Fast Transport Layer Handover Using Single Wireless Interface

Michio Honda

Faculty of Environmental Information
Keio University
5322 Endo, Fujisawa Kanagawa 252-8520 JAPAN

Submitted in partial fulfillment of the requirements for the degree of Bachelor

Advisors:

Professor Hideyuki Tokuda
Professor Jun Murai
Professor Osamu Nakamura
Associate Professor Kazunori Takashio
Instructor Ryuji Wakikawa

Abstract of Bachelor's Thesis

Fast Transport Layer Handover Using Single Wireless Interface

Many handover techniques in the Internet has been introduced with the development of mobile computing technologies. Although many proposed handover schemes utilize multiple interfaces, having multiple interfaces in a mobile device increases its power consumption, device installation space, and hardware costs. Therefore, we have been studying handover schemes for mobile nodes with a single wireless interface. To achieve seamless and efficient handover, we focus on Stream Control Transmission Protocol (SCTP) that offers a message-oriented, reliable and connection-oriented delivery transport service. Unlike other transport protocols like TCP, SCTP can provide an end-to-end handover mechanism with multi-homing feature. However, the handover mechanism in the current SCTP specification causes large handover latency particularly when a mobile node has only one single interface. This paper investigates current issues of SCTP handover mechanism, and proposes a new efficient handover scheme based on SCTP, which identifies a communication path as a pair of source and destination address. Additionally, we modified SCTP behavior when SCTP endpoints received SET PRIMARY ADDRESS messages which change primary destination of peer endpoint. This paper shows that our scheme can reduce the handover latency by two to thirty seconds.

Michio Honda

Faculty of Environmental Information

Keio University

Fast Transport Layer Handover Using Single Wireless Interface

単一無線インタフェースを利用した高速トランスポート層ハンドオーバ

近年のモバイルコンピューティング技術の発達により，インターネットにおける多くのハンドオーバ手法が提案されている．これらの手法の多くは複数の無線インタフェースを利用しているが，このことはモバイル端末にとって消費電力の増大やデバイス設置スペースの占有，コストの増大につながる．そのため，本研究では単一無線インタフェースを利用したハンドオーバ手法を提案する．本研究では，シームレスなハンドオーバを実現するため，SCTP (Stream Control Transmission Protocol) に着目する．SCTP はコネクション指向で信頼性のある，メッセージ指向の通信を提供するトランスポート層プロトコルである．SCTP によるハンドオーバは，マルチホーム機能により end-to-end で実現可能なことが特徴である．しかし現在の SCTP の仕様では，単一の無線インタフェースでハンドオーバを行う際に大きな遅延が発生する問題がある．本研究では，送信元と宛先の組み合わせ毎に輻輳制御を行うことと，SET PRIMARY ADDRESS メッセージを受け取った時の SCTP の動作を変更することでこの問題を解決する．SET PRIMARY ADDRESS メッセージは，対向のエンドポイントにプライマリアドレスを変更させるメッセージである．本研究ではこれらの変更により，ハンドオーバ時に発生する遅延を 2 秒から 30 秒短縮した．

慶應義塾大学 環境情報学部

本多 倫夫

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter describes our research background. Next, we describe our awareness of problems and the approach to challenge the issues.

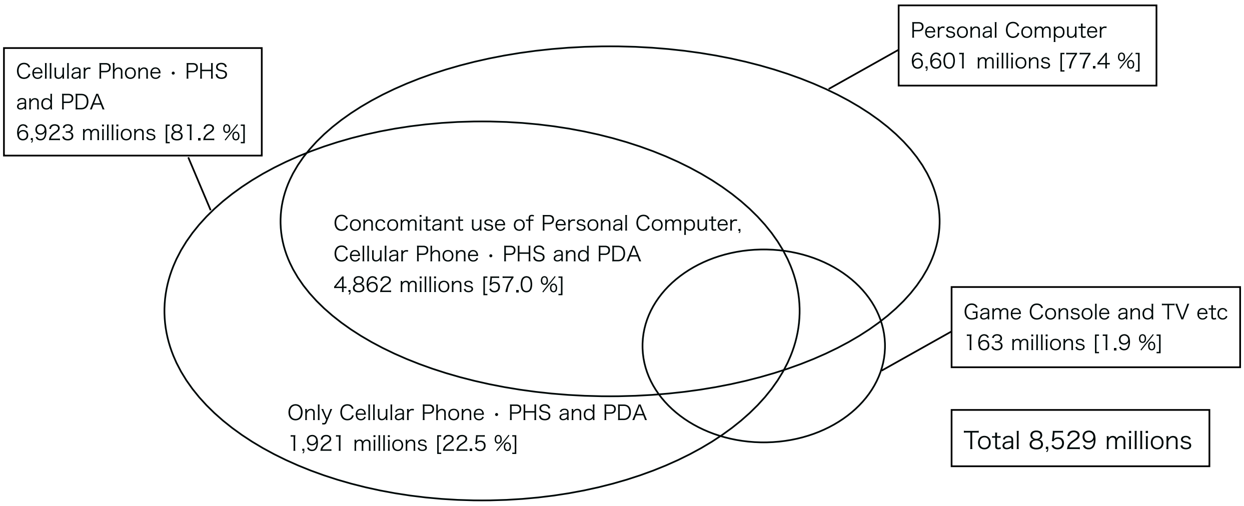


Figure 1.1: How the Internet is accessed in Japan. *Original Sources: White Paper 2006 "Information and Communications in Japan, " Ministry of Public Management, Home Affairs, Posts and Telecommunications, Japan, 2006*

1.1 Motivation

Recent hardware and wireless technology development causes interest in mobile computing. The miniaturization of hardware and the proliferation of wireless platforms enable mobile devices connect to the Internet anytime, anywhare. Figure 1.1 shows the distribution of access devices people select to connect to the Internet in Japan [3]. Most of people which access the Internet use mobile devices such as cellular phones and PDAs. This indicates that much people access to the Internet in various locations.

Recently, various wireless technologies that enable mobile nodes connect to the Internet have been developed. These are categorized into three types based on the communication coverage. First is wireless Local Area Network (LAN), second is wireless Metropolitan Area Network (MAN), third is wireless Wide Area Network (WAN). Figure 1.2 illustrates this classification .

Wireless LANs are mainly in-room, in-building, or small-area (e.g., up to about 200 meter)

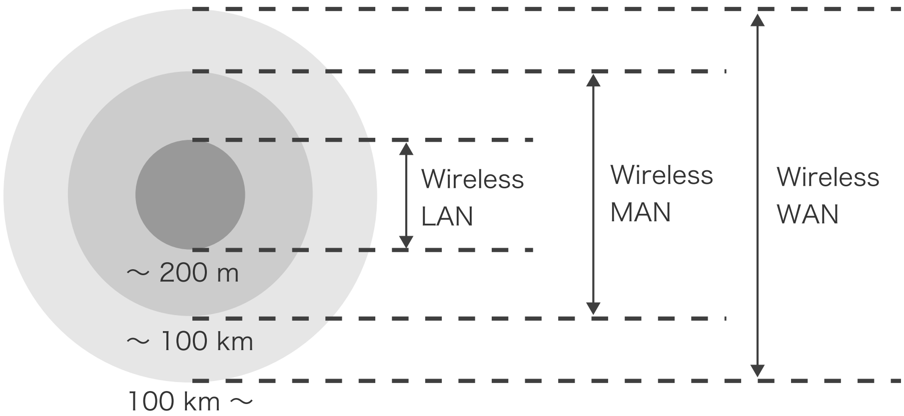


Figure 1.2: The Category of Wireless Network by the coverage.

networks, which provide maximum bandwidth to 108 Mbps over relatively small range. Wireless LAN technologies are standardized as 802.11b [16], 802.11g [14], 802.11a [15] by Institute of Electrical and Electronic Engineers (IEEE). 802.11b and 802.11g use 2.4 GHz frequency band, and 802.11a use 5.2 GHz frequency band. 802.11 families are widely deployed to the consumer public.

Since 802.11b and 802.11g use 2.4GHz frequency, no licenses are required. On the other hand, since 802.11a uses 5.2GHz frequency, the use is limited only within doors. Therefore, 802.11b and 802.11g are used popularly at not only home but cafes, airports and other public spaces. These public areas are called *Hotspots*. A number of the Hotspot increases drastically, and mobile users connect to the Internet through them with laptop computers and PDAs.

In addition, 802.11n [19] is being standardized, which uses 2.4 GHz frequency band. 802.11n offers over 100 Mbps bandwidth by innovating Muptiple-Input Multiple-Output (MIMO) technology.

Wireless MANs are middle area (e.g., up to about 30 kilo meters) networks, which provide maximum bandwidth up to 130 Mbps. Wireless MANs assume many communication scenarios, for example, access networks for users, communication lines between subscriber stations, last mile access between home and a base station, and high-speed mobile communications such as cars.

Wireless MAN technologies are being standardized as 802.16 [13] and 802.20 [12] by IEEE. Worldwide Interoperability for Microwave Access (WiMAX) [5] is the industry standards of 802.16. The consortium gives qualifications to products. Flash-OFDM [31] and iBurst [17] are being proposed to 802.20 as the standards.

Wireless WANs are wide area networks which cover country-level area, that provide maximum bandwidth up to 5 Mbps. These are used in cellular network. High-Speed Downlink Packet Access (HSDPA) [6] is standardized by 3rd Generation Partnership Project (3GPP), and Evolution-Data Optimized (EV-DO) [7] are standardized by 3rd Generation Partnership Project 2 (3GPP2). HSDPA provides maximum bandwidth 2 Mbps and EV-DO provides maximum bandwidth 3.1 Mbps in the downlink.

Not only wireless technologies, but also Internet technologies have been evolved. One of them is IPv6 [27], which is a next generation network protocol standardized by Internet Engineering Task Force (IETF). Although most of today's Internet uses IPv4, many common Internet applications already work with IPv6, and IPv6 is gradually becoming available [2]. IPv6 adds many improvements to IPv4. One of the advantages of IPv6 is a theoretical address space of $3.4 * 10^{38}$ addresses. This is because an IPv6 address is composed by 128 bit, while an IPv4 address is composed by 32 bit. This address space can accommodate to a large growth in the number of nodes which connect to the Internet.

In the near future, not only cellular phones and PDAs, but also mobile multimedia devices (e.g. portable music devices and digital camcorders) will be connected to the Internet through the wireless platforms. This situation will promote the emergence of new services, such as music streaming for portable music devices and video streaming to the Internet from digital camcorders. Some users will access those services while moving.

Hence, handover, which is a technology that supports a migration from one network to another, is becoming more important in IP communication. When the IP address of a mobile node is changed, all TCP connections on the node will be terminated and applications have to be reconnected. Wireless networks are provided by various carriers, which maintains each IP network separately. Thus, when a mobile node moves from one network to another, it usually traverses over different IP networks.

1.2 Challenges and Contributions

Recently, many handover technologies have been proposed and studied, and most of them presume that mobile nodes equip multiple network interfaces [11, 21, 20, 28].

However, the idea of having multiple network interfaces has several issues in mobile nodes due to the limitation of battery capacity and device installation space. In addition, it might increase the hardware costs. Hence, handover technologies that can be used for mobile nodes with a single wireless network interface are required.

In this paper, we present a fast transport layer handover technique for single wireless interface (WNIC) node, which is based on SCTP. We utilize ADD-IP extension[25] in SCTP functions. One of the advantages of utilizing SCTP is that an SCTP endpoint can support multiple IP addresses per association and ADD-IP extension allows an SCTP endpoint to change its IP addresses during communication. By using these features, SCTP itself can support handover. However, when SCTP and ADD-IP extension are used on nodes with a single interface, communication delay of SCTP caused by handover is significantly large compared to the disconnected time in the lower layer. In this paper, we focus on handover schemes for nodes with single WNIC and propose a fast handover scheme based on SCTP.

1.3 Structure of Thesis

The rest of this paper is organized as follows.

Chapter 2 describes related work and SCTP overview, and clarifies the handover issues for nodes with a single WNIC. Chapter 3 explains design of our proposed scheme. Chapter 4 explains our implementation. Chapter 5 evaluates it. Finally, the paper concludes with Chapter 6.

Chapter 2

Issues

This chapter describes related works. Next, we explain overview of SCTP. Then, we describe SCTP handover mechanism and its issues.

2.1 Related Work

There are several technologies which enable a handover between different IP networks using a single WNIC. These can be categorized into three approaches: combination of TCP and Mobile IPv6, TCP extension and application layer.

2.1.1 Combination of TCP and Mobile IPv6

The first approach is to leverage Mobile IPv6 [9] under TCP. Since Mobile IPv6 hides change of IP address to the upper layer, TCP can continue the communication when the configured IP address has changed.

Mobile IPv6 enables mobile nodes to be addressed by a home address even when the nodes connect to different networks in the Internet. Home agents and correspondent nodes maintain bindings of home addresses and care-of addresses, where care-of addresses are the addresses configured in the visiting network. When a mobile node connects to a new network, the node updates the association of the home address and the care-of address by sending messages to home agents and correspondent nodes. This is called a *Binding Update*.

However, a long handover latency caused by complexity of *Binding Update* occurs unnecessary congestion avoidance of TCP, which leads a performance degradation. Several techniques have been introduced to minimize this latency an extension of Mobile IPv6 such as Fast Mobile IPv6 [23] and Hierarchy Mobile IPv6 [8]. However, since they need modifications to network, long time is required to spread those modifications.

2.1.2 TCP extension

The second approach is to modify TCP. TCP-R [10] and Migrate TCP [29] maintain active TCP connections by notification from the mobile node when the disconnection occurs due to the change of the IP address. An advantage of this approach is that no modification is needed in networks and in existing TCP applications.

However, these approaches of combination of TCP and Mobile IPv6, and TCP extension assume TCP use. Although TCP is a prevalent connection-oriented transport layer protocol, SCTP provides useful new features based on TCP such as multi-home, multi-stream and partial reliability. Therefore, SCTP can offer more flexible communication to applications. Hence, we conduct this research based on SCTP.

2.1.3 Application Layer Approach

Implementing handover mechanism in application layer requires no kernel modification, e.g., MobileSocket [30]. However, applications are limited by the implementation of the mechanism, e.g., Java for MobileSocket. Furthermore, radical modifications are required for applications to exploit the mechanisms. Since SCTP has APIs similar to TCP, required modifications to exploit it are small compared with exploiting the application mechanisms.

2.2 SCTP

SCTP [26] is a transport layer protocol standardized by Internet Engineering Task Force (IETF). Same as TCP, SCTP offers an end-to-end, connection-oriented and reliable delivery transport service for applications communicating over IP networks. Therefore, although SCTP provides special socket APIs to utilize SCTP-specific features, many socket APIs of SCTP are similar to those of TCP.

One of the features of SCTP is multi-stream function. This feature allows applications to deliver data on multiple independent streams in an association, while TCP delivers data on a single stream in a connection. Since packet losses and latency on one stream do not affect the delivery on other streams, this feature is suitable for the applications which treat multiple data types such as video streaming and multimedia web documents. Other features of SCTP are Message-oriented transmission function and Partial Reliability extension [24]. These functions are appropriate for realtime applications such as telephony applications and streaming applications.

Additionally, an SCTP endpoint can have multiple IP addresses per association. When the peer endpoint has multiple addresses, SCTP performs congestion per destination with the same mechanism as TCP. An SCTP endpoint selects one of the IP addresses as a primary destination address. DATA chunks transmitted over an SCTP association will be sent using this primary destination address. Other addresses are used as secondary destinations, which are employed on retransmissions. In addition, if a sender-side SCTP continuously fails to receive SACK chunks, which is sent from the receiver to acknowledge received DATA, the endpoint changes its primary address to one of the secondary destination addresses. The threshold error count for the change of the primary address is usually set to 5 which is the recommended value in the SCTP specification. To verify the reachability of the peer, an SCTP endpoint transmits heartbeat chunks to all destination addresses at constant intervals.

ADD-IP [25] is an extension of SCTP that allows an SCTP endpoint to add and delete IP address used in the association. It also allows an SCTP endpoint to change its primary destination address. Special chunk types, called ASCONF and ASCONF-ACK are used for this extension. By sending ASCONF messages, SCTP endpoints can notify peer endpoints of new IP addresses when they move to another network so that applications can continue the communication without terminating.

2.3 Issues in SCTP Handover Mechanism

2.3.1 SCTP Handover Mechanism

Figure 2.1 depicts a handover scenario that we assume. In this scenario, a single-homed mobile node (MN) communicates to a correspondent node (CN) over SCTP associations.

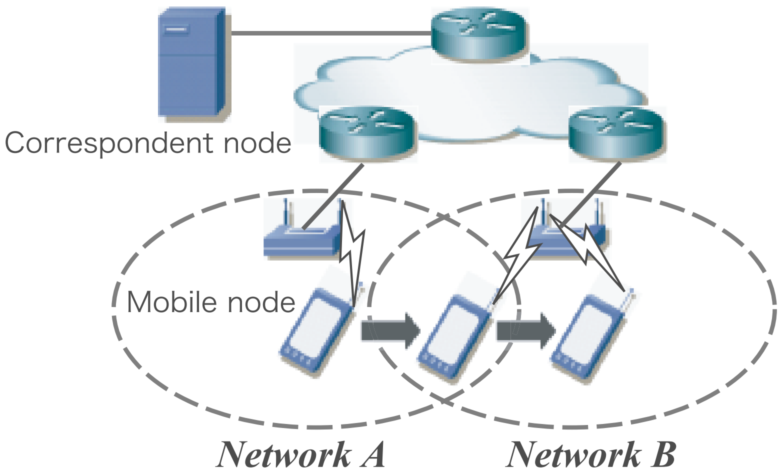
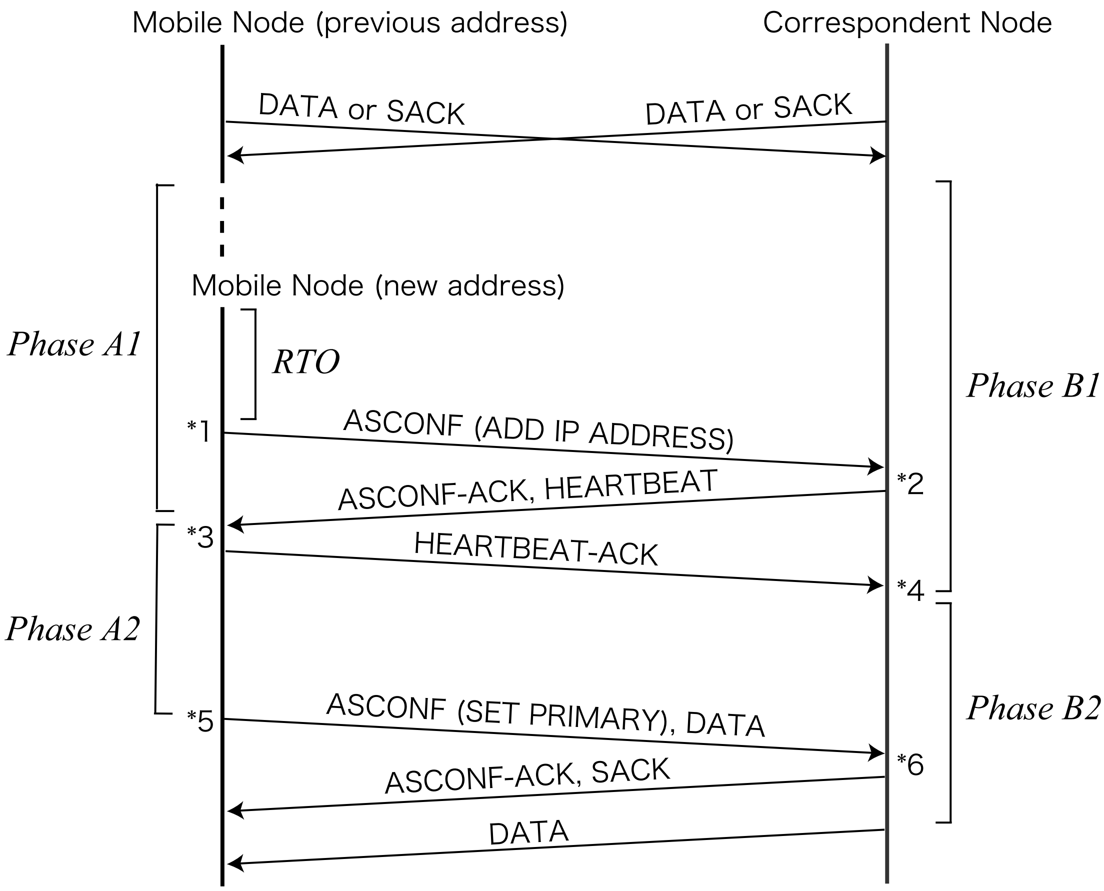


Figure 2.1: Handover Scenario

Both the MN and CN regularly transmit data each other. The MN connects to a wireless base station in *Network A*. After a certain period, the MN moves to *Network B*. During this movement, the signal strength from the current base station decreases and that from the base station in the visited network increases. Eventually, the MN stops using the current base station and connects to the new base station in the visited network. At this moment, the MN does not have IP reachability since it has not configured an available IP address yet. Since we presume the use of IPv6 in this scenario, a new IP address can be configured after the reception of the router advertisement message from the router in the visited network.

Figure 2.2 shows the handover procedure in the scenario described in Figure 2.1 based on the specification of SCTP and ADD-IP extension. When the MN's SCTP endpoint is notified of an activated address from the network layer, the endpoint adds this address to the temporary use list in the association. The MN sends an ASCONF chunk which includes ADD IP ADDRESS parameter in order to use this address as the source address of any chunks other than the ASCONF chunk (Figure 2.2, *1). The CN sends back an ASCONF-ACK chunk to the source address of the ASCONF chunk (*2). After the reception of the



Phase A1: Retransmissions from previous address by MN
Phase B1: Retransmissions to MN' s previous address by CN
Phase A2 and *B2*: Remaining time before the next retransmission

Figure 2.2: Handover Timeline

ASCONF-ACK for the ASCONF (*3), the MN can use the added address as the source address for all DATA and control chunks such as SACK.

To verify new destination, the CN sends a HEARTBEAT chunk to the added destination (*2), which might be piggybacked on the ASCONF-ACK chunk. Upon the reception of the HEARTBEAT chunk (*3), the MN sends back a HEARTBEAT-ACK chunk to the CN. Upon the reception of the HEARTBEAT-ACK chunk (*4), the CN regards the MN's new address as confirmed so that it can send DATA and other control chunks to the address.

As an option, to make the CN use the added address as the primary destination, the MN can send an ASCONF chunk includes SET PRIMARY ADDRESS parameter (*5), which might be piggybacked on other chunks such as DATA. Upon the reception of the ASCONF chunk (*6), the CN uses the added address as the primary destination and sends back an ASCONF-ACK chunk as the response. After that, the MN and CN continue the communication with normal procedure.

2.3.2 Issues

The issue in SCTP handover is that the handover process takes time especially when an MN has only one WNIC. This is caused by the lower layer disconnection when the MN changes the attached network. Since both of SCTP endpoints do not notice it, they continue the data retransmission failure while the disconnection. Therefore, the retransmissions cause unnecessary congestion avoidance or multiple Retransmission Time Outs (RTOs) in SCTP communication. The retransmission time out is doubled each time a timeout occurs. Thus, SCTP endpoints suffer long delay when multiple RTOs occurs.

When an MN configures a new address in the visited network, the MN tries to send an ASCONF chunk with ADD IP ADDRESS parameter from the new address. According to the current ADD-IP specification, the MN must send the ASCONF after the RTO timer for the destination expires. This is because the congestion control of SCTP is based on destination address. This means RTO or other parameters related to the condition of a communication path are maintained based on destination address. Hence, these parameters are reset only when the destination address changes and nothing happens to these parameters when the source address changes. This also means that SCTP endpoints recognize communication paths only by destination addresses. When multiple RTO expirations have occurred during the disconnection, the value of RTO becomes quite large and it takes long time to transmit the ASCONF chunk.

Additionally, RTO expirations during the disconnection cause another problem in the MN side. In Figure 2.2, *Phase A1* shows the period which the MN continues to retransmit chunks from the previous source address. Unfortunately, these retransmissions always fail. After receiving ASCONF-ACK, the MN can use the new address as the source address of all chunks. However, as shown in *Phase A2*, DATA chunks cannot be transmitted until the current RTO timer expires due to the failure of retransmissions occurred in *Phase A1*. If the MN retransmitted DATA chunks from the previous address after the MN attached to the visited network, the DATA chunks might reach to the CN. However, since the CN is not noticed the MN's new address by the exchange of an ASCONF and HEARTBEAT yet, the corresponding SACK chunks are sent to the MN's previous address. Therefore, the MN cannot receive the SACK chunks since the MN's previous address is already unreachable. Thus, the MN fails these retransmissions and the failures also cause extra increase of the RTO.

The CN side also has an issue, which causes long data transmission latency. During *Phase B1*, the CN continues to retransmit chunks to the MN's previous address. Since the MN's previous address is already unreachable, all retransmissions fail in this period. After receiving a HEARTBEAT-ACK from the MN's new address, the CN can use the MN's new address as the destination for all chunks. However, as shown in *Phase B2*, since DATA chunks cannot be transmitted until the current RTO timer expires due to the failure of retransmission in *Phase B1*, it takes long time for the CN to restart data transmission.

If an MN can have multiple WNICs, the MN's old address can be reachable during handover in some cases. For example, if the coverages of access points are overwrapped, the MN can belong to multiple access points simultaneously. In such a situation, data transmissions that use the MN's old address do not fail and the RTO value remains small. However, as described above, if an MN has only one WNIC, delays after the handover process become significant due to the failure of the data transmission that uses the MN's old address.

Chapter 3

Design of SmSCTP

This chapter describes our fast handover scheme named SmSCTP, which is composed of three algorithms.

3.1 SmSCTP Overview

To solve the handover issues in SCTP described in Section 2, we propose Single interface adaptive mobile SCTP (SmSCTP) as an extension of SCTP. SmSCTP achieves fast handover on single WNIC nodes.

Additionally, SmSCTP is designed based on several SCTP security policies to reconfigure address in existing SCTP association. The ADD-IP specification requires that an SCTP endpoint sends an ASCONF and receives an ASCONF-ACK for the ASCONF before the endpoint uses a new address as a source address to send any chunks without the ASCONF includes ADD IP ADDRESS parameter. This is because, when SCTP endpoints receive indirect chunks, these chunks might are passed applications. Therefore, SCTP endpoints reject chunks from the unknown destination, which is not verified by the reception of the ASCONF.

In addition, the SCTP specification requires that an SCTP endpoints which noticed a new address by an ASCONF verify the reachability for the destination by sending a HEARTBEAT. This is because, despiteous SCTP endpoints might send ASCONFs which include ADD IP ADDRESS parameter to attack third-party nodes. If the endpoint which was received the ASCONF did not verify the address, the endpoint might send many chunks to the third-party node.

SmSCTP consists of two algorithms: *Fast Association Reconfiguration* and *Fast Transmission Recovery*. Figure 3.1 illustrates the points that use these algorithms.

3.2 Fast Association Reconfiguration

As described Section 2, SCTP cannot send an ASCONF chunk until RTO timer expires.

To solve this issue, we propose *Fast Association Reconfiguration* algorithm. *Fast Associ-*

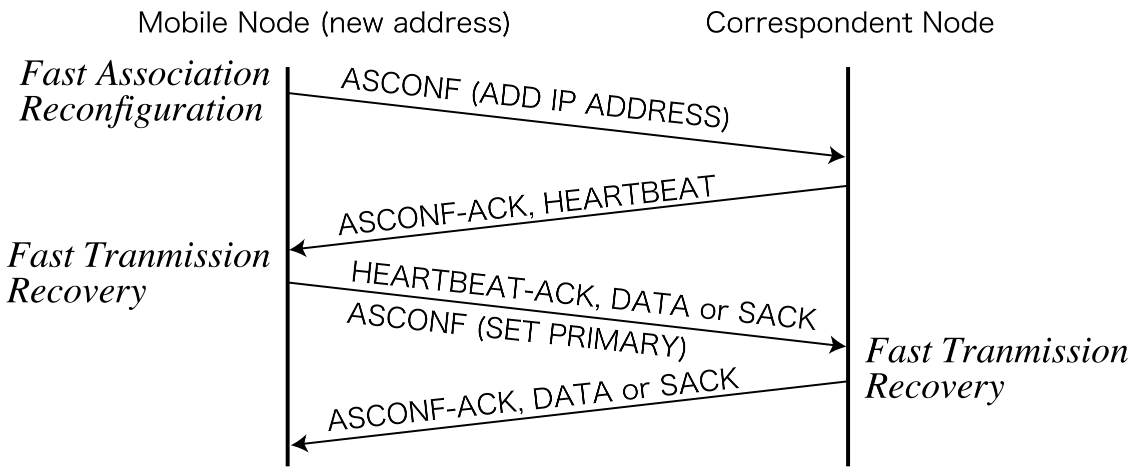


Figure 3.1: SmSCTP architecture

ation Reconfiguration allows a MN to send an ASCONF chunk as soon as the new address is configured without waiting for the RTO expiration. This algorithm regards the change of the source address in an SCTP association as the change of communication path. Hence, it resets RTO when it detects the change of source address and can transmit an ASCONF immediately. This means that SCTP endpoints recognize communication paths by a pair of source and destination addresses in this approach.

In some cases, the change of source address does not indicate the change of their communication path, such as renumbering source addresses. However, we believe this rarely happens while handover in wireless networks happens frequently. In addition, *Fast Association Reconfiguration* sends only one ASCONF chunk right after the change of the source address. A single small packet transmission hardly affects congestion status even if communication path has not been changed by the change of source address.

If the MN cannot receive the ASCONF-ACK for the ASCONF due to the loss of the ASCONF or ASCONF-ACK chunk, the MN retransmits the ASCONF chunk by the normal retransmission scheme in SCTP and doubles the current RTO. This is because the MN should consider that this event is caused by congestion in the new communication path. Based on these concepts, we believe that *Fast Association Reconfiguration* is a quite reasonable approach.

3.3 Fast Transmission Recovery

Another issue in SCTP handover is that data chunk cannot be retransmitted until RTO timer expires after the exchange of ASCONF includes ADD IP ADDRESS parameter and the ASCONF-ACK. To solve this issue, we propose another algorithm: *Fast Transmission Recovery*. This algorithm is used at both a MN and CN while *Fast Association Reconfiguration* is used at an only MN. In the MN side, this algorithm allows the MN to send DATA chunks right after the reception of ASCONF-ACK. In the CN side, this algorithm allows the CN to send DATA chunks to the MN's new address right after the reception of the HEARTBEAT-ACK from the MN's new address.

3.3.1 Algorithm in Mobile Node Side

As described above, in our approach, SCTP endpoints recognize communication paths by a pair of source and destination addresses. Therefore, a MN resets not only RTO but also other parameters related to congestion control, such as congestion window size. Besides that, the MN starts sending chunks with slow-start algorithm [18]. Hence, even if the MN's visiting network was congested, the MN could prevent the increase of the congestion.

Since our algorithm always tries to utilize newly added address, it might cause a problem in multi-homed environment. For example, if the MN configured to add another WNIC which has narrow bandwidth compared to the existing WNIC, it would be better to use the existing WNIC. To avoid this situation, we utilize our algorithm only when the newly added address and the previous address belong to the same interface.

3.3.2 Algorithm in Correspondent Node Side

In a CN side, this algorithm allows the CN to send and retransmit DATA chunks to the new primary destination right after the reception of a HEARTBEAT-ACK for the destination piggybacked an ASCONF chunk that includes SET PRIMARY ADDRESS parameter. If the exchange of the ASCONF with ADD IP ADDRESS parameter and the HEARTBEAT succeeds, it clearly indicates that the new destination becomes reachable while the reachability of the previous primary destination might be lost.

When the MN is multi-homed, it is possible that the MN sends an ASCONF includes SET PRIMARY parameter for the new primary address while the MN keeps the reachability of the old primary address. In this case, since the CN retransmits DATA chunks immediately after the reception of the ASCONF and HEARTBEAT, the MN might receive duplicate DATA chunks sent to both old and new primary addresses. However, since the CN starts sending DATA to the MN's new address with slow-start algorithm, the number of duplicate DATA packets should be minimal. Hence, these DATA packets could hardly burden network traffic.

Chapter 4

Implementation

This chapter describes the implementation of SmSCTP. At the same time, we describe modifications of the existing implementation to work SmSCTP.

We have implemented SmSCTP in the SCTP kernel implementation [4] on FreeBSD 6.1 [1]. The SCTP implementation provides basic SCTP features, ADD-IP extension, and other several extensions. Additionally, we modified several parts of existing SCTP codes to work SmSCTP correctly.

4.1 Implementation of SmSCTP

4.1.1 Fast Association Reconfiguration

Fast Association Reconfiguration is implemented into a part to compose ASCONF messages which include ADD IP ADDRESS parameter. This is because, in the current SCTP implementation, a timer for the current RTO starts when the ASCONF is composed.

After the timeout, the ASCONF chunk is sent to peer endpoints. Therefore, we define a new timer parameter, *FASCONF*. Since the timer timeouts immediately when *FASCONF* parameter is set, the ASCONF chunk is sent at the same time by *sctp_chunk_output* function.

4.1.2 Fast Transmission Recovery

An MN side algorithm of *Fast Transmission Recovery* is implemented in *sctp_addr_mgmt_ack()*, which handles ASCONF-ACKs including ADD IP ADDRESS parameters. Figure 4.1 shows the added code in *sctp_addr_mgmt_ack()*.

stcb is a structure allocated for each SCTP association in an endpoint. *net* is a structure allocated for each destination and manages destination information in an SCTP association. The information includes a destination address, source address, window size, RTO, and destination state.

when the interface for the destination and the interface which is assigned a new address are not same, following processes are skipped. This case indicates that the endpoint is multi-homed, and the added address cannot be used to send packets to the destination.

```
TAILQ_FOREACH(net, &stcb->asoc.nets, sctp_next) {
        if (net->ro.ro_rt != NULL) {
                if (net->ro.ro_rt->rt_ifp != addr->ifa_ifp)
                    continue;
        }
        sctp_timer_stop(SCTP_TIMER_TYPE_SEND, stcb->sctp_ep, stcb, net);
        sctp_init_congestion_info(net, stcb);
        TAILQ_FOREACH(chk, &stcb->asoc.sent_queue, sctp_next) {
                cnk->sent = SCTP_DATAGRAM_RESEND;
                sctp_ucount_incr(stcb->asoc.sent_queue_retran_cnt);
        }
        sctp_clear_src_cache(net);
}
```

Figure 4.1: Added codes in *sctp_addr_mgmt_ack()*

When the two interfaces are the same, the following processes are executed. *sctp_timer_stop()* stops a running timer specified by the timer type and the destination. In *sctp_addr_mgmt_ack()*, this function is called to prevent RTO and other congestion related values increase. *sctp_init_congestion_* initializes the destination information in *net*.

Figure 4.2 shows the substance of *sctp_init_congestion_info*. Lines 4 - 9 initialize IPv6 dependent parameters. Lines 11 - 17 initialize parameters related to congestion control.

A CN side algorithm of *Fast Transmission Recovery* is implemented in *sctp_process_asconf_set_primar* which handles ASCONF chunks including SET PRIMARY ADDRESS parameter Figure 4.3 shows our implementation. A part from lines 1 - 11 processes a retransmission queue, which contains DATA chunks which are already sent. When there are chunks in this queue, this part changes the destination address of these chunks into a new destination address. In addition, this part marks *SCTP_DATAGRAM_RESEND* to these chunks in order to send them immediately. A part from lines 12 - 17 checks a send queue, which contains DATA chunks which are not sent yet. When there are chunks in this queue, this part changes the destination address of these chunks to a new primary destination address.

```
void
sctp_init_congestion_info(struct sctp_nets *net, struct sctp_tcb *stcb)
{
        if (net->ro._l_addr.sa.sa_family == AF_INET6) {
                net->tos_flowlabel = stcb->asoc.default_flowlabel;
        }
        else if (net->ro._l_addr.sa.sa_family == AF_INET) {
                net->tos_flowlabel = stcb->asoc.default_tos;
        }
        net->mtu = net->ro.ro_rt->rt_ifp->if_mtu;
        net->RTO = stcb->asoc.initial_rto;
        net->ssthresh = stcb->asoc.peers_rwnd;
        net->cwnd = min((net->mtu * 4), max((2 * net->mtu),
            SCTP_INITIAL_CWND));
        if (net->cwnd < (2 * net->mtu)) {
                net->cwnd = 2 * net->mtu;
        }
}
```

Figure 4.2: *sctp_init_congestion_info function*

4.2 Modification to the Existing Implementation

To work SmSCTP, we modified the address notification module in the network layer. Additionally, we also modified the source address selection module of ASCONF chunks including ADD IP ADDRESS parameter in the current SCTP kernel implementation. These are common improvements of SCTP kernel implementation, rather than SmSCTP-specific modification.

4.2.1 Timing of address notification

We modified the notification timing of a newly added IPv6 address to the SCTP stack. In the current reference SCTP implementation, the new address is notified to the SCTP stack when it is configured. However, SCTP cannot use the address, since the status of the address is treated tentative. This is because IPv6 stack initiates Duplicate Address Detection (DAD)

```
if (!TAILQ_EMPTY(&stcb->asoc.sent_queue)) {
        sctp_timer_stop(SCTP_TIMER_TYPE_SEND, stcb->sctp_ep, stcb,
            prev_prim);

        TAILQ_FOREACH(chk, &stcb->asoc.sent_queue, sctp_next) {
                chk->whoTo = net;
                chk->sent = SCTP_DATAGRAM_RESEND;
                atomic_add_int(&net->ref_count, 1);
                sctp_ucount_incr(stcb->asoc.sent_queue_retran_cnt);
        }
}
if (!TAILQ_EMPTY(&stcb->asoc.send_queue)) {
        TAILQ_FOREACH(chk, &stcb->asoc.send_queue, sctp_next) {
                chk->whoTo = net;
                atomic_add_int(&net->ref_count, 1);
        }
}
```

Figure 4.3: Added code in sctp_process_asconf_set_primary

procedure to check whether other nodes on the link have the same address or not. Therefore, we modified it to notify SCTP from the IPv6 stack after the completion of DAD procedure for the new IPv6 address.

4.2.2 Source address selection module

We modified the source address selection module in order to output an ASCONF which includes ADD IP ADDRESS parameter from a new source address. In the current reference SCTP implementation, this module does not select the source address of SCTP packets based on default gateway in the routing table. This module select a output interface by *rtalloc()* function, which is a function in kernel to lookup the route for a destination.

Next, this module select the source address of the output interface lookuped by the *rtalloc()*. Then, selection of the source address is based on address scope, such as link-local addresses and global scope addresses. Therefore, when several global IPv6 addresses which have different prefix are configured on the interface, an inappropriate address can be selected as source address.

To select an appropriate source address, we added *sctp_v6src_match_nexthop* function, which compares the selected source address and the next hop gateway address prefixes. Figure 4.4 shows the code of *sctp_v6src_match_nexthop*.

```
int
sctp_v6src_match_nexthop(struct sockaddr_in6 *src6)
{
        struct nd_prefix *pfx = NULL;
        struct nd_pfxrouter *pfxrtr = NULL;

        /* get prefix entry of this address */
        LIST_FOREACH(pfx, &nd_prefix, ndpr_entry) {
                if (pfx->nbpr_stateflags & NDPRF_DETACHED)
                        continue;
                if (IN6_ARE_MASKED_EQUAL(&pfx->ndpr_prefix.sin6_addr,
                    &sin6->sin6_addr, &pfx->ndpr_mask))
                        break;
   }
   /* no prefix entry in the prefix list */
   if (pfx == NULL)
                return (0);
        for (pfxrtr = pfx->ndpr_advrtrs.lh_first; pfxrtr;
                pfxrtr = pfxrtr->pfr_next) {
                if (pfxrtr->router->installed)
                        return (1);
        }
        return (0);
}
```

Figure 4.4: A function of *Gateway Prefix Lookup* feature

This function returns 1 when the default gateway which belongs to the prefix of the passed address is installed to the routing table, returns 0 when it is not installed to the routing table.

nd_prefix is a structure to express a network address prefix. *nd_pfxrouter* is a structure to express a default gateway. A part of lines 8 - 14 obtains a prefix entry corresponding to the passed IPv6 address. A part of lines 18 - 22 checks if the default gateway in the prefix entry is installed to the routing table.

4.2.3 Policy to select a source address of ASCONF

Our last modification is the policy to select a source address of the ASCONF which includes ADD IP ADDRESS parameter. In the current reference SCTP implementation, the ASCONF is sent from a source address which has been used previously to send DATA. A new source address is selected at the retransmission of the ASCONF. Since an ASCONF-ACK is sent to the source address of the ASCONF, an MN cannot receive the ASCONF-ACK when it sent the ASCONF from the previous source address. Therefore, unnecessary latency arose to receive the ASCONF-ACK. We modified the policy to select the new address as a source address on the first ASCONF transmission.

Chapter 5

Evaluation

In this chapter, we evaluate the efficiency of SmSCTP, and prove it. Firstly, we describe the testbed to evaluate SmSCTP. Secondly, we show our experiment results.

We prove the efficiency of our scheme by some experiment results of the handover latency for SmSCTP and the original SCTP. To evaluate the performance of our scheme, we measure Transport Sequence Number (TSN) growth in SCTP associations during handover. In addition, to evaluate details of *Fast Association Reconfiguration* and *Fast Transmission Recovery*, we measure configuration time of a new address, sending time of ASCONF including ADD IP ADDRESS, and recovery time of DATA transmission.

Each experiment is conducted with and without DAD in network layer configuration process. Address configuration delay is a common severe problem in mobile environment, because mobile nodes reconfigure their IP address frequently on movement. Although most cases of DAD is far more likely to succeed than fail, it accounts for a large percentage of address configuration delay. Therefore, a proposition that avoid the DAD delay is standardized in IETF [22], and other techniques such as Proactive DAD [32] have been studied. According to such a situation, we believe that the DAD delay will be drastically reduced in the near future. Hence, we performs some experiments without DAD delay in addition to the experiments with DAD delay.

The experimental network environment is illustrated in Figure 5.1. The MN in Figure 5.1 is an IBM Thinkpad X40 with 1.3GHz Pentium M Processor and 1280MB memory with a single Lucent Hermes chip WNIC. The CN is an IBM Thinkpad X40 with 1.0GHz Pentium M Processor and 512MB memory with a wired-network. Each of *Network 1* and *2* contains an 802.11b wireless access point, and their network addresses are 2001:200:0:8861::/64 and 2001:200:0:8862::/64, respectively. The coverage area of the wireless networks overwrap. To generate 20 ms RTT between the MN and the CN, a bridge node with Dummynet is located in *Network 3*. We use two SCTP associations for the communication between the MN and the CN, which appear as *Association A* and *Association B* in Figure 5.1.

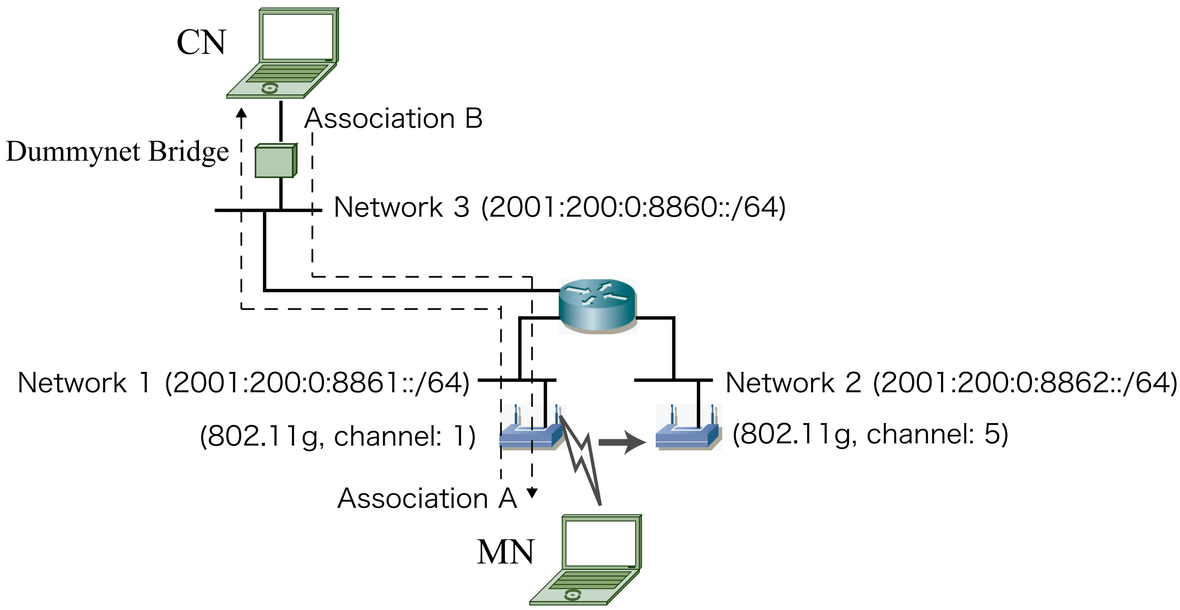


Figure 5.1: Experimental Network

The MN sends 1408 bytes user data to the CN at 50 ms interval over *Association A*, and the CN sends 1408 bytes user data to the MN at 50 ms interval over *Association B*. TSN growth is monitored at receiver nodes. To perform handover process, the MN changes the attached network from *Network 1* to *Network 2*. After this movement, the MN receives a new network address prefix from the router in *Network 2*.

5.1 With DAD Procedure

5.1.1 Association A

Figure 5.2 plots the TSN growth in *Association A*, when the MN executed DAD. The horizontal axis represents a time scale, and the vertical axis represents the TSN. In both of SmSCTP and normal SCTP, the network layer disconnection caused by handover occurs at 2.5 seconds. In the normal SCTP, the MN configures a new address after the 2.343-second network layer disconnection. The MN sends an ASCONF chunk containing ADD IP

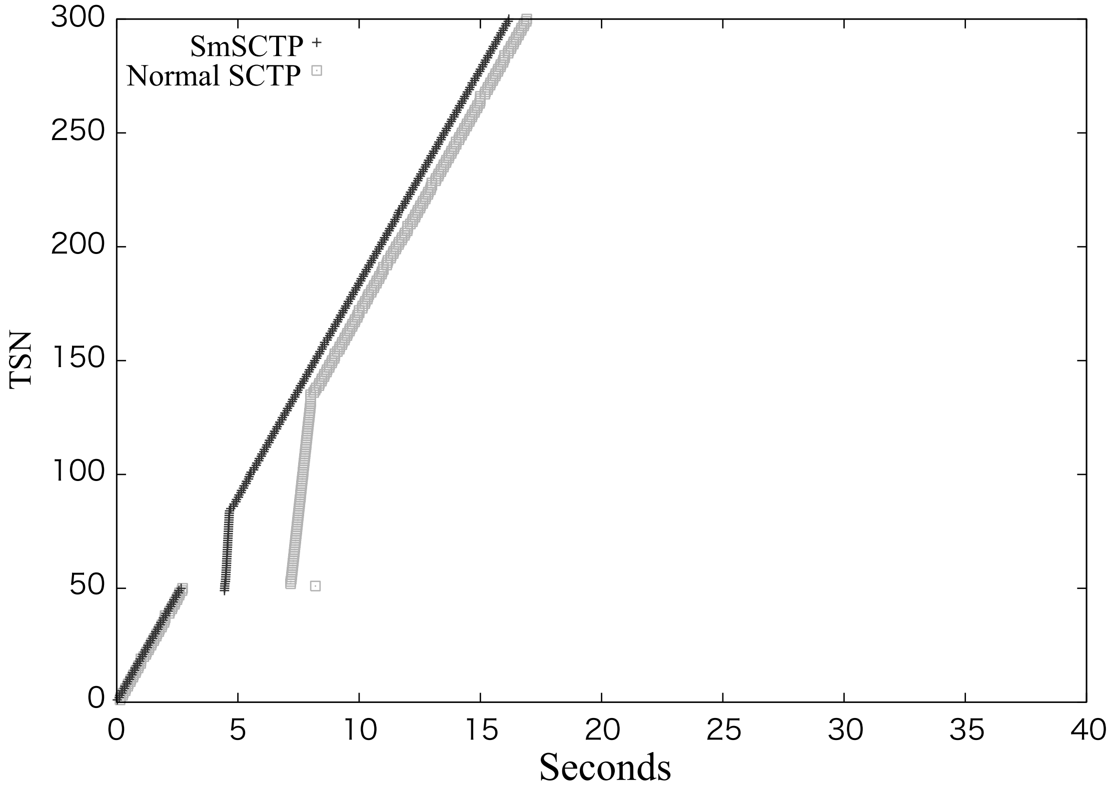


Figure 5.2: TSN growth in Association A with DAD

ADDRESS parameter (ADDIP) 2.001 seconds after the new address configuration. 1.025 seconds after the reception of an ASCONF-ACK for the ADDIP, the MN sends a DATA chunk which is the next TSN of the chunk that the CN received the most recently before the handover. However, the transmission of this chunk is delayed in the MN until the MN's RTO expiration or reception of 3 SACKS which report same gap TSNs. Although other TSN chunks are sent immediately after the reception of the ASCONF-ACK for the ADDIP, they are not passed to the application until the reception of the gap TSN when strict data sequence is specified for this communication. Therefore, the retransmission delay can cause undesirable results on some applications.

In SmSCTP, the MN configures a new address after the 1.766 seconds network layer disconnection. The MN sends an ADDIP 0.002 seconds after the new address configuration. The MN sends a DATA chunk which the CN does not received yet, 0.011 seconds after MN's reception of the ASCONF-ACK. Before this DATA transmission, 2 duplicate chunks are sent from the MN. This is because SACKs from the CN for these TSN chunks are lost during the MN's network layer disconnection or during the exchange of ASCONF messages, or the MN retransmit these TSN chunks before the reception of SACKs for these TSN chunks. Additionally, ASCONF chunk including SET PRIMARY parameter (SETPRIM) is sent 0.021 seconds after the reception of the ASCONF-ACK for the ADDIP.

Table 5.1 shows 10-time averages of this experiment. *Total handover latency* is CN-side sequence discontinuity. *Network layer disconnection* is MN's network layer disconnected time caused by connecting to the new wireless network and configuration of the new address. *Transport layer latency* is CN-side sequence discontinuity without MN's network layer disconnected time. *ADDIP sent time* represents the time when the ADDIP is sent at MN-side after the network layer disconnection. *Sender sequence recovery* time is MN-side DATA sequence recovery time after the reception of ASCONF-ACK for ADDIP.

Table 5.1: Elements of latency Association A with DAD (10-time averages)

	SmSCTP	Normal SCTP
Total handover latency	**2.149 s**	5.747 s
Network layer disconnection	**2.087 s**	2.413 s
Transport layer latency	**0.062 s**	3.334 s
ADDIP sent time	**0.002 s**	2.202 s
Sender sequence recovery time	**0.007 s**	1.065 s

Since the MN performs DAD, the network disconnection period is relatively long (2 - 2.5 seconds). The MN sends ADDIP 0.002 and 2.202 seconds after the network layer disconnection in SmSCTP and the normal SCTP, respectively. This performance improvement is attained by *Fast Association Reconfiguration* in our scheme. As a result, *Fast Association Reconfiguration* shortens the handover latency 2.2 seconds. Similarly, the MN sends new DATA chunks, which the CN does not receive before, 0.007 and 1.065 seconds after the reception of an ASCONF-ACK for the ADDIP in SmSCTP and the normal SCTP, respectively. This performance improvement is realized by our *Fast Transmission Recovery* algorithm in MN-side. Consequently, MN-side *Fast Transmission Recovery* shrinks the handover latency 1.058 seconds. As a whole, 3.282-second reduction of the transport layer handover latency is achieved by SmSCTP.

5.1.2 Association B

Figure 5.3 plots the TSN growth in *Association B*. Same as the previous experiment, the MN performs DAD procedure. In both of SmSCTP and normal SCTP, the network layer disconnection caused by handover occurs at 2.5 seconds. In the normal SCTP, the MN configures a new address after the 1.960-second network layer disconnection. The MN sends an ADDIP 1.002 seconds after the new address configuration. 1.000 seconds after the reception of an ASCONF-ACK for the ADDIP, the MN transmits a SETPRIM to the CN. However, even after the reception of the SETPRIM, the CN sends remained DATA chunks in the send queue to the previous primary destination. Therefore, the CN continues transmission failure of the DATA chunks. These DATA chunks are sent to the new primary destination when they are retransmitted. Hence, long delay (11.032 seconds) is occrred from CN's reception of SETPRIM to CN's continuous transmission to the new primary destination.

In SmSCTP, the MN configures a new address 1.892-second after the network layer disconnection. The MN sends an ADDIP 0.002 seconds after the configuration of the new adrress. 0.001 seconds after the MN's reception of an ASCONF-ACK for the ADDIP, the MN sends

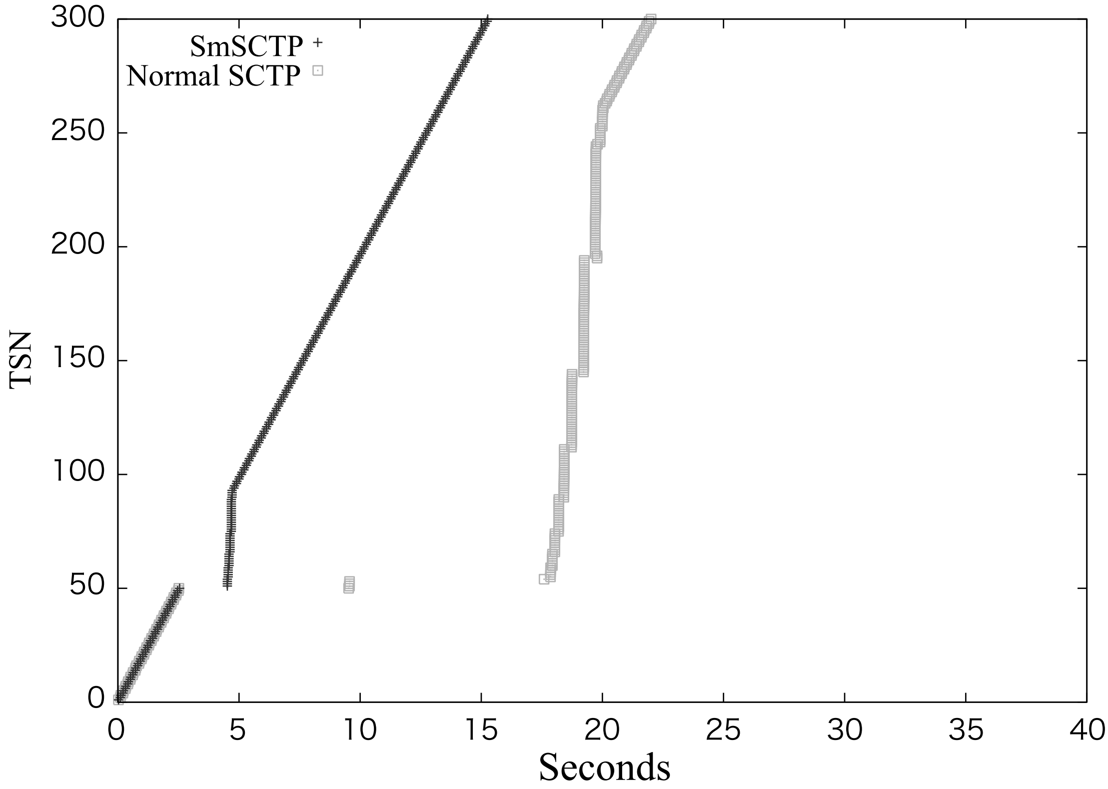


Figure 5.3: TSN growth in Association B with DAD

a SETPRIM. 0.0001 seconds after the CN's reception of the SETPRIM, the CN retransmits DATA chunks which the CN is not acknowledged by the reception of SACKs yet. At the same time, the CN starts continuous transmission of the DATA chunks which remain in the send queue to the new primary destination.

Table 5.2 shows 10-time averages of this experiment. *Total handover latency* is MN-side sequence discontinuity. *Transport layer latency* is MN-side sequence discontinuity without MN's network layer disconnected time. *ADDIP sent time* represents the time that the ADDIP is sent after the MN's network layer disconnection. *SETPRIM sent time* represents the time when the MN sends SETPRIM after the reception of ASCONF-ACK for the ADDIP. *Sender sequence recovery time* is the time when the CN starts continual DATA transmission to the MN's new address.

Table 5.2: Elements of latency Association B with DAD (10-time averages)

	SmSCTP	Normal SCTP
Total handover latency	**2.024 s**	17.388 s
Network layer disconnection	**1.945 s**	2.186 s
Transport layer latency	**0.079 s**	16.202 s
ADDIP sent time	**0.002 s**	1.002 s
SETPRIM sent time	**0.001 s**	1.000 s
Sender sequence recovery time	**0.0009 s**	13.883 s

Same as the previous experiment, because of DAD, relatively long (1.945 - 2.186 seconds) network layer disconnection arises. The MN sends ADDIP 0.002 and 1.002 seconds after the network layer disconnection in SmSCTP and the normal SCTP, respectively. After that, the MN sends SETPRIM 0.001 and 1.000 seconds after the reception of the ASCONF-ACK for the ADDIP in SmSCTP and the normal SCTP, respectively. These performance improvement is attained by *Fast Association Reconfiguration* in our scheme. As a result, *Fast Association Reconfiguration* shortens the handover latency 1.999 seconds in this experimental case.

Similarly, the CN starts the continuous DATA transmission to the MN's new address 0.0009 and 13.883 seconds after the reception of the SETPRIM. Note the start of continuous DATA transmission does not includes duplicate TSNs for the MN. This performance improvement is realized by our CN-side *Fast Transmission Recovery* algorithm. As a result, CN-side *Fast Transmission Recovery* shrinks the handover latency 13.882 seconds. As a whole, 16.123-second reduction of the transport layer handover latency is achieved by SmSCTP.

5.2 Without DAD Procedure

5.2.1 Association B

Figure 5.4 plots the TSN growth in *AssociationA*, when the MN does not performed DAD. Same as the previous experiments, in both of SmSCTP and normal SCTP, the network layer

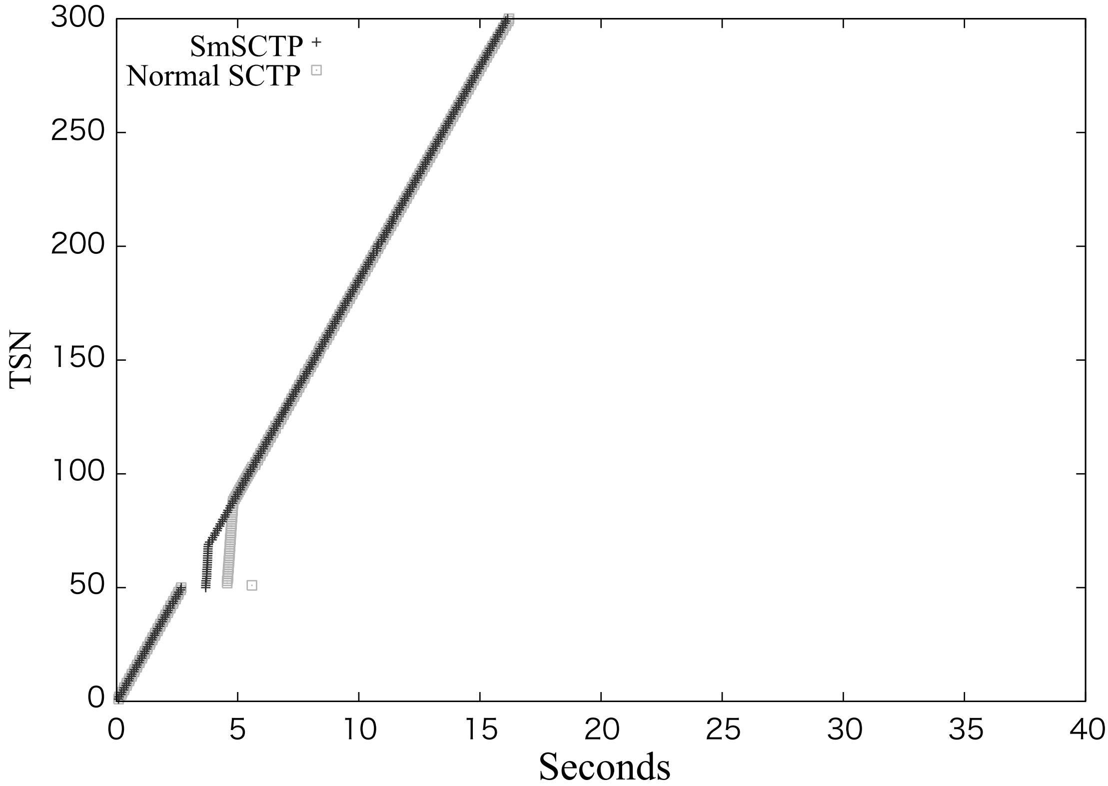


Figure 5.4: TSN growth in Association A without DAD

disconnection occurs at 2.5 seconds. In the normal SCTP, the MN configures a new address after the 0.796-second network layer disconnection. Since MN does not perform DAD, the disconnected time is shorter than the disconnected time with DAD. The MN sends an ADDIP 1.002 seconds after the new address configuration. 1.025 seconds after the reception of an ASCONF-ACK for the ADDIP, the MN sends a DATA chunk which is the next TSN of the chunk that the CN received the most recently before the handover. Same as Figure 5.2, the delay of this chunk increases handover latency.

In SmSCTP, the MN configures a new address after the 0.968 seconds network layer disconnection. Same as the normal SCTP case, the disconnected time is shorter than it with DAD. The MN sends an ADDIP 0.002 seconds after the new address configuration. The MN transmits a DATA chunk which the CN does not received yet, 0.005 seconds after MN's reception of the ASCONF-ACK for the ADDIP. Before this DATA transmission, a duplicate chunk is sent from the MN, because of the loss of a SACK which reports the reception of this DATA chunk during the handover process, or MN's retransmission of these TSN chunks before the reception of SACKs for these TSN chunks.

Table 5.3 shows 10-time averages of this experiment. The MN sends ADDIP 0.002 and

Table 5.3: Elements of latency Association A without DAD

	SmSCTP	Normal SCTP
Total handover latency	**1.063 s**	3.557 s
Network layer disconnection	**0.990 s**	0.741 s
Transport layer latency	**0.072 s**	2.816 s
ADDIP sent time	**0.002 s**	1.202 s
Sender sequence recovery time	**0.006 s**	1.146 s

1.202 seconds after the network layer disconnection in SmSCTP and the normal SCTP, respectively. Same as the experiment with DAD, This performance improvement is attained by *Fast Association Reconfiguration* in SmSCTP. As the result, *Fast Association Reconfiguration* shrinks the handover latency 1.200 seconds. Similarly, the MN sends new DATA chunks, which the CN does not receive before, 0.006 and 1.146 seconds after the reception of an ASCONF-ACK for the ADDIP in SmSCTP and the normal SCTP, respectively. Same as the experiment with DAD, this performance enhancement is achieved by MN-side

Fast Transmission Recovery algorithm. As the result, MN-side *Fast Transmission Recovery* shrinks the handover latency 1.140 seconds. As a whole, 2.744-second reduction of the transport layer handover latency is achieved by SmSCTP.

5.2.2 Association B

Figure 5.5 plots the TSN growth in *Association B*, when the MN does not perform DAD. Same as the previous experiment, in both of SmSCTP and normal SCTP, the network layer

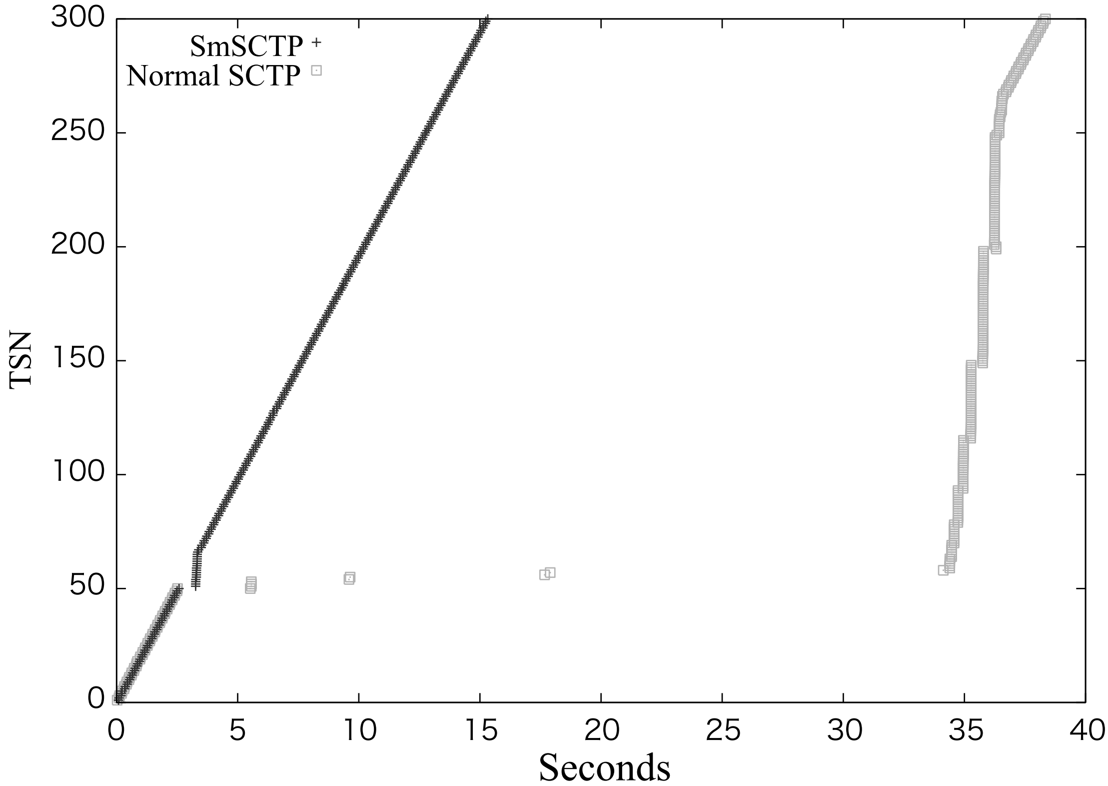


Figure 5.5: TSN growth in Association B without DAD

disconnection occurs at 2.5 seconds. In the normal SCTP, the MN configures a new address after the 0.525-second network layer disconnection. The MN sends an ADDIP 1.001 seconds after the new address configuration. 1.000 seconds after the reception of an ASCONF-ACK for the ADDIP, the MN transmits a SETPRIM. Same as Figure 5.3, the CN sends remained

DATA chunks in the send queue to the previous primary destination, the CN continues transmission failure. Therefore, long delay (28.611 seconds) is occurred from CN's reception of SETPRIM to CN's continuous transmission to the new primary destination.

In SmSCTP, the MN configures a new address after the 0.624-second network layer disconnection. The MN sends an ADDIP 0.002 seconds after the configuration of the new address. 0.0007 seconds after the MN's reception of an ASCONF-ACK for the ADDIP, the MN sends a SETPRIM. 0.0001 seconds after the CN's reception of the SETPRIM, the CN retransmits DATA chunks which the CN is not acknowledged by the reception of SACKs yet. At the same time, the CN starts continuous transmission of the DATA chunks which remain in the send queue to the new primary destination.

Table 5.4 shows 10-time averages of this experiment. Same as previous experiment, *Network layer disconnection* is shorter than it with DAD. The MN sends ADDIP 0.002 and 1.001 seconds after the network layer disconnection in SmSCTP and the normal SCTP, respectively. Besides that, the MN sends SETPRIM 0.0008 and 1.000 seconds after the reception of the ASCONF-ACK for the ADDIP in SmSCTP and the normal SCTP, respectively. As a result, *Fast Association Reconfiguration* shrinks the handover latency 1.998 seconds in this experimental case.

Table 5.4: Elements of latency Association B without DAD

	SmSCTP	Normal SCTP
Total handover latency	**0.709 s**	31.612 s
Network layer disconnection	**0.625 s**	0.828 s
Transport layer latency	**0.084 s**	30.784 s
ADDIP sent time	**0.002 s**	1.001 s
SETPRIM sent time	**0.0008 s**	1.000 s
Sender sequence recovery time	**0.001 s**	27.384 s

Similarly, the CN starts the continuous DATA transmission to the MN's new address which the CN does not receive yet, 0.001 and 27.384 seconds after the reception of the SETPRIM. This performance improvement is realized by our CN-side *Fast Transmission Recovery* algorithm. As a result, CN-side *Fast Transmission Recovery* shrinks the handover latency 27.383 seconds. As a whole, 30.700-second reduction of the transport layer handover latency is achieved by SmSCTP.

Chapter 6

Conclusion and Future Work

This chapter describes a summary of this research. Additionally, we describes future work of it to the advanced research.

In this paper, we discussed the issues in the handover mechanism of SCTP and proposed a new transport layer handover algorithm called SmSCTP. SmSCTP can improve communication performance during handover when a mobile node has only one WNIC. If a mobile node can have multiple WNICs, it might be possible to reduce handover latency by utilizing multiple WNICs simultaneously. However, this will consume more battery or require more device installation space and might increase hardware costs. Thus, we believe handover methods for a single WNIC will be important in future mobile environments.

SmSCTP distinguishes communication paths based on a pair of source and destination addresses while normal SCTP distinguishes communication paths based on only destination address. Hence, SmSCTP can react to the change of source address immediately and can achieve smooth handover process without causing unnecessary packet retransmissions. We conducted some experiments with our scheme and verified that our scheme can reduce transmission latency by two to thirty seconds after the disconnection period caused by handover. When the hardware, radio and network layer technologies advance, our scheme can become more practical.

As a future work, we plan to propose our scheme to IETF, as a part of the SCTP specification. We also plan to conduct some experiments that combine SmSCTP with Mobile IPv6 that can provide location management features.

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Hideyuki Tokuda, for his technical and professional advice, guidance, and encouragement.

I would like to thank Professors Jun Murai, Osamu Nakamura, Hiroyuki Kusumoto, Kazunori Takashio and Ryuji Wakikawa for their valuable and technical comments on this thesis.

I would like to be extremely thankful to Dr. Yoshifumi Nishida at Sony Computer Science Laboratory for his considerate support, valuable advice and discussions on my all successive research activities.

I would like to be thankful to Masahiro Kozuka in SCTP working group at WIDE project for their technical discussions such as my implementation.

I am grateful to Dr. Jin Nakazawa and Dr. Hideaki Imaizumi for their daily advanced technical supports since I have started studying and researching. I am also grateful to Dr. Masayuki Iwai, Masato Saito and Hitomi Takahashi.

I appreciate members of move! research group in Hide Tokuda Laboratories, Dr. Tomohiro Nagata, Shigeru Moriwake, Hiroshi Sakakibara, Masao Ideuchi, Takuro Yonezawa, Kengo Koizumi, Shingo Miyajima, Hikoichiro Nakai, Kohei Funaki and Tomohiro Ito for daily productive and pleasant discussions.

I also appreciate same generation members of Tokuda, Murai, Nakamura, Kusumoto, Takashio and Wakikawa Laboratories, especially Masayoshi Mizutani, Takashi Tomine, Yuki

Oyabu, Yuichi Nakamura, Hidetoshi Tokuda, Mizuki Kawazoe, Eriko Tsuda, Tomomi Nakamura, Sayaka Ogura, Akira Kanai, Yohei Kuga, Yusuke Okumura, Yuri Nagai and Megumi Nakazato.

Finally, I appreciate invaluable friends, Masashi Horiguchi, Shinsuke Jibiki, Kennichi Inagaki, Shotaro Suzuki, Miyuki Kanai, Miho Imase, Kaori Kan and Takae Miyamoto for pleasant private.

February 1, 2007

Michio Honda

References

Published Papers Related to this Thesis

- 本多倫夫, 榊原寛, 徳田英幸

  “小型端末のための無線ネットワーク間におけるハンドオーバ機構, ”

  情報処理学会 マルチメディア通信と分散処理研究会 (DICOMO),

  Jul. 2006.

  Winner of the Young Researcher's Award

Posters and Demos

- 本多倫夫, 榊原寛, 徳田英幸

  “SmSCTP: Efficient Handover Management Mechanism with SCTP for Single-home Environment,”

  日本ソフトウェア学会 SPA X,

  Aug. 2006.

- 本多倫夫, 榊原寛, 徳田英幸

  “SHINE: 小型端末のための，無線ネットワーク間におけるハンドオーバ機構,”

  日本ソフトウェア学会 SPA 2006,

  Feb. 2006.

- 本多倫夫, 榊原寛, 白畑真, 徳田英幸

  “SCTP を用いた単一無線インタフェースによる複数無線ネットワーク間のローミン

グ機構,"

日本ソフトウェア学会 第 4 回 SPA サマーワークショップ,

Aug. 2005.

- 丸山伸, 小塚真啓, 小野祐介, 本多倫夫

  "SCTP を用いたシングルインタフェイスでのハンドオーバ実験,"

  Internet Conference 2006,

  Oct. 2006.

Bibliography

[1] FreeBSD http://www.freebsd.org/.

[2] IPv6 http://www.ipv6.org/.

[3] Ministry of Public Management, Home Affairs, Posts and Telecommunications. 2006 WHITE PAPER Information and Communications in Japan.

[4] SCTP kernel implementation. http://www.sctp.org/.

[5] WiMAX Forum. http://www.wimaxforum.org/.

[6] 3rd Generation Partnership Project. *High Speed Downlink Packet Access: Physical Layer Aspects, v 5.0.0, http:/www.3gpp.org.* 3GPP, 2002.

[7] 3rd Generation Partnership Project 2. 1xEV-DO Inter-Operability Specification (IOS) for CDMA 2000 Access Network Interfaces, June 2001.

[8] Claude Castelluccia. HMIPv6: A hierarchical movile IPv6 protocol. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2000.

[9] D. Johnson, C. Perkins, J. Arkko. Mobility Support in IPv6. *RFC 3775*, Jun. 2004.

[10] D. Funato, K. Yasuda, and H. Tokuda. Tcp-r: Tcp mobility support for continuous operation. *icnp*, 00:229, 1997.

[11] Hung-Yun Hsieh, Kyu-Han Kim, and Raghupathy Sivakumar. An end-to-end approach for transparent mobility across heterogeneous wireless networks. *MONET*, 9(4):363–378, 2004.

[12] IEEE. IEEE 802.20 Mobile Broadband Wireless Access (MBWA).

[13] IEEE. The IEEE 802.16 Working Group http://grouper.ieee.org/groups/802/16/.

[14] IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards comittee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Futher Higher-Speed Physical Layer Extension in the 2.4 GHz Band*. IEEE, Feb 2003.

[15] IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz band*, Feb 2000.

[16] IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee). *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed Physical Layer extension in the 2.4 GHz band*. IEEE, Feb 2000.

[17] Kyocera. iBurst Broadband Wireless http://global.kyocera.com/prdct/telecom/office/iburst, 2004.

[18] M. Allman, V. Paxson and et al. TCP Congestion Control. *RFC 2581*, Oct. 1999.

[19] M. K. Abdul Aziz and A. R. Nix And P. N. Fletcher. *A Study of Performance and Complexity for IEEE 802.11n MIMO-OFDM GIS Solusions*. IEEE Communications Society, 2004.

[20] Maltz. D. A and Bhagwat. P. MSOCKS: an architecture for transport layer mobility. *IEEE INFOCOM*, 3:1037–1045, 1998.

[21] Junwen Lai Ming Zhang and et al. A transport layer approach for improving end-to-end performance and robustness using redundant paths. *USENIX 2004 Annual Technical Conference*, pages 99–112, 2004.

[22] N. Moore. Optimistic Dupricate Address Detection (DAD) for IPv6. *RFC 4429*, Apr. 2006.

[23] R. Koodli. Fast Handovers for Mobile IPv6. *RFC 4068*, Jul. 2005.

[24] R. Stewart, M. Ramalho and et al. Stream Control Transmission Protocol (SCTP) Partial Reliability Extention. *RFC 3758*, May. 2004.

[25] R. Stewart, M. Ramalho, Q. Xie and et al. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. *Internet Draft*, Aug. 2003.

[26] R. Stewart, Q. Xie and et al. Stream Control Transmission Protocol. *RFC 2960*, Oct. 2000.

[27] S. Deering, R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. *RFC 2460*, Dec. 1998.

[28] S. J. Koh, H. Y. Jung, J. H. Min. Transport Layer Internet Mobility based on mSCTP. *IEEE ICACT*, 1:329–333, 2004.

[29] Alex C. Snoeren and Hari Balakrishnan. An end-to-end approach to host mobility. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 155–166, New York, NY, USA, 2000. ACM Press.

[30] T. Okoshi, M. Mochizuki, Y. Tobe and H. Tokuda. MobileSocket: Toward Continuous Operation for Java Applications. In *Proceedings of IEEE 8th International Conference on Computer Communications and Networks*, pages 50–57, 10 1999.

[31] Flarion Technologies. Flash-OFDM Whitepaper http://www.flarion.com/.

[32] Chien-Chao Tseng, Yung-Chang Wong, Li-Hsing Yen, and Kai-Cheng Hsu. Proactive dad: A fast address-acquisition strategy for mobile ipv6 networks. *IEEE Internet Computing*, 10(6):50–55, 2006.