

# WLAN에서 RA 선수신을 이용한 mSCTP 핸드오버 성능 향상

## (Performance Enhancement of Handover in mSCTP using Pre-acquisition RA in WLAN)

최순원<sup>†</sup>      김광렬<sup>†</sup>      민성기<sup>\*\*</sup>  
(Soon-won Choi) (Kwang-ryoul Kim) (Sung-gi Min)

**요약** mSCTP(Mobile SCTP)는 SCTP(Stream Control Transmission Protocol)와 DAR(Dynamic Address Reconfiguration) 확장을 이용해 트랜스포트 계층에서 이동성을 제공하기 위해 최근에 제안된 프로토콜이다. 현재 mSCTP의 가장 큰 문제점은 핸드오버 지연시간(Handover Latency Time)이 길어 실시간 어플리케이션에 적합하지 못하며 핸드오버 시점을 결정(Handover Decision)하는 명시적인 방법이 없다는 것이다. 이를 해결하기 위해 본 논문에서는 RA(Router Advertisement)를 미리 수신함으로써 핸드오버 지연시간을 최소화 하는 방법을 제안하고자 한다. 또한, RA를 빠르게 얻을 수 있는 방법으로 RA 캐시(Cache)를 이용한 방법, FMIPv6(Fast Handovers for Mobile IPv6)와 연동을 통한 방법, 듀얼 인터페이스(Dual Interface)를 이용한 방법 등 세가지를 소개 한다. 마지막으로, 제안된 방안의 성능 향상을 확인 하기 위해 세가지 방식 중 FMIPv6와의 연동 방식을 리눅스 기반의 실제 시험 환경을 구성하여 핸드오버 성능을 측정하였다. 실험한 결과 전체 핸드오버 지연시간의 대부분이 RA를 받는데 있음을 확인 할 수 있었고, 이를 줄임으로써 전체 핸드오버 성능을 크게 향상시킬 수 있었다.

**키워드** : mSCTP, 핸드오버, 이동성, 트랜스포트 프로토콜

**Abstract** The SCTP (Stream Control Transmission Protocol) implementation with the DAR (Dynamic Address Reconfiguration) extension is called the mSCTP (Mobile SCTP) that is proposed recently for mobility support in transport layer. The mSCTP does not satisfy short handover latency for real-time applications and it has no specific handover decision mechanisms. In this paper, we propose fast handover schemes for mobile nodes that are moving into different subnet using pre-acquisition RA (Router Advertisement) and L3 trigger for improving handover performance. Furthermore, we introduce three specific methods which are RA cache, FMIPv6 (Fast Handovers for Mobile IPv6) and dual interface and how proposed scheme can be interoperated with handover process respectively. Finally, we show two experimental results which are the mSCTP and the mSCTP using FMIPv6 on Linux platforms. Experimental results show that handover performance is improved with reducing the time of receiving RA which takes most of total handover latency.

**Key words** : mSCTP, handover, mobility, transport protocol

### 1. 서론

최근 Mobile IP를 시작으로 IP의 이동성 연구가 활발히 진행되고 있다. 차세대 네트워크에서는 기존의 WLAN, 셀룰러 네트워크뿐만 아니라, WiBro(Wireless Broad-

band)와 같은 새로운 무선망이 IPv6를 기반으로 All-IP 네트워크 형태로 발전하게 된다. 이에 따라 IP의 이동성이 더욱 중요해 지고 있다. 이와 함께 다양한 멀티미디어 서비스의 요구가 증가되기 때문에 실시간 어플리케이션에 적합한 짧은 핸드오버 지연시간이 매우 중요한 요구사항 중 하나이다. 또한, 동일 망에서의 이동성뿐 아니라 이 기종 망 사이에서 이동성의 필요성이 증가될 것이다.

하지만 MIPv6의 경우 긴 핸드오버 지연시간을 갖게 되므로 실시간 어플리케이션에는 적합하지 못하다. 이를

<sup>†</sup> 비회원 : 고려대학교 컴퓨터학과  
bluemyrna@korea.ac.kr  
biofrog@korea.ac.kr

<sup>\*\*</sup> 정회원 : 고려대학교 컴퓨터학과 교수  
sgmin@korea.ac.kr

논문접수 : 2005년 10월 22일  
심사완료 : 2006년 1월 4일

해결하기 위해 FMIPv6 [1]과 HMIPv6 [2]이 제안되었다. 하지만 이러한 방법들은 네트워크에 이를 지원하는 기능이 추가되어야 한다.

최근에 이와는 다른 관점에서 SCTP를 이용한 트랜스포트 계층에서의 이동성 제공에 대한 연구가 시작되었다. 그 결과, SCTP의 멀티호밍(Multi homing)과 DAR 확장을 이용한 mSCTP가 제안되었고 이를 이용하면 SCTP를 사용하여 트랜스포트 계층에서 종단간 이동성을 지원할 수 있다. 하지만 이 역시 핸드오버 지연 시간이 긴 문제점과, 핸드오버 결정 방식이 명시적이지 못한 문제점을 가지고 있다.

본 논문에서는 mSCTP의 핸드오버 지연시간을 최소화 할 수 있는 방안으로 RA 캐시를 이용한 방법, FMIPv6와 연동을 통한 방법, 듀얼 인터페이스를 이용한 방법을 제안 한다. 또 제안된 방법의 성능 향상을 확인하기 위해 리눅스 기반의 실제 시험 환경을 구성하여 성능을 측정하고 분석한다.

본 논문은 다음과 같이 구성된다. 2절에서는 IP계층의 이동성에 관한 연구와 이에 대한 문제점을 기술한다. 3절에서는 현재 mSCTP에서의 이동성 지원 문제점에 대해서 기술한다. 4절에서는 이를 해결하기 위한 방안에 대해 기술한다. 5절에서는 이를 이용한 실험 환경과 결과를 기술하고, 이에 대한 성능 분석결과를 기술한다. 마지막으로 6절에서는 결론을 기술한다.

## 2. 관련 연구

MIPv6는 이동 단말이 서로 다른 IPv6 서브넷으로 이동 후에도 통신이 지속적으로 이루어 질 수 있도록 이동성을 제공해 주는 네트워크 계층의 프로토콜이다 [4,5].

네트워크 계층에서 이동성을 제공하면 모든 상위 계층 프로토콜에 투명한 이동성을 제공한다. 하지만 네트워크 계층에서 이동성을 지원하기 위해 MIPv6를 사용하게 되면 특별한 기능의 라우터(Home Agent: HA, Foreign Agent: FA)가 필요하게 되고 터널링(Tunneling), 삼각 라우팅(Triangular Routing) 문제와 이로 인한 오버헤드가 발생하게 된다. 또, 망에 따라 MIPv4와 MIPv6를 써야 하는 망 종속적인 문제를 가지고 있다. 그리고 긴 핸드오버 지연시간으로 인해 실시간 어플리케이션에는 적합하지 못한 단점을 가진다.

최근에는 MIPv6환경에서 실시간 어플리케이션에 적합하도록 FMIPv6 프로토콜이 제안되었다. FMIPv6는 MIPv6에서 핸드오버 지연시간으로 인한 패킷 손실과 오버헤드, 이로 인한 긴 핸드오버 지연시간을 최소화 한다. FMIPv6의 사용으로 핸드오버 기능은 FMIPv6이 담당하므로 핸드오버 지연시간이 줄어들게 되었지만 이

동성 지원은 여전히 MIPv6에 기반하고 있다. FMIPv6의 단점은 네트워크에서 이를 지원해 줘야 한다는 점인데, 여러 서브넷을 빠르게 핸드오버할 경우 계속해서 터널링을 하게 되고 이에 대한 실제 경로 최적화는 MIPv6의 BU(Binding Update)에 의지하고 있다[1].

최근에 이와는 다른 관점에서 트랜스포트 계층에서 이동성을 제공하기 위한 연구가 진행 되었고, SCTP [6,7]에 DAR(Dynamic Address Reconfiguration)[8]을 추가하여 종단간 이동성을 제공하도록 SCTP를 확장한 프로토콜인 mSCTP (Mobile SCTP)가 제안되었다 [9,10].

SCTP는 호스트에 탑재된 어플리케이션들끼리 종단간에 통신을 제공하는 트랜스포트 계층의 프로토콜이다. SCTP는 연결지향적이고, 신뢰성 있는 연결을 제공하며 TCP와 같이 혼잡제어 방식 등을 제공한다. 특히, 멀티 스트리밍(Multi-Streaming)과 멀티호밍은 기존 TCP와 SCTP를 구분하는 가장 중요한 특징이다. 멀티스트리밍은 하나의 연결 내에서 여러 개의 스트림(Stream)이 존재하는 형태이다. 멀티호밍은 SCTP는 양쪽 종단 점에서 여러 개의 IP주소를 가지고 연결설정을 지원한다. 이중 하나는 주 경로(Primary Path)로 데이터 전달을, 다른 하나는 부 경로(Secondly Path)로서 에러로 인한 재전송을 위해 사용된다. SCTP는 하트비트(heartbeat)를 사용하여 연결상태를 모니터링 하고 주 경로의 에러를 탐지 할 수 있다. 이를 통해 에러가 탐지되면 부 경로로 전송 경로를 바꿀 수 있다. 이와 함께 최근에 SCTP에서 실시간 어플리케이션을 위한 연구도 진행되고 있다 [11]. SCTP는 이러한 새로운 특징으로 인해 트랜스포트 계층에서 TCP와 UDP를 대신해 향후 많이 사용될 것으로 기대된다.

그림 1에서 보듯이, mSCTP는 DAR 확장을 통해 SCTP 연결이 유지되면서 종단 지점에 매핑된 IP주소를 동적으로 추가(ASCONF-ADDIP) 혹은 삭제(ASCONF-DELETEIP)할 수 있다.

이동성 관리 측면에서 mSCTP는 대단히 효율적이라 할 수 있다. 양 종단간의 호스트끼리 통신을 하기 위해서 mSCTP세션을 연결하기만 하면 Mobile IP의 FA나 HA같은 특별한 라우터의 도움이나 다른 이동성 지원 프로토콜의 도움 없이도 이동성을 제공해 준다. 또한, IPv4망과 IPv6망을 가리지 않고 사용할 수 있고, 이 기종 망에서의 이동성 제공을 쉽게 해 줄 수 있다[12].

현재 신뢰성 있는 전송을 원하는 대부분의 어플리케이션이 TCP를 사용한다. TCP에서 이동성은 양 종단에서의 IP주소와 포트번호가 바뀌는 것으로 생각할 수 있다. 즉, TCP 바인딩의 변경이라 볼 수 있고, 이러한 관점에서 이동성의 지원은 종단에서의 바인딩 정보 변경

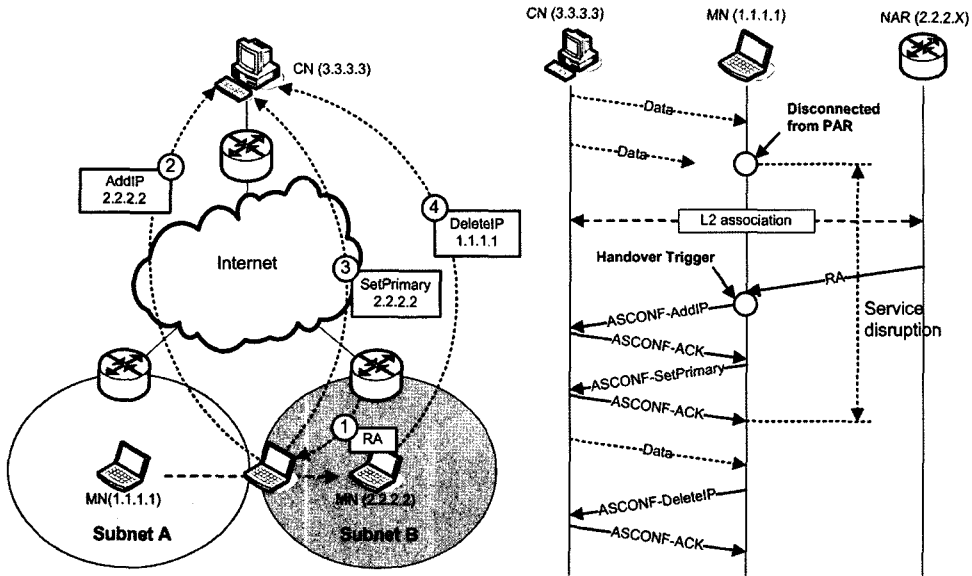


그림 1 mSCTP의 핸드오버 절차

을 필요로 한다. MIPv6의 경우 바인딩 정보를 MIPv6의 헤더를 통해 가지고 다니게 되고 이로 인한 오버헤드가 발생한다. 만약 동적 바인딩 정보 변경을 통해 mSCTP와 같은 트랜스포트계층에서 이동성을 지원하게 된다면 특별한 기능의 라우터(HA, FA)들이 필요 없게 되며, 터널링에 따른 오버헤드가 발생 하지 않을 것이다. 또한 네트워크 계층에서 이동성 지원이 필요하지 않고, MN이나 CN과 같은 단말에만 이동성 지원 프로토콜이 탑재 되어도 종단간 이동성을 제공하게 된다.

핸드오버 관점에서 살펴보면, mSCTP는 새로운 IP를 얻게 되면 ASCONF-ADDIP 메시지를 통해 CN(Correspondent Node)에게 등록을 하게 된다. 하지만 언제 주 경로를 변경할 지에 대한 명확한 기준이 현재 정해지지 않은 상태이다. 이 때문에 주 경로를 변경하는 문제는 mSCTP의 핸드오버 성능을 결정하는 가장 중요한 요소가 된다. 또한, 긴 핸드오버 지연시간은 실시간 어플리케이션에 적합하지 않은 단점으로 볼 수 있다. 그러므로 현재 mSCTP의 핸드오버 성능을 향상시키기 위해

서는 주 경로를 변경하는 시점을 결정하는 문제와 핸드오버 지연시간을 줄이는 문제에 대한 해결이 필요하다.

### 3. mSCTP에서 핸드오버에 관한 문제점

#### 3.1 핸드오버 지연시간

앞 절에서도 보았듯이 핸드오버 지연시간은 현재 mSCTP에서 가장 중요한 문제 중 하나 인데, mSCTP 핸드오버 지연 시간은 다음 그림 2와 같이 3가지 구간으로 구분하여 생각할 수 있다.

1. 링크계층 핸드오버 지연시간 ( $\Delta 1$ ): 이 구간은 새로운 서브넷으로 이동 시 링크계층 구성하거나 물리계층의 채널을 구성하는데 걸리는 시간이다.
2. IP 계층 구성 지연시간 ( $\Delta 2$ ): 이 구간에서는 NAR(New Access Router)로부터 RA를 받은 후 새로운 IP를 생성하는 구간이다. MN은 주기적으로 받는 RA를 통해 새로운 서브넷 구간임을 인식하고 NCoA(New CoA)를 생성하게 된다.
3. 바인딩 정보 갱신 지연시간 ( $\Delta 3$ ): MN은 NCoA를

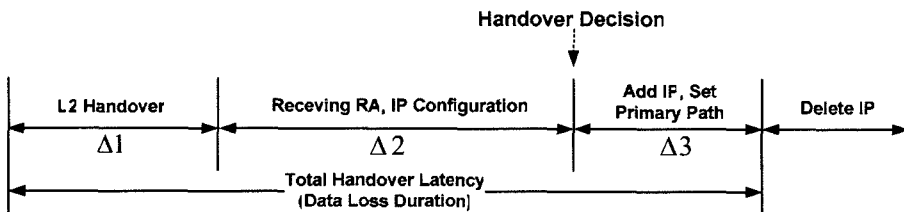


그림 2 전체 핸드오버 지연시간

CN에 등록하고 주 경로를 변경하여 전체 핸드오버 동작을 완료한다. 이 절차는 SCTP DAR 확장의 ASCONF 메시지에 포함된 ADDIP 청크(Chunk)와 SetPrimary 청크를 이용하여 수행된다.

△1의 경우 링크 계층이나 하드웨어 의존적이므로 본문에서는 다루지 않도록 하겠다. △2의 경우 핸드오버 지연시간뿐 아니라 핸드오버 시점을 결정하는데 큰 영향을 끼친다. MN이 새로운 서브넷으로 이동했는지 여부는 주기적으로 받는 RA에 포함된 네트워크 IP를 통해서 알 수 있다. △2에서의 지연시간은 RA를 받는 주기에 많은 영향을 받는다. 그러므로 RA를 어떻게 빨리 받느냐가 지연시간을 줄임과 동시에 핸드오버 시점을 결정 하는데 중요한 요소가 된다. △3의 경우 ASCONF-ADDIP 메시지를 통해 먼저 등록하고 핸드오버 시점에서 ASCONF-SetPrimary 메시지를 통해 주 경로를 바꿔 줄 수도 있고, 핸드오버 시점에서 둘을 하나의 메시지에 넣어서 보낼 수도 있다.

### 3.2 명시적 핸드오버 시점 결정 방안

MN이 새로운 서브넷으로 이동할 경우, MN은 특별한 규칙에 의해 새로운 IP주소로 주 경로는 바꿔 주어야 한다. 앞서 살펴본 바와 같이 mSCTP에서는 새로운 IP주소를 언제 ADDIP 해주느냐, 그리고 주 경로를 언제 바꿔 주느냐가 핸드오버 성능에 큰 영향을 미치며, 이는 현재 가장 중요한 해결 과제 중 하나이다[3,10]. 특히, 핸드오버를 결정하는 주 경로 변경에 대한 규칙이 뚜렷이 정해져 있지 않고, 이를 정의하는 것이 주요 논쟁으로 남아 있다. 현재 이에 대한 몇몇 가능한 규칙들을 정리하면 다음과 같다[10].

1. MN이 ASCONF-ADDIP 메시지에 대한 ASCONF-ACK 메시지를 받자마자 주 경로를 변경해 준다.
2. 링크계층이나 물리계층으로부터 주 경로 변경 트리거(Trigger)를 받는다.
3. 상위 계층으로부터 트리거를 받는다.
4. 패킷 손실로 인한 재전송 횟수가 특정 임계 값 이상일 경우 주 경로를 변경한다.

첫 번째 경우를 살펴보면 MN은 먼저 새로운 서브넷으로 이동을 하게 되면 새로운 NCoA를 생성하게 된다. 이 후에 CN쪽에 ASCONF 메시지에 ADDIP 청크를 포함해서 MN의 IP를 등록하게 되고, ASCONF-ACK 메시지를 받자마자 ASCONF에 SetPrimary 청크를 통해 주 경로를 바꿔주게 된다. 혹은 이 두 가지 청크를 하나의 ASCONF 메시지에 포함해서 처리할 수도 있다. 이때 MN은 RA를 받아야 새로운 서브넷으로 이동했는지를 알 수 있고 또한 새로운 IP주소를 생성할 수 있다. RA 메시지는 NAR로부터 주기적으로 수신하게 된다. 이로 인한 통신 두절 현상은 그림 1을 통해 자세히

확인할 수 있다. 위 절차가 제대로 이루어진다면 MN은 SCTP 연결이 유지된 상태에서 계속해서 통신을 할 수 있을 것이다. 만약에 MN이 2개의 서브넷을 빠른 속도로 왕복해서 이동하게 되면 mSCTP는 계속해서 주 경로를 바꾸게 될 것이고, 이로 인한 오버헤드는 핸드오버 성능을 저하 시키게 된다.

두 번째 경우를 살펴보면, 전파 세기 같은 링크 및 물리 계층의 정보를 이용하면 이를 이용하지 않는 방법보다 핸드오버 성능을 높일 수 있음이 알려져 있다 [3]. 하지만 이를 위해서는 트랜스포트 계층에서 링크 계층 혹은 물리 계층으로부터 정보를 받아서 이를 저장하고 이를 평가해서 핸드오버 여부를 판단하는 기능을 하는 독립적인 컴포넌트가 필요하다. 하지만 이러한 컴포넌트 구현이 간단하지 않을 뿐 아니라, 링크 계층이나 물리 계층의 정보만을 가지고 핸드오버 시점을 판단하기가 매우 힘들다. 그리고 프로토콜 계층 구조를 위반하여 정보가 전달되는 문제가 발생하게 된다.

세 번째의 경우는 상위 계층에서 핸드오버 판단을 하고 mSCTP에게 트리거를 보내는 방법이다. 일반적으로 상위 계층에서는 MN의 이동과 관련된 정보 이외에도 핸드오버에 관련된 더 많은 정보를 가지고 있다. 만약 여러 종류의 무선 네트워크들이 있고, MN가 이들 네트워크 사이를 핸드오버 할 경우 상위 계층은 네트워크 대역폭, 비용, 기타 여러 가지 성능 인자들을 고려해서 핸드오버를 결정하게 된다. 이러한 이 기종 망에서의 핸드오버를 수직적 핸드오버(Vertical Handover)라고 부르고 MN의 이동이 없이도 핸드오버가 발생할 수 있다. 그러므로, mSCTP는 어느 때나 상위 계층으로부터 핸드오버 요청 트리거를 받을 수 있고 이를 수행하는 것이 바람직하다[12].

마지막으로 SCTP의 재전송 방식을 이용하는 방법이다. SCTP는 기본적으로 멀티호밍일 경우 패킷 손실로 인한 재전송을 주 경로를 통해 수행하게 된다. 이를 이용해 재전송 횟수가 임계치 이상일 경우 핸드오버를 하는 방법을 고려할 수 있다. 하지만 이를 위해서는 인터페이스가 두 개 이고, 동시에 두 개의 AR과 통신을 하고 있어야 한다. 또한 임계치 이상이 될 때까지 통신이 두절되기 때문에 핸드오버 지연시간이 길어진다[13].

지금까지 살펴본 mSCTP에서 주 경로를 변경하는 문제는 전체 핸드오버 성능에서 또 다른 주요한 문제이고 이것은 하위계층(Layer2, Layer3)의 핸드오버 상태와 밀접한 관련이 있다. 하지만 mSCTP 핸드오버 시점을 알려 주기 위해 2계층 이하의 계층에서 직접 트랜스포트 계층의 mSCTP에게 트리거를 주는 방법은 통신 프로토콜의 계층 구조를 위반하기 때문에 구현이나 적용에서 어렵고 비효율적일 것이다. 그러므로 L2(Layer

2) 핸드오버 이후 네트워크 계층 구성 후 mSCTP에 핸드오버 시작을 알려 주는 L3(Layer 3) 핸드오버 트리거의 정의가 필요하다.

#### 4. RA수신방식 개선을 통한 mSCTP 핸드오버 성능 향상

앞 절에서 살펴 본 바와 같이 mSCTP 전체 핸드오버 지연 구간에서 Δ2 구간을 줄여 주는 것이 핸드오버 성능 향상에 가장 중요한 요소가 된다. 그리고, Δ2 구간에서 대부분을 차지하는 RA수신 시간을 제거 하게 되면 핸드오버 지연시간을 크게 줄일 수 있게 된다. 이와 함께 명시적 핸드오버 결정 방법 또한 필요하다. 본 논문에서는 실제 핸드오버 시점 이전에 미리 NAR의 RA를 수신함으로써 핸드오버 성능을 향상시키는 3가지 방법을 제안하고 명시적 핸드오버 시점 결정에 필요한 L3 트리거를 제안된 각 방법에 대해 정의한다.

##### 4.1 RA 캐시를 이용한 방안

그림 3은 RA 캐시를 이용한 방법을 보여주고 있다. PAR(Previous Access Router)와 NAR은 AR의 서브넷 IP주소, AP의 MAC주소, RA정보를 포함한 RA 캐시를 유지하고 있다. PAR은 MN에게 주기적으로 RA 캐시 정보를 전달하고 MN은 이들 정보를 조합해서 새로운 서브넷에서 사용할 IP주소를 미리 생성한다. 링크 계층 핸드오버가 완료 되어 네트워크 계층에 L2 트리거를 보내면, 네트워크 계층은 여기 포함된 MAC 주소를 가지고 자신의 RA 캐시 테이블에서 일치하는 RA 정보를 찾는다. 이 찾아진 RA 정보를 사용하여 MN은 NAR 정보를 알 수 있고 NCoA를 설정 할 수 있다. 이 과정이 끝나면, 네트워크 계층은 L3 트리거를 mSCTP

에 전달 하고 mSCTP는 CN에 NCoA를 등록하기 위해 ASCONF-ADDIP 메시지를 보내고 주 경로를 갱신하기 위해 ASCONF-SetPrimary 메시지를 보낸다. 이 때 성능 향상을 위해 ASCONF 청크에 ADDIP와 SetPrimary를 함께 보내는 방안도 고려 할 수 있다.

RA캐시를 이용 하는 방법은 MN가 미리 RA 정보를 얻어 NCoA를 구성할 수 있기 때문에 L2 핸드오버가 끝날 때 바로 mSCTP가 핸드오버를 시작할 수 있는 장점이 있다. 하지만 이 방법은 PAR이나 NAR에 RA 캐시 테이블을 유지해야 하며, 이를 교환하기 위한 절차 또한 필요하다. MN 역시 RA 캐시 정보를 유지 관리해야 하며 라우터와 MN 사이에 캐시 정보 교환에 대한 절차 역시 정의 되어야 한다.

##### 4.2 FMIPv6와의 연동을 통한 방안

그림 4는 FMIPv6와 mSCTP의 핸드오버 절차를 나타내고 있다. MN은 RtSolPr(RouterSolicitation for Proxy) 메시지와 PrRtAdv (Proxy Router Advertisement) 메시지 교환을 통해 NAR의 RA정보를 미리 알 수 있다. MN은 이를 이용해 새로운 서브넷에서 사용할 IP주소를 생성하게 되고 이 과정이 완료 되면 mSCTP는 ASCONF-ADDIP 메시지를 통해 CN에 등록 하게 된다. 이 후 FMIPv6의 핸드오버 절차가 진행 되어 MN이 NAR에게 FNA(Fast Neighbor Advertisement) 메시지를 보내는데 이 메시지의 교환은 L2 핸드오버 및 FMIPv6 핸드오버 절차가 완료 되었음을 나타낸다. FMIPv6는 FNA 절차 이 후 mSCTP에 L3 트리거를 전달하여 핸드오버 시작을 알려 주고 mSCTP는 ASCONF-SetPrimary 메시지를 보내어 주 경로를 변경한다. 즉 mSCTP는 FMIPv6 가 생성한 L3 핸드오버

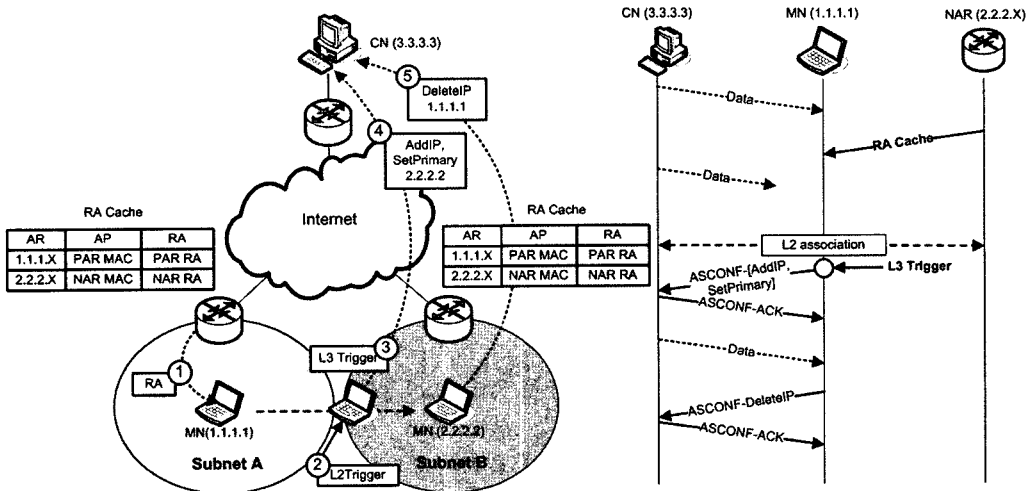


그림 3 RA 캐시를 이용한 mSCTP의 핸드오버 절차

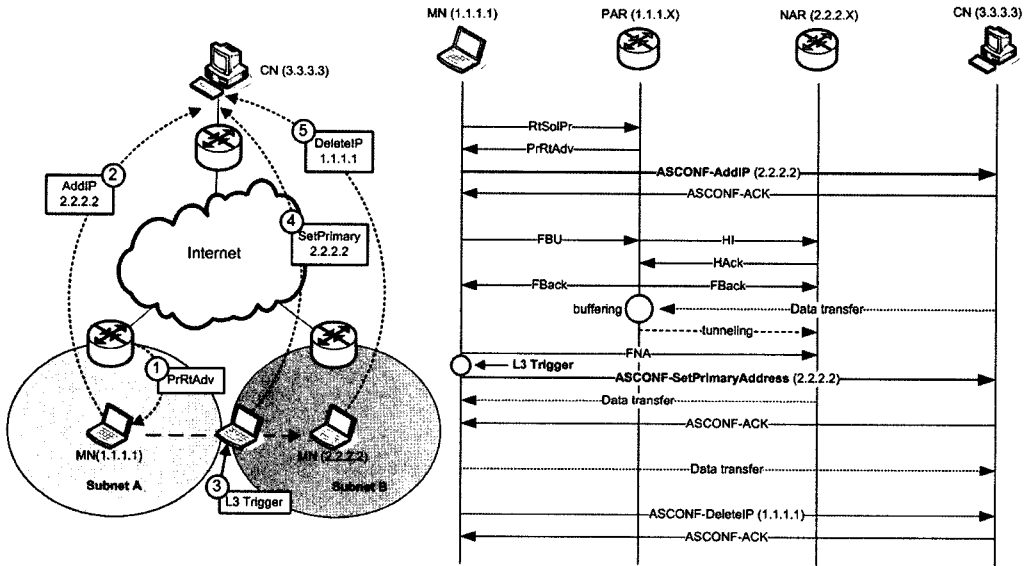


그림 4 FMIPv6와의 연동을 통한 mSCTP의 핸드오버 절차

트리거를 통해 MN가 NAR이 속한 망에 접속이 완료되었음을 확신 할 수 있게 되어 mSCTP 핸드오버 절차를 수행 할 수 있는 것이다.

이처럼 FMIPv6를 이용할 경우 핸드오버 지연 시간을 줄임과 동시에 안정적인 핸드오버를 보장하게 된다. 그리고 이동성 지원은 mSCTP가 담당하게 되고 핸드오버 결정 방식을 FMIPv6가 담당하게 된다.

FMIPv6와 연동을 통한 방안의 단점은 관련 연구에

서 언급했듯이 라우터나 MN에 FMIPv6 구현이 필요하고, 네트워크의 자원을 사용한다는 점이다. 즉, 터널링과 버퍼링(Buffering)을 해야 하기 때문에 추가적인 네트워크의 자원을 소모하게 된다.

### 4.3 듀얼 인터페이스를 이용한 방안

그림 5는 듀얼 인터페이스를 이용한 방법을 보여주고 있다. MN은 NAR의 RA를 수신하기 위해 RA 수신을 위한 전용 인터페이스를 추가로 가진다. 이를 통해 MN

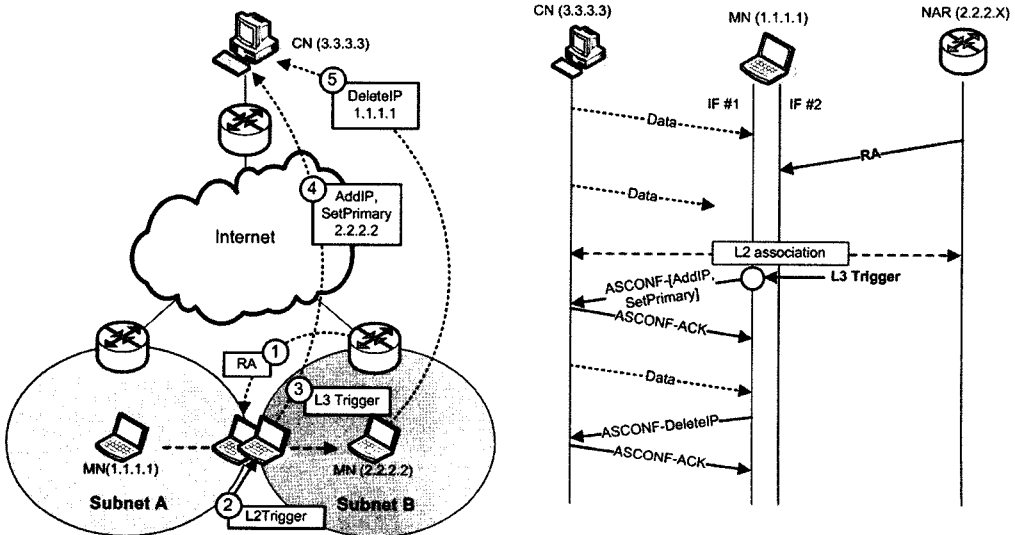


그림 5 듀얼 인터페이스를 이용한 mSCTP의 핸드오버 절차

은 주변 AP의 목록을 관리하고, MN이 PAR와 NAR의 중첩 영역(Overlay Area)으로 이동하게 되면 MN의 RA 수신 인터페이스는 NAR의 RA를 수신하게 된다. MN은 AP 목록에 PAR과 NAR 2개의 AP정보를 가지고 있게 되며 이중 링크 계층의 핸드오버 결정에 의해 대상 AP를 선택하게 된다.

가장 일반적인 링크 계층에서의 핸드오버 결정 방식은 신호세기(signal strength)가 가장 큰 AP를 선택하는 방법이다. 링크 계층에서의 핸드오버 결정 방식에 의해 선택된 AP의 정보는 L2 트리거를 통해 네트워크 계층으로 전달되고 네트워크 계층의 핸드오버 과정에 사용된다. 이후 L3 트리거가 mSCTP모듈에 전달되면 ASCON-ADDIP 메시지를 보내게 된다.

듀얼 인터페이스의 장점은 네트워크에서 추가적인 변화 없이 중단 호스트만 듀얼 인터페이스 방식으로 바뀌 주변 중단간 이동성을 지원하게 된다. 하지만 중단 호스트에 2개의 인터페이스가 필요하고 이를 처리하기 위한 모듈이 필요하다.

### 5. 실험 및 성능 분석

본 장에서는 제안된 RA수신 방식의 개선을 통한 mSCTP 핸드오버의 성능 향상을 실험을 통해 확인 하기 위해 mSCTP 핸드오버와, FMIPv6를 사용한 mSCTP 핸드오버의 성능을 비교 분석을 하고자 한다.

그림 6은 실험에 사용된 테스트 베드의 구성도 이다. 본 구성에서 CN은 PC 기반의 고정된 호스트 이고, MN으로는 노트북에 Lucent Orinoco USB client silver WLAN카드를 장착하여 사용 했다. 또한 MN과 CN은 OS로 리눅스 커널 2.6.12.2버전을 사용 하였는데, 여기에는 lksctp[14] 프로젝트로 공개된 SCTP 및 DAR 확장 모듈이 포함되어 있다. PAR와 NAR은 IPv6 라우팅을 지원하는 리눅스 커널 2.4.20 버전 기반의 라우터

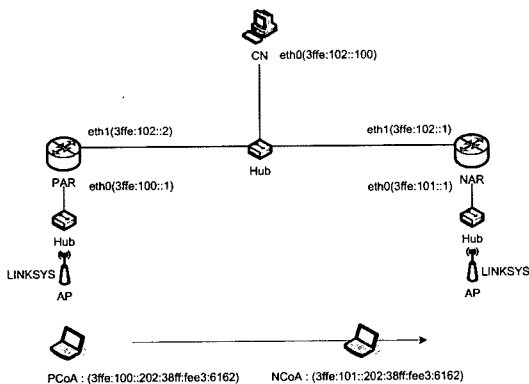


그림 6 테스트 베드 구성도

인데, Linksys wireless-G AP를 연결하여 사용 하였다. 그리고 각 라우터와 호스트에는 본 연구에서 직접 구현한 FMIPv6 모듈이 설치되어 있다.

현재 lksctp에서 제공하는 SCTP모듈은 ASCONF-ADDIP 메시지를 MN이 보내면, CN은 이에 대한 ASCONF-ACK를 ASCONF-ADDIP 메시지에 들어있는 소스 IP로 보내게 된다. 하지만 MN은 이미 핸드오버를 한 후이고, ASCONF-ACK 메시지를 받을 수 없다. 결국 시간초과로 인해 핸드오버 절차가 실패하게 되는데, 이러한 문제를 수정하기 위해 본 실험에서는 리눅스 커널에 속한 SCTP 모듈을 수정해서 만약 핸드오버 과정에 ASCONF-ADDIP 메시지를 CN이 받았을 경우 Address Parameter에 포함되어 있는 주소로 ASCONF-ACK 메시지를 보내도록 하였다.

상기와 같은 실험 환경 하에 본 논문에서는 다음 두 가지 사항을 측정하였다.

1. mSCTP의 핸드오버 지연시간 측정: MN은 주기적(본 실험에서는 1초 간격)으로 자신이 속한 서브넷이 바뀌었는지를 검사한다. 만약 자신의 주소가 바뀌었으면 ASCONF-ADDIP 메시지를 보내게 된다. 이후 CN으로부터 ASCONF-ACK 메시지를 받게 되면 ASCONF-SetPrimary 메시지를 보내게 되고 ASCONF-ACK를 받게 되면 핸드오버 절차가 끝나게 된다. MN은 PAR와 NAR사이를 20초 주기로 핸드오버 하게 된다.
2. FMIPv6를 이용한 mSCTP의 핸드오버 지연시간 측정: MN은 핸드오버 후 FMIPv6 모듈로부터 L3트리거를 받게 된다. 이를 통해 MN은 ASCONF-ADDIP 메시지를 바로 보내게 된다. ASCONF-ACK를 받게 되면 핸드오버 절차가 끝나게 되며 MN은 PAR와 NAR사이를 20초 주기로 핸드오버 하게 된다.

그림 7과 그림 8은 이 두 가지 결과를 나타내고 있다. 그래프에서 Y축은 핸드오버 지연시간을 X축은 핸드오버를 수행한 횟수를 나타낸다.

그림 7은 FMIPv6를 이용하지 않은 mSCTP의 핸드오버 결과를 나타내고 있다. 각각의 핸드오버 측정 값은 3개의 막대로 이루어져 있다. 첫 번째 막대( $\Delta_2$ )는 L2핸드오버가 수행된 후 ASCONF-ADDIP 메시지를 보내기 전까지의 구간이다. 두 번째 막대( $\Delta_2 + \Delta_3$ )의 경우 L2핸드오버가 수행된 후부터 ASCONF-ACK 메시지를 받기까지의 구간이다. 마지막 막대( $\Delta_1 + \Delta_2 + \Delta_3$ )는 전체 핸드오버 지연시간을 나타낸다.

이를 분석한 결과 대부분의 핸드오버 지연시간은  $\Delta_2$  구간이 차지하는 것을 알 수 있다. MN은 핸드오버 시작 후 NAR로부터 RA수신을 기다리게 된다. 실험에서

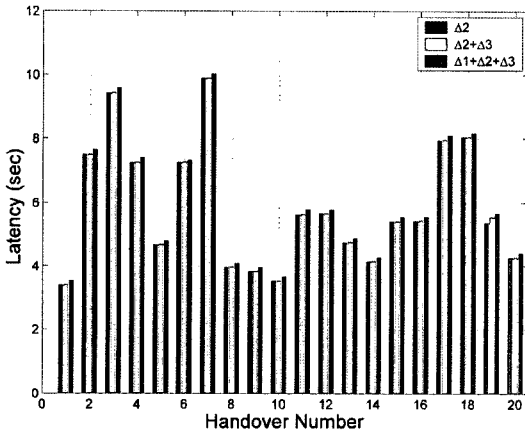


그림 7 mSCTP의 핸드오버 성능

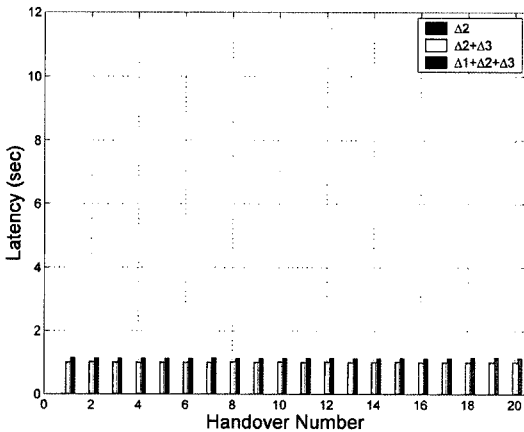


그림 8 FMIPv6+mSCTP의 핸드오버 성능

NAR은 RA를 최대 10초사이의 임의의 값을 선택, 주기적으로 보내게 된다. 그렇기 때문에  $\Delta 2$ 구간은 변화폭이 매우 큰 값을 가지게 된다.

그림 8은 FMIPv6를 이용한 mSCTP의 핸드오버 지연시간을 나타내고 있다. 첫 번째 막대( $\Delta 2$ )는 L2 핸드오버 이후부터 ASCONF-SetPrimary 메시지를 보내기 전까지의 구간이다. 두 번째 막대( $\Delta 2 + \Delta 3$ )는 L2 핸드오버가 수행된 후부터 ASCONF-ACK를 받기까지의 구간이다. 마지막 막대( $\Delta 1 + \Delta 2 + \Delta 3$ )는 전체 핸드오버 지연시간을 나타낸다.

여기서  $\Delta 2$ 구간의 값은 평균 1.2ms 정도로 매우 작음을 알 수 있다. MN은 핸드오버 하기 전에 RA를 미리 받기 때문에 대부분의 핸드오버 지연시간을 차지하는  $\Delta 2$ 구간이 매우 작은 값을 가지게 된다. 하지만 이 실험에서 ASCONF-ACK 메시지를 받는데 까지 시간이 약 1초가 걸리는 것을 볼 수 있는데, 이는 IPv6에서 자동 주소할당 후 첫 번째 패킷은 DAD(Duplicate Address

Detection), NUD(Neighbor Unreachability Detection) 과정을 거치기 때문이다. 본 실험 환경에서는 IP 계층에 속하는 이 부분을 최적화 하지 않았지만 핸드오버 성능 향상을 위해서는 최적화 과정이 필요할 것으로 보인다. 실험 결과 FMIPv6를 통해 미리 RA를 받음으로써 전체 핸드오버 지연시간이 매우 작아졌고, 또한, 핸드오버 지연시간의 변화 량이 매우 작은 값을 가지게 되어 전체적인 핸드오버의 안정성이 크게 향상된 것을 볼 수 있다.

### 6. 결론

본 논문에서는 mSCTP의 성능을 향상시키기 위해 3 가지 방법을 제안하였다. 첫 번째는 RA 캐시를 이용한 방법, 두 번째는 FMIPv6와의 연동을 통한 방법, 세 번째는 듀얼 인터페이스를 이용한 방법이다. 세가지 방법 모두 RA를 줄이는데 목적이 있고 L3 트리거를 정의 하였다. 하지만, 첫 번째 방법은 RA 캐시를 구현해야 하며, 이를 처리하는 모듈이 네트워크에 포함되어야 한다. 두 번째 방법은 빠르고 안정적인 핸드오버를 제공하지만 역시 FMIPv6를 처리해야 하기 때문에 네트워크에서 이를 제공해야 한다. 세 번째 방법인 경우 MN만 수정하면 가능하기 때문에 간단하게 종단간 이동성을 제공하게 된다. 하지만 종단 호스트에 2개의 인터페이스가 필요하고 이를 처리할 수 있는 모듈이 MN에 필요하다.

실험 결과에서 우리는 mSCTP와 FMIPv6를 연동한 mSCTP의 핸드오버 성능을 비교 분석하였다. 실험을 통해 핸드오버의 대부분의 시간이  $\Delta 2$  구간이라는 것을 볼 수 있었다. 또한, RA를 핸드오버 하기 전에 미리 수신함으로써  $\Delta 2$ 구간을 줄일 수 있었고, 이를 통해 핸드오버 성능을 향상시킬 수 있었다.

### 참고 문헌

- [1] Rajeew Koodli, "Fast Handovers for Mobile IPv6," IETF Internet-Draft <http://www.ietf.org/internet-drafts/draft-ietf-mipshop-fast-mip6-03.txt>, July 2004.
- [2] Castelluccia, K. El Malki, L. Bellier "Hierarchical Mobile IPv6 Mobility Management," IETF RFC 4140, August 2005, Work in progress.
- [3] Moon J. Chang, Mee J. Lee, "A Transport Layer Mobility Support Mechanism," ICONS 2004 LNCS 3090, pp. 287-296.
- [4] C.E.Perkins, "Mobility Support in IPv6," in Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MOBICOM'96), November 1996.
- [5] D.Johnson, C.Perkins, J.Arkko, "Mobility Support in IPv6," Draft-ietf-mobileip-ipv6-24.txt, IETF Mobile IP Working Group, December 2003.



- [6] R. Stewart et al., "Stream Control Transport Protocol," IETF RFC 2960, Oct. 2000.
- [7] Ong, L. et al., "An introduction to the Stream Control Transmission Protocol (SCTP)," RFC 3286, May 2002.
- [8] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, P. Conrad, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration," Internet-Draft, February 2005, <http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-addip-sctp-11.txt>, Work in progress.
- [9] W.Xing, H. Karl, and A. Wolisz, "M-SCTP: Design and Prototypical Implementation of an End-to-End Mobility Concept," Proc. 5th Int'l. Wksp., Berlin, Germany, Oct. 2002.
- [10] Seok J. Koh, Qiaobing Xie, "Mobile SCTP (mSCTP) for Internet Mobility," IETF Internet-Draft, March. 2005, <http://www.ietf.org/internet-drafts/draft-sjkoh-msctp-00.txt>, Work in progress.
- [11] S. Kashihara, T. Nishiyama, K. Iida, H. Koga, Y. Kadobayashi, and S. Yamaguchi, "Path Selection Using Active Measurement in Multi-Homed Wireless Networks," Proc. The 2004 International Symposium on Applications and the Internet (SAINT2004)," pp.273-276, 2004.
- [12] Li MA, Fei Yu, Victor C.M.Leung, "A New Method to Support UMTS/WLAN Vertical Handover Using SCTP," IEEE Wireless Communication, August 2004.
- [13] Li Ma, Fei Yu and Victor C.M.Leung, "SMART-FRX: A Novel Error-Recovery Scheme to Improve Performance of Mobile SCTP during WLAN to Cellular Forced Vertical Handover," IEEE Communications Society, 2005.
- [14] The Linux Kernel SCTP project, <http://lksctp.sourceforge.net/>.

관심분야는 QoS 제공, 성능분석, 이동성지원



민 성 기

1988년 고려대학교 컴퓨터학과 학사  
1989년 런던대학교 석사. 1994년 런던대학교 박사. 1994년~2000년 LG 정보통신 중앙연구소 책임연구원. 2001년~현재 고려대학교 컴퓨터학과 부교수. 관심분야는 분산 시스템, 고속통신, 무선 통신



최 순 원

2001년 단국대학교 전자계산학과 학사  
2006년 고려대학교 컴퓨터학과 석사  
2001년~2003년 ESGroup 연구원. 관심 분야는 이동성 지원, 핸드오버



김 광 렬

1994년 경북대학교 전자공학과 학사. 1996년 경북대학교 전자공학과 석사 1996년~2000년 LG 정보통신 중앙연구소 주임연구원. 2000년~2005년 LG 전자 시스템 연구소 책임 연구원. 2005년~현재 LG-Nortel R&D Center 책임 연구원

2002년~현재 고려대학교 컴퓨터학과 박사 과정 재학 중