

제2계층 트리거를 이용한 FMIPv6 구현 및 분석

오 승 훈[†] · 이 성 식^{**} · 김 영 한^{***}

요 약

본 논문에서는 신속한 이동성 관리기법인 FMIPv6(fast mobile IPv6)를 리눅스 기반으로 구현하여 기존 시스템과의 성능을 비교 실험하였다. FMIPv6 기법은 제 2계층의 동작에 의존하고 있기 때문에 이에 맞게 네트워크 드라이브를 수정하여 단말의 이동에 따라 제2계층에서의 트리거를 발생하도록 하였고 액세스 라우터와 이동 노드로 동작을 구분하여 설계하여 구현하였다. 구현된 시스템으로 기존 MIPv6(mobile IPv6)가 적용됐을 때와 비교하여 핸드오버기간동안의 패킷손실과 지연특성을 실험하였으며 FMIPv6에서의 개선된 성능을 고찰 하였다.

키워드 : 모바일 아이피, 핸드오버, 2계층 트리거

Implementation and Analysis of the FMIPv6 (Fast Handover for Mobile IPv6) Using Layer 2 Triggers

Seung Hun Oh[†] · Sung Sik Lee^{**} · Young Han Kim^{***}

ABSTRACT

In this paper, we implement the fast handovers for mobile IPv6 (FMIPv6) on Linux system. Due to its dependency on operations in layer-2 (L2), we have added some functions into the network driver to generate triggers as the mobile node moves. We design and implement the FMIPv6 functions divided into two parts as an access router and a mobile node. We compare the packet loss and delay of the FMIPv6 implementation during the handover period with those of the MIPv6 and investigate the performance improvement.

Key Words : Mobile IP, Handover, Layer-2 Trigger.

1. 서 론

무선 멀티미디어 서비스는 무선 액세스 구간에서의 신속한 핸드오버 기술을 필요로 하고 있다[1]. 특히, 실시간성 서비스에는 핸드오버 기간 동안의 지연과 패킷 손실 요구를 서비스에서 요구하는 수준을 필수적으로 만족시켜야 한다. 현재 인터넷에서의 이동성 기술인 MIPv6 기술은 단말이 이동하더라도 같은 IP 주소를 유지할 수 있어 응용 세션을 계속 유지해 줄 수 있다. 그러나 핸드오버시 걸리는 지연시간이 긴 특성을 가지고 있어 실시간 서비스들에게는 부적합하다. MIPv6의 지연시간은 단말이 이동했음을 3계층 수준에서 감지하는데 소요되는 시간과 생성된 CoA(care-of address)의 유효성을 검증하고 HA(home agent)에 등록하는데 걸리는 시간으로 구성된다[1]. 핸드오버 지연시간은 곧 데이터 손실을 유발하기 때문에 손실과 지연시간에 민감한 응용프로그램들에게는 MIPv6

를 적용할 수 없다.

패킷 손실 없이 신속히 핸드오버를 수행할 수 있는 기법들을 개발하기 위한 다양한 연구들이 진행되고 있다[2-8]. 이들 연구들은 단말의 이동을 신속히 감지해 내는 기술과 패킷 손실을 최소화 시키는 연구로 나뉘 볼 수 있다. 그 중 GPS(global positioning system)장치를 이용해서 단말의 위치를 파악하는 기술[2]은 그 정확성이 우수하여 단말 이동 감지가 빨라 핸드오버를 신속히 시동할 수 있다. 그러나 모든 장치에 장착하기에는 어려운 경제적인 문제가 존재한다. GPS 없이 단말의 위치를 파악하는 연구에서는 3개 이상의 기지국들의 전파세기를 이용하여 위치를 계산하는 방법[6]이 있다. 그러나 무선 환경에는 다양한 통신 장애요소 때문에 단말이 고정되어 있어도 전파신호의 세기가 변하여 그 결과 계산된 위치의 정확도가 떨어질 가능성이 높아 핸드오버 결정에 상당한 오류가 발생할 수 있다.

핸드오버 기간 동안 발생하는 패킷 손실을 제거하기 위한 방법으로써 원래 목적지로 전달될 패킷을 핸드오버가 발생할 후보 지역으로도 복사하여 동시에 멀티캐스팅 방식으로 전송하는 방식이 제안되었다[3, 4]. 그러나 트래픽이 전달될 후부

※ 본 논문은 정보통신부지원 BcN ITRC 사업의 지원에 의해 이루어졌음.

† 정 회 원 : 숭실대학교 정보통신공학과 박사과정

** 준 회 원 : 숭실대학교 정보통신공학과 석사

*** 총신회원 : 숭실대학교 정보통신전자공학부 부교수 (교신저자)
논문접수 : 2005년 2월 2일, 심사완료 : 2005년 7월 18일

라우터들을 멀티캐스트 그룹으로 관리하는 제어방법이 복잡하고 불필요한 패킷 전달에 의한 자원 낭비가 심한 문제점을 갖고 있다.

현재 IEEE802.21 표준 그룹을 중심으로 연구되고 있는 2계층 트리거 기술은 다양한 미디어에 독립적으로 핸드오버를 예측할 수 있는 정보를 전달할 수 있기 때문에 단말의 이동을 신속히 감지할 수 있다. 또한 대부분의 무선 기술에 적용할 수 있도록 표준화되고 있기 때문에 이종망간 핸드오버가 더욱 수월해 지는 장점을 가진다[5]. 이런 장점 때문에 트리거를 이용한 다양한 핸드오버 기술들이 연구 중이다[4, 6-8]. Kempf J.는 FMIPv6을 구현하여 그 성능을 분석하였으나, 분석 환경이 실제 무선 환경이 아닌 무선링크를 에뮬레이션 장비로 통해 구성하였다[7]. 그리고 AR(access router)와 AP(access point)들의 매핑 정보를 사전에 저장하여 핸드오버에 이용한 연구[6]는 FMIPv6의 reactive모드의 동작과 개념 상 큰 차이가 없고 여전히 핸드오버 지연시간과 패킷이 손실을 완전히 해결하지 못했다.

본 논문은 FMIPv6 프로토콜을 구현하고 그 동작을 최적화할 수 있는 제 2계층 트리거를 무선랜 환경에서 동시에 구현하였다. 2계층 트리거는 무선랜 링크 계층을 계속적으로 감시하는 기능을 Prism III[9] 기반 무선랜 카드 디바이스에 드라이버에 구현하여 2계층 이벤트를 발생시켰다. 그리고 이를 실제 무선 환경에서 실험하여 기존 MIPv6에 비해 성능이 개선됨을 손실률과 지연시간측면에서 증명하였다. 그리고 MIPv6의 동작에 보안관련 절차는 중요한 요소이나 본 논문에서는 범위 밖으로 FMIPv6의 내용인 핸드오버 처리에서 보안절차를 제외한 부분에 대해서만 고려한다.

서론에 이어 제 II장에서는 FMIPv6 기술을 살펴보고 III장에서 제 2계층 트리거와 FMIPv6의 설계 및 구현 내용을 기술한다. 이어서 IV장에서 구현한 내용을 이용한 성능 실험 결과를 살펴보고 V장에서 본 논문의 결론을 맺는다.

2. FMIPv6

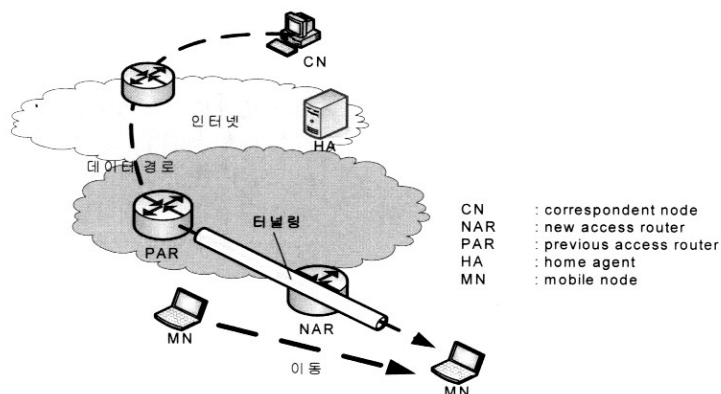
FMIPv6의 기본 개념은 (그림 1)과 같이 두 AR사이의 터널을 사용하여 핸드오버 동안에도 손실 없이 연속적으로 서

비스를 제공하는 데 있다. FMIPv6가 정상적으로 운영되기 위해서는 FMIPv6를 지원하는 이동노드가 NAR(new AR) 방향으로 이동할 때 이동노드와 PAR(Previous AR)의 연결을 여전히 유지하는 동안에도 NAR와 관련된 2계층 정보를 미리 얻을 수 있다고 가정한다. 또한 AR에는 AR에 연결되어 있는 AP들의 2계층 주소와 자신의 주소를 같이 묶어서 저장하고 이를 이웃 AR들과 교환한다. 그 결과 모든 AR들은 이웃 AR들의 IP 주소뿐만 아니라 그것에 연결되어 있는 AP의 2계층 주소도 알고 있다고 가정할 수 있다[8].

FMIPv6는 (그림 1)과 같이 MIPv6가 시작하기 전 2계층 트리거를 이용하여 PCoA와 NCoA 사이에 터널을 미리 생성케 한다. 이를 위해 FMIPv6는 터널 끝단을 구성하는 NCoA(New CoA)를 미리 생성하고 그 유용성을 검사한 후 이를 이용하여 터널을 생성한다. 그리고 마지막으로 단말의 존재를 NAR에게 신속하게 통보하여 패킷 손실을 최소화한다.

PAR의 영역에서 NCoA를 생성하는 기능은 제 2계층의 트리거로부터 얻은 2계층 정보를 이용하여 NAR의 Prefix 정보를 얻을 수 있기 때문에 가능하다. 이를 위해서는 NAR의 IP 주소와 2계층 주소 그리고 그 AR에 연결되어 있는 모든 AP들의 2계층 정보(주소)들이 같이 관리되어 있어야 한다. 이 기능은 CARD (candidate access router discovery) 프로토콜에 의해 이루어진다[10]. 미리 생성된 NCoA는 터널의 끝단의 주소로 쓰일 뿐 아니라, 추후 MIPv6의 HA에 등록할 때도 바로 쓰인다.

PCoA(previous CoA)와 NCoA 사이의 터널 생성은 무선단말이 전송하는 FBU (fast binding update) 메시지에 의해서 시작된다. 그런데 FBU를 수신한 PAR은 미리 생성된 NCoA의 유용성을 확인한 후에야 터널생성 성공을 알리는 FBack (fast binding acknowledgement)를 단말에 전달해 줄 수 있다. NCoA의 유용성을 검사는 일반 DAD (duplicate address detection)[11]과정보다는 NAR가 가지고 있는 하부망 노드 리스트를 이용하여 수행된다. 그 결과 DAD 과정에 소요되는 시간 단축하게 된다. 끝으로 핸드오버에 의한 서비스 단절 시간을 최소화하기 위해서는 단말이 NAR과 2계층 링크를 연결하자마자 3계층 수준의 자신의 존재를 신속하게 알려야 한다. 이를 위해 FNA(fast neighbor advertisement) 메시지를 새롭



(그림 1) FMIPv6의 기본 동작

게 정의했으며, 이 메시지를 NAR가 수신하는 순간 버퍼링되어 있던 트래픽을 신속하게 전달한다.

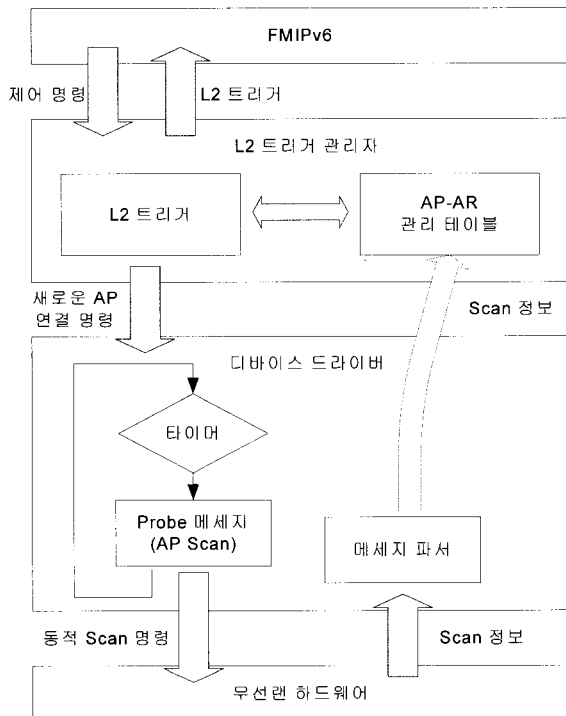
3. 제 2계층 트리거를 이용한 FMIPv6 구현

3.1 제 2계층 트리거 설계 및 구현

FMIPv6 규격에는 단순 트리거만 언급하고 있기 때문에 본 논문에서는 [5]와 [12]에서 정의하고 있는 핸드오버관련 트리거를 참조하여 실제 제 2계층 트리거들을 구현하였다. 구현은 리눅스 커널 2.4.22 시스템의 Prism II 칩셋을 기반으로 하는 Orinoco 무선랜 카드 디바이스 드라이버를 기반으로 구현 하였다.

(그림 2)와 같이 효율적으로 트리거를 발생하기 위해서 3 계층과 2계층 사이에 L2 트리거 관리자를 두어 AP정보 수집을 위해 하드웨어를 제어하고, 위로는 시스템 호출을 이용해서 FMIPv6에게 트리거를 전달할 수 있도록 구현하였다.

2계층 트리거의 핵심은 주변 AP들을 인지하고 그들의 전파 신호세기를 측정할 수 있어야 한다. AP 스캔 방법에는 명시적인 요청에 의한 것과 AP들의 Beacon 신호를 측정하는 방법이 있다. AP에 스캔 요청하고 응답들을 수신하는 데 소요되는 시간은 수백 msec가 걸린다[12]. 이는 신속히 FMIPv6에게 트리거를 발생시키는 데에 적합하지 않다. 이에 반해 AP의 Beacon 신호는 100ms 주기로 발생되나 단말이 다른 AP의 Beacon을 수신할 수 없다. 이에 따라 본 구현에는 스캔을 통해서 주변 AP들의 정보를 주기적으로 갱신하도록 하고 두 번째 방법을 통해서 현 AP의 신호에 따라서 트리거 발생을 결정하도록 하였다.



(그림 2) 제 2계층 트리거 구현 블록도

AP 스캔 요청 메시지를 주기적으로 발생시키기 위해서는 무선랜 메인칩을 직접 제어할 수 있는 시스템 명령어를 이용하여 디바이스 드라이버를 수정해야 한다. 새로운 H/W명령과 디바이스 드라이버에 타이머를 구현하여 주기적인 AP 스캔이 가능하도록 하였다.

AP 스캔의 결과로 얻어진 주변 AP의 정보는 핸드오버할 새 AR를 찾기 위한 힌트정보이므로 L2 트리거 관리자의 테이블에 관리된다. 단말은 이렇게 수집된 AP의 정보를 힌트로 현 AR으로부터 얻어온 NAR 정보와 측정된 전파신호 세기도 함께 테이블에 저장된다.

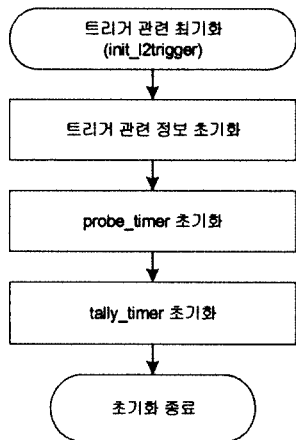
FMIPv6를 시동할 트리거는 현 AP의 Beacon 신호가 특정 기준값(임계값)보다 낮고 테이블에 미리 등록된 주변 AP들 중 전파신호세기가 가장 우수한 것보다 낮을 때 발생된다. 이때 전파신호가 가장 우수한 AP의 정보도 파라미터로 함께 전달해 준다.

FMIPv6 프로토콜이 완료되거나 현 AP 신호 세기가 최소에 이르게 되면 제 2계층 트리거 관리자는 AP-AR 정보 테이블에서 가장 강한 전파신호를 가진 AP에 새로운 연결을 맺도록 트리거를 발생 시키게 된다.

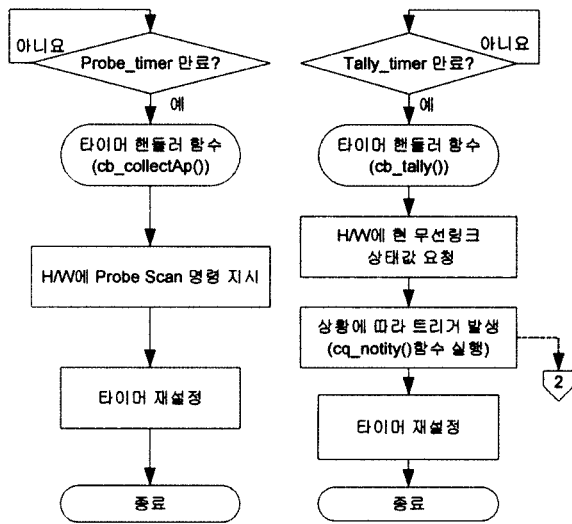
<표 1> 제 2계층 트리거 관련 RID[9]

RID 정의	RID값	설명
HERMES_RID_CNFJOINREQUEST	0xFC2E	새로운 AP에 연결 명령
HERMES_RID_COMMSQUALITY	0xFD43	주기적인 Beacon 제어 정보 읽기 명령
HERMES_INQ_TALLIES	0xF100	주기적인 Beacon 제어 정보 읽기 인터럽트 이벤트
HERMES_RID_CNFSCANREQUEST	0xFCE1	동적 AP Scan 명령
HERMES_INQ_SCAN	0xF101	동적 AP Scan 명령 인터럽트 이벤트

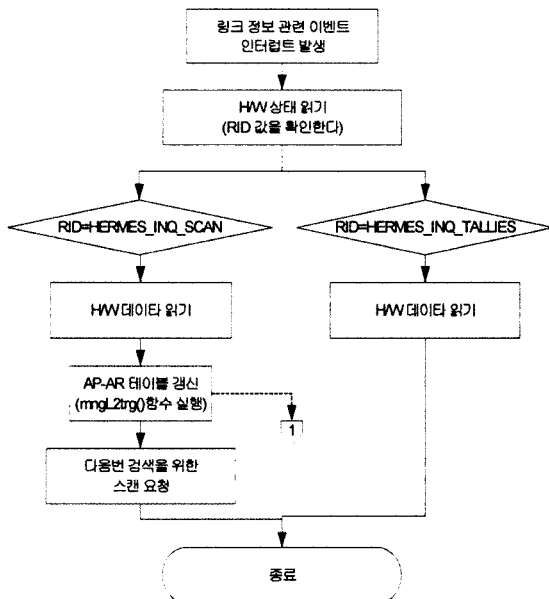
<표 1>은 제 2계층 트리거를 드라이버단에 구현하기 위해 참조하고 있는 하드웨어 RID(Resource ID)값이다. HERMES_RID_CNFJOINREQUEST은 FMIPv6가 마무리 되거나 현 AP의 극한의 신호 세기를 보일 때 강제적으로 AP-AR테이블에서 가장 우수한 AP에게 연결 할 때 하드웨어에 내리는 명령이다. HERMES_RID_COMMSQUALITY는 주기적으로 수신되는 Beacon[9] 제어 신호로부터 얻은 하드웨어 정보를 드라이버에 알려 줄 것을 요구하는 명령이고 이때 드라이버는 HERMES_INQ_TALLIES로 인터럽트를 발생하는 방식으로 AP의 정보를 읽을 수 있다. HERMES_RID_CNFSCANREQUEST은 하드웨어에게 동적인 AP정보 수집을 지시하는 명령이고 HERMES_INQ_SCAN으로 인터럽트 방식에 의해 AP들의 정보를 수집 한다[9]. 이렇게 RID를 이용하여 트리거 발생시키는 방법은 향후 IEEE802.21에서 정해질 표준을 구현할 수 있는 하나의 방법으로써 의미가 있다.



(그림 3) 트리거 초기화



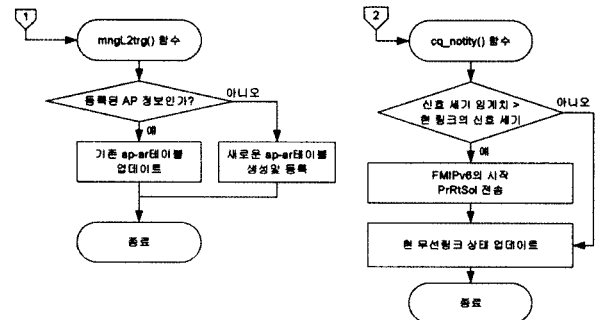
(그림 4) 타이머 핸들러 함수들



(그림 5) 제 2계층 트리거 인터럽트 핸들러

제 2계층 트리거의 초기화는 (그림 3)과 같이 Probe Scan(Active Scanning)[9] 타이머(probe_timer)와 통신 상태 값을 읽기 위한 Tally[9] 타이머(tally_timer)를 설정하는 과정이다. 초기화 된 타이머는 각각 설정된 타이머 시간이 지나면 한번 씩 (그림 4)의 타이머 핸들러를 호출 하게 된다. 이때 Probe타이머는 Probe Scan을 Tally타이머는 현 링크 값 읽기 명령을 H/W에게 지시한다.

타이머 핸들러에서 지시한 H/W명령의 결과는 인터럽트 핸들러로 반환 한다. (그림 5)는 인터럽트 핸들러로 다양한 인터럽트를 처리하고 있지만 본 연구에서 새로 추가한 두 개의 인터럽트 처리루틴만을 보여 주고 있다. <표 1>에 명시한 HERMES_INQ_SCAN 이벤트이면 주변 AP의 정보를 관리하는 mngL2trg()함수로 주변 AP의 정보를 인자로 전달하고, HERMES_INQ_TALLIES의 이벤트이면 현 링크의 값을 H/W에게 새로 갱신 할 것을 명령 한다.



(그림 6) 제 2계층 트리거 발생 및 정보 갱신

(그림 6)은 트리거 관리자의 테이블의 정보를 갱신하는 부분과 트리거를 발생하는 부분이다. 주변 AP의 정보를 전달 받은 mngL2trg() 함수는 테이블에 저장된 기존 AP의 상태값을 갱신하고 새 AP의 정보는 새로 생성하여 값을 기록한다. cq_notify() 함수는 Beacon으로 측정된 현 AP의 전파신호가 특정 임계치보다 작으면 제 2계층 트리거를 발생시켜 FMIPv6를 시동 시킨다. <표 2>에 FMIPv6에 구현된 주요 함수들을 정리하였다.

<표 2> FMIPv6에 구현된 주요 함수들

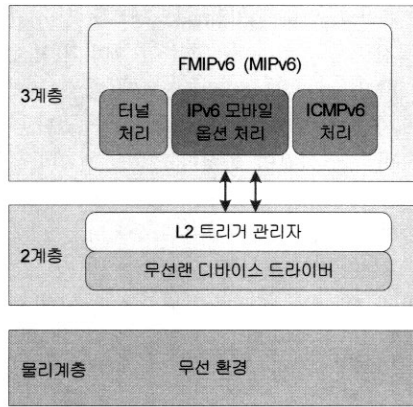
함수 명	기능
cb_collectAp()	무선랜카드에서 AP 스캔을 요청메시지를 전송하게 함
cb_tally()	현 AP의 Beacon신호를 수신한 H/W정보를 읽어옴
cq_notify()	현 AP전파 신호를 검사하여 트리거 생성시킴
mngL2trg()	AP 스캔 결과에 따라 AP-AR 테이블 갱신함

3.2 FMIPv6 프로토콜 구현

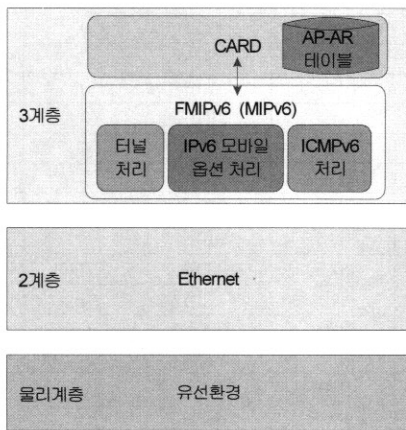
3.2.1 전체 동작 설계 및 구현

FMIPv6 프로토콜 구현은 크게 MN(mobile node) 부분과 AR 부분 두 동작으로 나눌 수 있다. MN 부분에는 제 2계층 트리거를 생성하는 트리거 관리자를 디바이스 드라이버에 구현하고 FMIPv6의 새 ICMPv6메시지와 추가된 모바일 헤더

를 처리하는 기능, PCoA와 NCoA의 터널을 위한 PAR의 터널을 생성(PCoA와 NCoA 바인딩)기능, 마지막으로 패킷의 터널 송수신을 처리할 터널 처리하는 기능들이 구현 되었다. AR 부분은 MN 부분과 유사하나 2계층 트리거 부분이 없으며 주변 AR들의 AP-AR 정보를 관리하기 위한 CARD 프로토콜을 구현해야 한다. 본 구현에서는 이 정보를 스크립트로 설정하여 CARD프로토콜의 기능을 대신하였다(그림 8). AP-AR 테이블은 AR의 서브넷을 구성하는 AP들의 MAC 주소와 무선랜 채널 정보, ESSID 정도들을 저장하고 있다.



(그림 7) MN측 구현 모듈

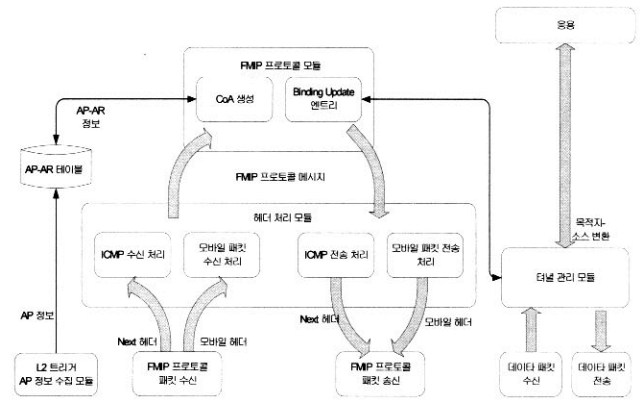


(그림 8) AR측 구현 모듈

3.2.2 MN 측 구현

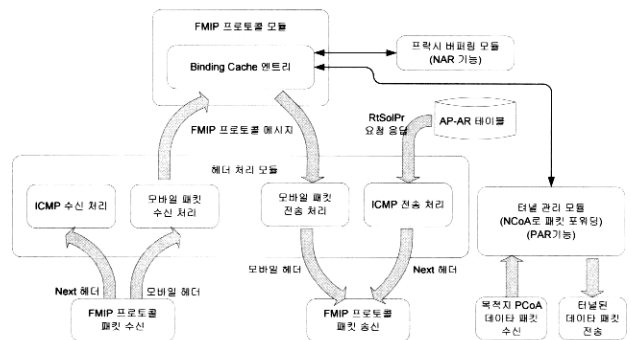
MN의 FMIPv6 구현은 4가지 모듈로 구성되는데, FMIPv6 프로토콜 모듈과 헤더 처리 모듈, 터널 처리 모듈, 제 2계층 트리거 모듈들이다. FMIPv6 프로토콜 모듈에서 만들어진 FMIP 패킷은 헤더 처리 모듈에 보내져 ICMP나 모바일 헤더로 구성되어 전송되고 수신된 ICMP나 모바일 헤더도 헤더 처리 모듈에 의해 FMIPv6 모듈로 입력된다.

MN의 전체적인 동작을 살펴보면 (그림 9)와 같다. 먼저 제 2계층 트리거에 의해 무선랜 환경의 이벤트를 받게 되면 FMIPv6모듈은 새로운 AP에 해당하는 AR의 정보를 얻기 위해 RtSolPr(router solicitation for proxy)에 AP의 BSSID(basic service set ID)를 담아 헤더 처리 모듈을 통해 ICMPv6



(그림 9) MN의 구현 모듈 세부 구조도

메시지를 전송 한다. 이의 응답으로 PrRtAdv(proxy router advertisement)메시지를 받게 되면 헤더 처리 모듈의 ICMP 수신 처리 모듈을 거쳐 FMIPv6이 CoA생성 모듈에 의해 NCoA를 생성 한다. 다시 NCoA는 FMIPv6 프로토콜 모듈에 의해 생성된 FBU메시지에 포함되어 모바일 헤더 처리 모듈에서 모바일 패킷을 전송하고 또한 BU 엔트리에 추가된다. 그런 후 FBack메시지를 수신하게 되면 NCoA와 PCoA의 터널이 정상적으로 생성됨을 인지하게 된다. 터널 생성이 마무리 되고 나면 MN는 바로 NAR로 L3 핸드오버를 위해 NAR의 AP에 연결을 맺고 FNA를 보내 NAR에게 자신이 이동했음을 알림으로써 MN측의 FMIPv6 제어동작은 완료된다.



(그림 10) AR의 구현 모듈 구조도

3.2.3 AR 측 구현

FMIPv6 프로토콜을 지원하기 위해 AR에는 FMIPv6 프로토콜[8] 처리 모듈, 터널 생성 및 관리 모듈, 프락시 패킷 버퍼 모듈을 구현 하였다. FMIPv6의 AR 기능은 MN가 PAR에 NAR 정보를 요청하는 RtSolPr메시지에 이에 대한 응답으로 요청한 NAR의 주소, Prefix, 링크로컬 주소 등을 담은 PrRtAdv를 전송한다. IPv6의 RA메시지와 달리 FMIPv6의 메시지는 RtSolPr의 요청이 있어야만 전송된다.

AR의 내부 구현 모듈을 보면 (그림 10)과 같다. MN와 같이 헤더 처리 모듈에서는 FMIPv6의 ICMPv6 헤더 처리와 모바일 메시지의 헤더 처리 부분을 구현 하였다. FBU 메시지를 받은 AR는 그 안의 NCoA정보와 PCoA정보를 바탕으로 BU과정을 수행 하게 되는데 이를 FMIPv6 프로토콜 모듈

의 Binding Cache 엔트리에서 관리 되도록 구현하였다.

AR의 AP-AR 테이블은 MN의 것과 유사하다. 이는 서로 오는 RtSolPr메시지에 대한 응답으로 PrRtAdv에 해당 AP에 대한 AR의 정보를 교환하고 관리하는데 사용된다.

터널 관리 모듈은 FBU 과정이 성공적으로 수행 되면 PAR로 동작하는 AR에서 PCoA와 NCoA를 바탕으로 터널을 생성 및 관리하도록 구현 하였다.

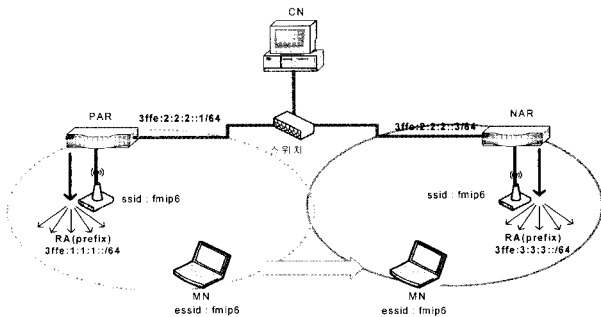
4. 실험 및 성능 분석

본 논문에서 구현한 FMIPv6의 성능을 분석하기 위해서 실제 환경에서 실험을 통해 핸드오버 지연시간과 손실률 측면에서 MIPv6와 비교하였다. 핸드오버 지연시간은 단말이 이전 네트워크에서 마지막으로 데이터 패킷을 수신한 시점부터 새 네트워크에서 처음으로 패킷을 수신한 시점까지의 시간을 의미한다. 핸드오버 손실수는 핸드오버 동안 수신하지 못한 패킷의 양을 의미하고 손실률은 일정 데이터 패킷량에 대한 핸드오버 손실의 비율을 의미한다.

4.1 실험 환경

MIPv6와 FMIPv6를 공평하게 비교하기 위해서 (그림 11) 과 같이 동일한 실험환경을 적용하였다. PAR에 HA 기능을 하도록 하였고 CN는 FMIPv6를 위해 수정된 사항이 없다. 무선랜의 2계층 핸드오버가 원활하게 진행되도록 PAR과 NAR쪽의 AP의 SSID를 fmp6로 통일하였고 각 AR이 해당 prefix정보를 RA 메시지를 통해서 광고하도록 설정하였다.

CN에서는 MN에게 데이터를 전송하게 하고 데이터를 수신하고 있던 MN을 NAR 쪽으로 MIPv6 또는 FMIPv6 각각 이용하여 핸드오버가 발생하도록 하였다. 이때 CN에서 발생하는 트래픽은 Mgen[13] 프로그램을 이용하였고 핸드오버 자체의 성능을 분석하기 위해 CBR (constant bit rate)의 트래픽을 발생하도록 하였다. 이 때 트랜스포트 프로토콜은 UDP를 사용하였다. CBR 트래픽은 멀티미디어 스트림을 흉내 내기 위해 50Kbps, 100Kbps, 300Kbps, 500Kbps, 700Kbps 등의 다양한 전송속도로 발생시켰으며, 각 전송속도에 있어 250byte, 500byte, 1000byte, 1400byte로 패킷 크기를 변경하였다. 성능 분석은 MN이 Mgen 트래픽을 수신한 시간과 수신 양 등을 기록하도록 하여 그 값을 토대로 분석하였다



(그림 11) 실험 환경

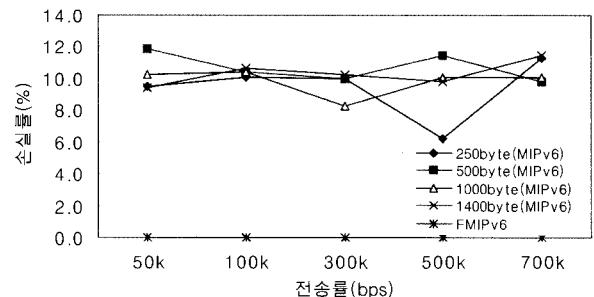
4.2 성능 비교

<표 3>은 동일한 실험 환경 하에서 FMIPv6와 MIPv6 각각을 실험한 결과를 나타낸 표이다. 표의 패킷 손실수는 핸드 오버동안 단말이 수신하지 못한 패킷의 수를 기록한 결과이다. 특히 이 결과로 부터 FMIPv6는 평균 256ms의 지연이 있었지만 MIPv6에서는 평균 331ms의 지연이 발생함을 확인할 수 있다.

(그림 12)는 <표 3>을 토대로 핸드오버 시 발생한 패킷 손실률을 나타낸 것이다. 그림 상단의 라인들은 MIPv6이고 하단의 것은 FMIPv6이다. MIPv6는 대체로 8%에서 10%사이의 손실을 보이고 있다. 331msec 핸드오버 시간동안 이전 서브넷에서 전달된 패킷들 중 손실된 패킷의 비율을 의미한다. 하지만 FMIPv6는 모든 전송속도와 패킷 크기에 관계없이 핸드오버 손실이 발생하지 않음을 알 수 있다. 약 256msec 정도 핸드오버 지연시간이 발생하는데도 손실이 없는 이유는 NAR에 버퍼를 설치하여 단말이 FMIPv6 핸드오버 절차를 수행하는 동안 패킷을 저장하였기 때문이다.

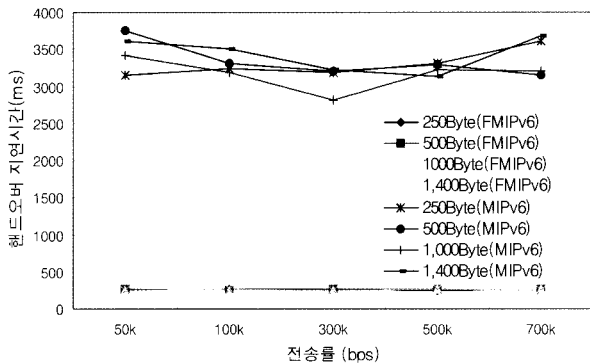
<표 3> MIPv6와 FMIPv6의 패킷 전송 실험 결과

bps	packet size (byte)	MIPv6		FMIPv6	
		패킷손실수 (packet)	지연시간 (ms)	패킷손실수 (packet)	지연시간 (ms)
50k	250	76	3150.5	0	251.5
	500	47.5	3749	0	263
	1,000	20.5	3410.5	0	245.5
	1,400	13.5	3603	0	249
100k	250	161	3230.5	0	258.5
	500	83	3310.5	0	252
	1,000	41.5	3190	0	249
	1,400	30.5	3511	0	260.5
300k	250	478	3180	0	257.5
	500	240	3206	0	263.5
	1,000	99.5	2810.5	0	246.5
	1,400	87.5	3228	0	253.5
500k	250	496.5	3313.5	0	243
	500	460	3290.5	0	269.5
	1,000	202	3212.5	0	261.5
	1,400	140	3126.5	0	267.5
700k	250	1264	3616	0	257.5
	500	550.5	3150.5	0	264.5
	1,000	282	3210.5	0	263
	1,400	230	3683	0	256.5



(그림 12) 핸드오버시 트래픽별 패킷 손실률

(그림 13)은 핸드오버시 MIPv6와 FMIPv6의 핸드오버 지연시간을 보여주고 있다. 그림의 상단의 모여 있는 것이 MIPv6의 핸드오버 지연을 나타낸 것이고 하단의 것이 FMIPv6의 것을 나타낸다. 상단의 MIPv6 선들은 3300ms정도를 기준으로 큰 폭으로 지연시간이 발생하고 있으며 하단에 모여 있는 FMIPv6는 거의 직선에 가깝게 동일한 핸드오버 시간이 발생했다. MIPv6은 단말이 서브넷 변경됨을 발견하는데 소요되는 시간과 CoA를 생성하고 검증하는 시간, CoA를 HA에 바인딩하는 시간들이 약 3300ms 정도 소요됨을 알 수 있다. 반면 FMIPv6는 트리거를 수신한 시점부터 신속하게 핸드오버가 수행되기 때문에 일정한 지연시간을 갖는다. 위 결과로 미루어 볼 때, 실제 무선 환경에서 AP들의 전파 신호를 기반으로 트리거를 발생시키는 트리거 관리자는 FMIPv6기법에 약 250 msec 핸드오버 지연시간을 보장할 수 있을 정도로 잘 구현되었음을 확인할 수 있다. 그리고 250 msec의 지연시간은 패킷 손실을 유발할 수 있으나 FMIPv6의 버퍼링 기법을 NAR에 잘 구현하여 무손실을 보장할 수 있어 FMIPv6는 실시간 응용 서비스에 seamless한 핸드오버 서비스를 제공이 가능하다.



(그림 13) 핸드오버 지연

5. 결론

본 논문은 제 2계층 트리거를 이용하여 3계층 핸드오버 성능을 개선한 FMIPv6를 리눅스 머신에 구현하였고 손실률과 지연시간 측면에서 MIPv6에 비해 우수함을 무선 환경 실험을 통해 고찰하였다. 2계층 트리거에 성능이 좌우 되는 FMIPv6를 효율적으로 동작시키기 위해 무선랜 디바이스 드라이버에 트리거 관리자를 구현하였고 트리거 관리자를 통해 무선랜 제 2계층 정보를 수집하고 적시에 제 2계층 트리거 발생시킬 수 있도록 하였다.

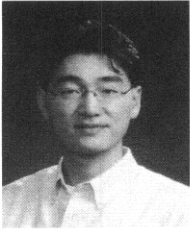
FMIPv6의 기능들 중 먼저 트리거를 이용하는 FMIPv6의 RtSolPr/PrRtAdv 메시지를 송수신할 때 통신할 때 필요한 주변 AR와 AP들의 정보들은 각 AR에 설정하여 관리하도록 하였다. 그리고 AR에는 새로 추가된 FMIPv6 메시지의 정보를 처리할 수 있도록 하고 손실방지를 위한 버퍼링 기법 및 패킷 터널링 기능 등을 구현하였다.

구현된 FMIPv6의 성능을 손실률과 핸드오버 지연시간 측면에서 분석하기 위해 MIPv6와 같이 동일한 실제 무선 환경에서 다양한 전송률과 패킷 사이즈 별로 핸드오버 실험을 수행하였다. 그 결과 전송량이나 패킷 사이즈 별로 다양한 패킷 손실을 보였던 MIPv6에 비해 FMIPv6에서는 0%의 손실을 보였다. 그리고 MIPv6는 3311ms의 핸드오버 지연이 발생하는데 비해 FMIPv6는 약 250ms정도의 지연시간을 가졌다. 이 결과로 부터 2계층 트리거를 이용한 FMIPv6를 이용하여 실시간 응용서비스에 Seamless 핸드오버를 제공할 수 있음 알 수 있었다.

참고 문헌

- [1] D. Johnson et al., "Mobility Support in IPv6," Internet Draft, draft-ietf-mobileip-ipv6-24.txt, December 2003.
- [2] Masaki Bandai and Sasase, "A Low Latency Handoff Scheme Using Positional Information for Mobile IP Based Networks," Globecom2003, 2003.
- [3] Youngjune Gwon and Alper Yegin, "Enhanced forwarding from the Previous Care-of Address(EFWD) for Fast Handovers in Mobile IPv6," WCNC2004, vol. 2, pp. 21-25, Mar., 2004.
- [4] T. Pagtzis, et al., "Proactive Seamless Mobility Management for future IP Radio Access Networks," Computer Communications, 2003.
- [5] IEEE 802.21, "A Proposal for MIH Function and Information Service," November, 2004.
- [6] Robert Hsieh, et al., "S-MIP: A Seamless Handoff Architecture for Mobile IP," Inform2003, 2003.
- [7] Kempf, J., et al., "Fast Mobile IPv6 Handover Packet Loss Performance: Measurements for Emulated Real-time Traffic," WCNC2003, Mar., 2003.
- [8] Rajeev Koodli, "FMIPv6 Handover for Mobile IPv6," Internet Draft, draft-ietf-mipshop-FMIPv6-mip6-01.txt, Jan., 2004.
- [9] Intersil, "PRISM Driver Programmers Manual version 3.00", Jul., 2003.
- [10] Marco L., et al., "Candidate Access Router Discovery," Internet draft, draft-ietf-seamoby-card-protocol-08.txt, Sep., 2004.
- [11] S. Deering and R. Hinden, "Internet Protocol, Version 6:(IPv6) Specification," RFC 2460, Dec. 1998.
- [12] A. E. Yegin et al., "Link-layer Triggers Protocol," draft-yegin-l2-triggers-00.txt, Dec., 2002.
- [13] MGEN: The Multi-Generator Toolset Homepage, <http://mgen.pf.itd.navy.mil/>

오 승 훈



2000년 숭실대학교 전자공학과 졸업
 2002년 숭실대학교 정보통신공학과(석사)
 2002년~현재 숭실대학교 정보통신공학과
 박사과정
 관심분야: 컴퓨터 네트워크, 인터넷 네트
 워킹, 인터넷 QoS, 멀티캐스팅,
 Xcast, IP-이동성, 무선 네트워
 크, 광인터넷 복원, 광버스트스
 워킹

이 성 식



2000년 군산대학교 전기공학과 졸업
 2004년 숭실대학교 정보통신공학과(석사)
 관심분야: 컴퓨터 네트워크, 무선 네트워
 크, 모바일 IP, IPv6, 실시간 멀
 티미디어 방송, 임베디드 시스템

김 영 한



1984년 서울대학교 전자공학과(공학사)
 1986년 한국과학기술원 전기 및 전자공학
 과(공학석사)
 1990년 한국과학기술원 전기 및 전자공학
 과(공학박사)
 1987년~1994년 디지콤정보통신연구소 데
 이터통신연구부장
 1994년~현재 숭실대학교 정보통신전자공학부 부교수, 통신학회
 인터넷연구회 위원장, VoIP포럼 차세대기술분과위원장
 관심분야: 컴퓨터네트워크, 인터넷 네트워킹, 이동 데이터 통신
 망등임.