneous throughput was around 4 kilobytes/sec, which corresponds to 32kbps. This roughly corresponds to 3 downlink timeslots at an average throughput of 9.6kbps with overheads.

To effectively show the GPRS throughput against the significantly higher throughput experienced on the WLAN network, the vertical scale was restricted to 150 kilobytes per second.

7.4 Importance and Implications

The mobility management application presented in this chapter is an example of an end-to-end transport layer mobility management solution. The mobile client initiates handovers based on link-layer information and may switch between heterogeneous network interfaces during a file transfer.

When a faster link, in this case WLAN, is available, the mobile client uses it as the primary link for receiving data packets. When the mobile client is leaving the coverage of the WLAN network, it initiates procedures to switch data transfer to the alternative link and deletes the WLAN network address from the association. The mobile host is able to switch network links without restarting the connection, as would be required in TCP.

In the field trial, the client starts a file transfer on the GPRS interface and gradually moves towards and finally away from a WLAN hotspot. The inverse would also be possible, i.e. the client could start its file transfer on the WLAN interface, switch to the GPRS interface as it moves away from the hotspot and resume data transfer on the WLAN interface as it approaches another access point.

The time taken to switch from one link to another is shown to be much shorter than the SCTP retransmission timeout, which, as explained in Section 2.4, is set at 63s. This is because the mobile host reacts to link-layer information that is locally available. The mobile host is responsive to ever-changing network conditions and the association is more resilient to network failures.

Once the user has been authenticated on both the WLAN and the cellular network, the switching between them becomes transparent to the user. This demonstrates internetworking between WLAN and 2.5G/3G GPRS cellular technology, and may be further extended to 3G cellular technologies such as WCDMA. Within a company or a campus with WLAN access, users would be able to access network services anywhere and anytime. They would be able to harness the high data rates and access intranet services offered by WLAN when they are within the range of an access point (typically 60m in NUS) while continuing their data transfers on cellular-based technologies elsewhere. This offers possibilities for enterprise-wide solutions for everywhere connectivity.

In the public domain, as hotspots are deployed in cafés, airports and hotels, users would be able to access high data rate network services on the go. They would not need to remain within the range of access points for fear of losing access to network services. Instead, they would have continued access to useful and essential information on the Internet related to travel, tourism, transport and entertainment.

The mobile host manages the links in a way that is transparent to the user. Users are able to roam between WLAN/3G networks and even, as has been demonstrated, different network operators. This minimises the need to restart connections and enhances the mobility experience for users.

Chapter 8 Elimination of single point of failure in asymmetric networks using source address selection

8.1 Introduction

The source address selection feature described in Section 6.6 allows for the definition of a comprehensive NAT traversal solution for SCTP and ASCONF chunks. In this chapter, we seek to extend this feature to other SCTP chunks, with the primary purpose of eliminating the single point of failure described in Section 4.3.

8.2 Selective Acknowledgment (SACK) chunks

A SACK chunk serves to acknowledge successful data reception in SCTP. One SACK chunk may acknowledge several DATA chunks. In a multi-homed mobile client – single-homed fixed server connection, the SACK chunk is one of the principal chunks that would be sent from the client to the server.

As shown in previously in Figure 4.8, regardless of whether data chunks arrive at Endpoint A via Interface 1 or 2, acknowledgment chunks are sent via Interface 1 only. Using source address selection, an acknowledgment chunk may be sent from the interface a host last received data packets on. This ensures that data packets are acknowledged as long as they are successfully received, regardless of the link conditions of the default interface.

8.3 Experimental scenario

The multi-homed mobile client initializes an association with the single-homed fixed server using the GPRS interface. As it approaches a hotspot, the WLAN interface is added to the association using the NAT traversal solution for ASCONF

described earlier. Once the WLAN interface is active, all packets leave the mobile client by the WLAN interface by default.

When a file transfer is requested by the mobile client, it receives DATA packets on the GPRS interface, which remains the primary destination. However, the mobile client transmits acknowledgment SACK packets from the WLAN interface.

This is clearly shown in Figure 8.1, which shows the destination address of outgoing packets transmitted by the server and the source address of incoming packets received by the server during a file transfer.

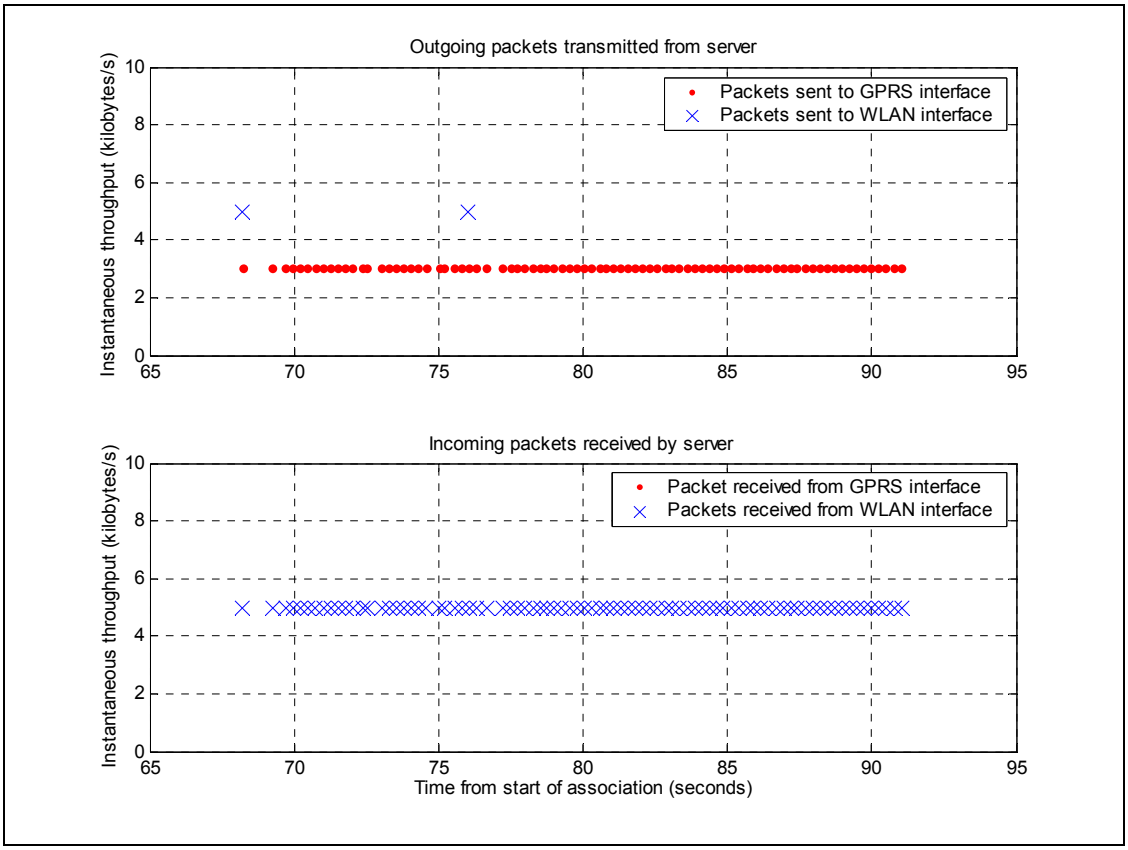

Figure 8.1 Data acknowledgment (SACK) chunks from default WLAN interface

8.4 Results

Using the source interface selection feature previously applied to ASCONF chunks, we show that SACK chunks may be sent from the last interface that received DATA chunks. Figure 8.2 shows that with this enhancement, SACK acknowledgment chunks are sent from the interface that receives DATA packets, i.e. the GPRS interface.

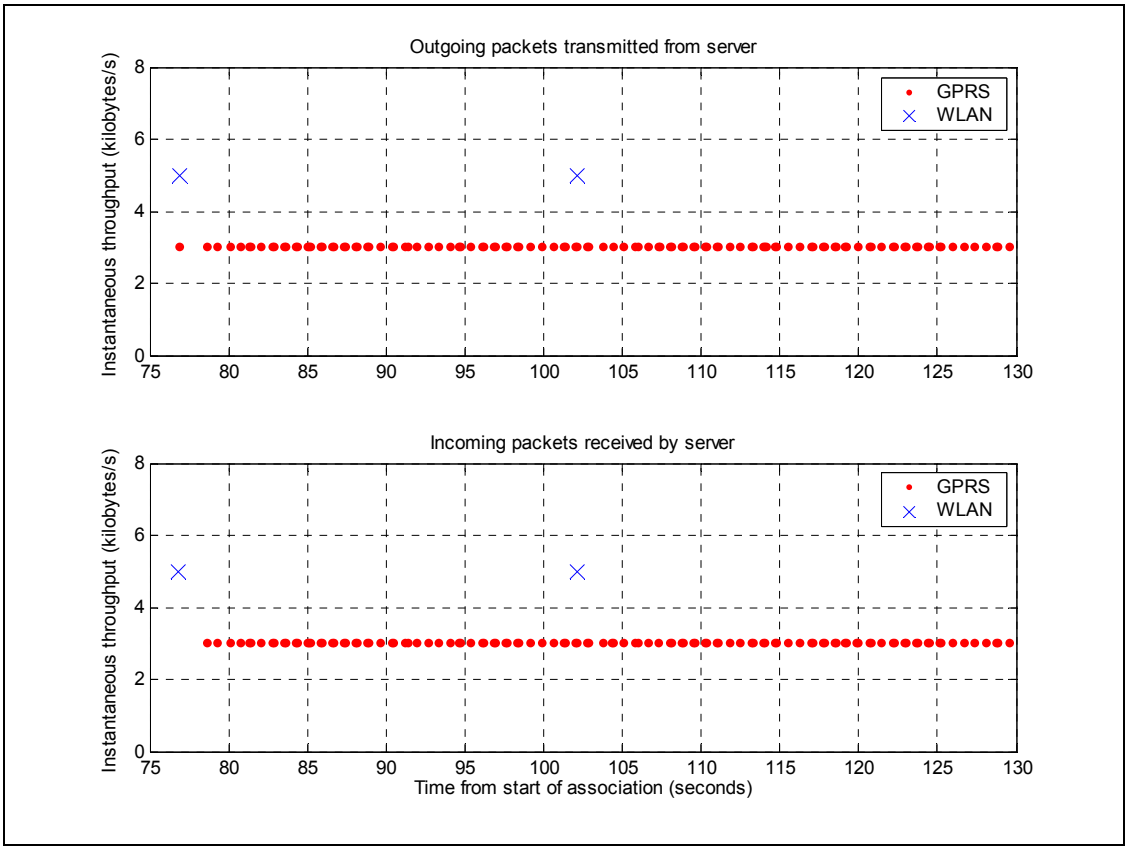

Figure 8.2 Packets acknowledged from interface that last received packets

This shows that the host is not limited to the routing table's default choice. The flexibility to select the outgoing interface for acknowledgment chunks allows the association to survive, even if the default outgoing interface experiences network failure.

8.5 Applicability to HEARTBEAT chunks

Source interface selection may be similarly applied to HEARTBEAT chunks, which are regularly transmitted to inactive destination address to test their validity. Allowing acknowledgment HEARTBEAT-ACK chunks to be sent from the interface that received the HEARTBEAT chunk would provide a more accurate representation of conditions on the link being probed.

8.6 Importance and Implications

In this chapter, we have shown that with the novel source address selection feature, a host is no longer reliable on the routing table's default choice. It would be able to switch to alternative network interfaces or perform a round-robin of its available network interfaces to improve fault resilience, without the need to manually configure routing tables or have multiple routing tables in place. Although the `IP_PKTINFO` option is not yet part of the official socket extensions for SCTP, it would be easy to include and would offer both hosts and applications the flexibility of selecting the outgoing network interface. This would be particularly useful for mobile host, as it lifts the restriction of staying within the coverage area of the default network interface and allows the association to survive in the event of link failure.

Chapter 9 Conclusion and Future Directions

9.1 Conclusions

The key weakness impeding the deployment of SCTP in IPv4 networks has been identified and overcome. Following the wide adoption of Network Address Translators (NAT), it is necessary to define a NAT traversal solution for SCTP and its Dynamic Address Reconfiguration (ASCONF) Extension.

A host behind a NAT should initialize an association using a single network interface, and subsequently add its other network interfaces to the association. For this purpose, the Dynamic Addition Reconfiguration extension has been modified to allow private network addresses to be added, deleted and set as the primary address to receive packets upon during an active connection. This is valid under the assumption that an endpoint can present multiple IP-port pairs to its peer and that NATs have basic support for SCTP, as they currently dispose of for transport protocols such as UDP and TCP.

One of the key guiding principles is not to introduce changes that would only be applicable to certain hosts. The modifications proposed to the protocol were designed with the aim of allowing hosts in private networks, be they wired or wireless, to reliably configure their addresses. However, they may even be applied as general modifications to the Dynamic Address Reconfiguration extension as they do not impede hosts with fixed global network addresses from reconfiguring their addresses.

A novel feature, source address selection, is proposed to be included in SCTP. This allows the outgoing network interface to be selected and is a key component of

the NAT traversal solution. Moreover, this improves fault resilience in asymmetric multi-homing configurations, which suffer from a single point of failure in SCTP. It is proposed that this flexibility be included in the socket extensions of SCTP, to allow asymmetric multi-homing configurations to be more fault-tolerant. Upper-layer developers will thus have the option of invoking this feature to improve fault resilience and failure recovery.

The modified SCTP allows multi-homed mobile hosts to take advantage of multiple network interfaces. An example of a transport-layer mobility management solution based on the enhanced protocol is developed. It is shown through field trials that a mobile client with heterogeneous links, in this case WLAN and GPRS, can perform soft handovers between the two links without having to restart the connection.

Finally, we extend source address selection to SCTP acknowledgment chunks. This allows associations to survive even if the default outgoing interface is experiencing network failure. More importantly, sending HEARTBEAT chunks from the interface being probed would allow for a more accurate reflection of link conditions.

The key obstacles to the deployment of SCTP in IPv4 networks have been identified and overcome. The proposed modifications and features allowed a simple but powerful mobility management solution to be developed and tested in existing commercial networks. Once SCTP, with the proposed modifications, is deployed across networks, mobile hosts may be able to take advantage of high bandwidth links whenever available and enjoy seamless roaming between heterogeneous wireless technologies. This demonstrates internetworking between WLAN and 3G cellular networks. With the increasing prevalence of WLAN in public domains, university campuses, enterprises and homes, users will be able to ride on the increased data rates

within the range of WLAN access points while relying on cellular networks as an always-on backup.

Two research papers have been submitted to relevant conferences.

9.2 Future Directions

9.2.1 One-time authentication

Before conducting the field trials, it is necessary to authenticate the host in the GPRS network as well as in the WLAN network. This facilitated a seamless handover when the mobile node approached the hotspot from a point outside the coverage area. A user who has to manually login and logout of each network service would delay handovers and reduce the convenience of transport layer link management. Furthermore, timeouts based on inactivity are often implemented in public WLAN hotspots to prevent the user from being excessively billed (e.g. 5 minutes in Singtel wireless surf zones).

The authors of [37] from AT&T Labs discuss the concept of Virtual Single Account (VSA) where a client requires only one username and password. A single sign-on interface is presented to the user and a VSA client manages the real authentication credentials and authorization rights for various wireless networks. This concept would make the handover process completely transparent to the user, as he would only need to authenticate himself once on a terminal. This would remove the hassle of re-authentication. Furthermore, the local username and password may also hide the real username and password required for various networks from prying eyes. VSA is presented as part of an integrated Internet Roaming Solution based on Mobile IP, and requires a VSA server to be installed in the administration core of networks.

It would be desirable to implement a standalone one-time authentication application for mobile clients. This would allow users to seamlessly roam between networks provided by different operators, while being appropriately billed for their access.

9.2.2 Enabling mobility management in existing applications

The mobility management demonstrated in this work is developed as an application in the userspace SCTP Reference Implementation. Link layer information is passed to the upper layers to decide which link to use and when to initiate handovers. However, this information need not be passed to the uppermost layers. An adaptation layer implemented above SCTP should be able to manage network links while presenting a homogeneous programming interface using existing socket calls to the upper layers. This would allow link management to be transparent to the upper layers and facilitate the porting of existing applications for mobility management and multi-homing support.

9.2.3 Implications of IPv6

NAT traversal issues or "middlebox issues", as they are otherwise known, remain a constant reality for mobile networks today. The modifications described in this work allow Mobile SCTP to be rapidly deployable in existing networks, as has been demonstrated NATs were primarily designed to counter the shortage of IPv4 addresses and have been widely adopted by network operators. It is not clear if the deployment of IPv6 will completely obliterate the need for NATs. The Internet-Draft

on Mobile SCTP [21] neither addresses the NAT traversal issue, nor does it identify IPv6 as a prerequisite.

In this work, it was not possible to test the enhancements in an IPv6 environment as it has been deployed in neither the Singtel WLAN nor M1 GPRS networks. However, it is expected that the enhanced SCTP would work well in IPv6 networks, with the added overhead of an additional ASCONF chunk in the dynamic addition process.

Future work may concentrate on the implications of IPv6 on Mobile SCTP. In the likely scenario where IPv6 is not deployed across all networks at the same time, the enhancements to SCTP would continue to be valid and useful to all hosts, especially those in wireless networks.

9.2.4 Load sharing and load balancing

Support for multi-homing gives SCTP the additional potential to support load-sharing and/or load balancing over the multiple interfaces. With load sharing, a single file may be split into multiple streams for transmission across the various available interfaces, thereby increasing throughput by bandwidth aggregation. A client may also balance the load across all the links available, as a function of the perceived instantaneous throughput that varies according to network conditions.

While SCTP does not currently support load sharing or load balancing, a transport protocol with these features would be extremely beneficial for mobile hosts that typically suffer from high error rates and limited bandwidth on each link.

References

[1] Vocal Technologies (2003), "GPRS Data Rates", [Online]. Available: http://www.vocal.com/data_sheets/gprs_rat.html .

[2] UMTS World (2003), "UMTS World WCDMA specification and information page", [Online]. Available: http://www.umtsworld.com/technology/wcdma.htm .

[3] IEEE 802.11b, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications – Higher-Speed Physical Layer Extension in the 2.4 GHz Band", 1999.

[4] J. Hasan, et al, "Authentication, Authorization, Accounting and Charging for the Mobile Internet", presented at 1st Mobile Communications Summit, Barcelona, September 2001.

[5] R. Stewart and Q. Xie, *Stream Control Transmission Protocol (SCTP): A Reference Guide*. Boston, MA: Addison-Wesley, 2002, pp. 287-289.

[6] R. Stewart et al., "Stream Control Transmission Protocol (SCTP)", RFC 2960, October 2000.

[7] Sun Microsystems, Inc (2002), "All IP Wireless, All the Time". [Online]. Available: http://research.sun.com/features/4g_wireless/ .

[8] M. Atiquzzaman and W. Ivanic, "Evaluation of SCTP Multi-streaming over Satellite Links", presented at 12th International Conference on Computer Communications and Networks, Dallas, TX, 2003.

[9] R. Stewart et al, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", Internet Draft (work in progess), draft-ietf-tsvwg-addip-sctp-07, February 2003.

[10] IBM (2003), "Mobility Management for wireless networks", [Online]. Available: http://www-106.ibm.com/developerworks/wireless/library/wi-mobility/ .

[11] C. Perkins, "IP Mobility Support", RFC 2002, October 1996.

[12] L. Magalhâes, R. Kravets, "A Transport Layer Approach to Host Mobility", presented at ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Atlanta, 2002.

[13] T. Kubo et al, "Path Management of SCTP to Eliminate Single Point of Failure in Multi-homing", in *Proc. IEEE 5th International Conference on Advanced Communication Technology (ICACT2003)*, pp.135-139, Korea, January 2003.

[14] H. Balakrishnan, S. Seshan, E. Amir et al., "Improving TCP/IP Performance over Wireless Networks", presented at the 1st ACM International Conference on Mobile Computing and Networking 1995.

[15] H. Balakrishnan, V. N. Padmanabhan, S. Seshan and R. H. Katz, "A Comparison of Mechanisms for Improving TCP/IP Performance over Wireless Links, " in *Proc. ACM SIGCOMM*, Stanford, CA., 1996.

[16] J. S. A. Liew, C.W. Ang and K.F. Ho, "Improving TCP Performance over Multi-Slot GSM", in *Proc. ICC*, Vancouver, June 1999.

[17] F. Lefevre and G. Vivier, "Understanding TCP's behavior over wireless links", in *Proc. Symposium on Communications and Vehicular Technology 2000*, pp. 123-130.

[18] A. Snoeren and H. Balakrishnan, "An End-to-End Approach to Host Mobility", presented at 6th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Boston, 2000.

[19] E. Kohler, M. Handley, S. Floyd, "Designing DCCP: Congestion Control without Reliability", International Computer Science Institute Centre for Internet Research, Berkeley.

[20] E. Kohler, M. Handley, S. Floyd, "Data Congestion Control Protocol", Internet Draft (work in progress), draft-ietf-dccp-spec-04, June 2003.

[21] M. Riegel et al, "Mobile SCTP", Internet Draft (work in progress), draft-riegel-tuexen-mobile-sctp-03, August 2003.

[22] S.J. Koh et al., "Architecture of Mobile SCTP for IP Mobility Support", Internet Draft (work in progress), draft-sjkoh-sctp-mobility-02, June 2003.

[23] K. Egevang, P. Francis, "The IP Network Address Translator (NAT)", RFC 1631, May 1994.

[24] P. Srisuresh, M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[25] R. Stewart and M. Tuexen, "Stream Control Transmission Protocol (SCTP) IPv4 Address Scoping, Internet Draft (work in progress), draft-stewart-tsvwg-sctpipv4-00, May 2002.

[26] L. Coene, "Stream Control Transmission Protocol Applicability Statement", RFC 3257, April 2002.

[27] P. Srisuresh, G. Tsirtsis et al., "DNS Extensions to Network Address Translators (DNS_ALG)", RFC 2694, September 1999.

[28] L. Coene, "Multihoming issues in the Stream Control Transmission Protocol", Internet Draft (Work in Progress), draft-coene-sctp-multihome-04, June 2003

[29] CNETAsia, (2003). "Where's your nearest Wi-Fi hotspot?". [Online]. Available: http://asia.cnet.com/reviews/hardware/networking/0,39001739,39126240,00.htm.

[30] L.M.H.P. Yarroll and K. Knutson, (2001). “Linux Kernel SCTP: The Third Transport”. [Online]. Available: http://old.lwn.net/2001/features/OLS/pdf/pdf/sctp.pdf .

[31] University of Essen, “Stream Control Transmission Protocol”. [Online]. Available: http://www.sctp.de/sctp.html .

[32] R. Stewart et al. “SCTP Partial Reliability Extension”, Internet Draft (Work in Progress), draft-stewart-tsvwg-prsctp-01, August 2003.

[33] L. Ong, R. Stewart, et al., “Tunneling of SCTP over Single UDP Port”, Internet Draft (Work in Progress), draft-ietf-sigtran-sctptunnel-00, March 2000.

[34] R. Stewart, Q. Xie, et al, “Sockets API for Stream Control Transmission Protocol (SCTP)”, Internet Draft (Work in Progress), draft-ietf-tsvwg-sctpsocket-07, August 2003.

[35] Unix man pages: ip (4) [Online]. Available: http://www.mcsr.olemiss.edu/cgi-bin/man-cgi?ip+7 .

[36] J. Tourrihes. (1997), “Linux Wireless LAN Howto: Wireless Extensions for Linux”. [Online]. Available: http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireless.Extensions.html .

[37] H. Luo, et al, “A WLAN/3G Integration System for Enterprises”, AT&T Labs, http://www.nd.edu/~hluo/publications/SPIE02.pdf .