

OAI PMIPv6를 이용한 FPMIPv6 테스트베드의 구현 및 분석

도미술, Khuong Quoc Anh, 박창용, 손민한, 추현승
성균관대학교 정보통신대학
e-mail: msdo56@skku.edu, kqanhdev@gmail.com,
{gspcy, minari95}@skku.edu, choo@ece.skku.ac.kr

An Implementation and Analysis of FPMIPv6 Test Bed Using OAI PMIPv6

Misol Do, Khuong Quoc Anh, Changyong Park,
Minhan Shon, Hyunseung Choo
College of Information and Communication Engineering,
Sungkyunkwan University

요 약

PMIPv6 (Proxy Mobile IPv6)는 네트워크 기반 이동성 지원 프로토콜로 MN (Mobile Node)의 이동성 프로토콜 스택 유무와 관계없이 MN의 이동성을 지원한다. 하지만 핸드오버 과정에서 이동성 관련 시그널링을 완료할 때까지 지연이 발생하고 핸드오버 동안 MN으로 향하는 패킷이 손실된다는 단점을 갖고 있다. 이러한 단점을 보완하기 위한 기법으로 FPMIPv6 (Fast Handover for PMIPv6)가 제안되었다. 본 논문에서는 PMIPv6의 오픈소스인 OAI PMIPv6 (OpenAirInterface PMIPv6)를 기반으로 FPMIPv6 테스트베드를 구축하고 MN의 핸드오버를 실험하여 실제 FPMIPv6 환경에서의 핸드오버 지연 및 패킷 손실 정도를 측정한다.

1. 서론

네트워크 기반 이동성 지원 프로토콜인 PMIPv6는 이동성 프로토콜 스택을 탑재하지 않은 MN에게도 이동성을 지원한다[1]. 또한 MN이 이동성 관련 시그널링에 참여하지 않으므로 무선 상에서의 터널링 오버헤드 및 무선 링크를 통한 이동성 관련 시그널링 메시지 교환이 감소하여 시그널링 비용을 줄일 수 있다[2]. 그러나 MN과 소속 네트워크의 연결이 끊어진 후에 핸드오버가 진행되므로 이 과정에서 지연이 발생하고, 핸드오버 동안 MN으로 전달되는 패킷에 대한 조치가 없어 해당하는 패킷이 손실된다는 단점을 갖고 있다.

핸드오버 지연을 줄이고 패킷 손실을 방지하기 위해 FPMIPv6이 제안되었다[3]. FPMIPv6은 MN의 핸드오버가 일어나기 이전에 기존의 MAG(Mobile Access Gateway)와 MN이 새로 접속할 MAG 간 시그널링을 시작하여 핸드오버 지연을 줄인다. 또한 핸드오버 중에 전달되는 패킷을 MN이 새로 접속할 MAG에서 버퍼링해둔 다음 핸드오버가 완료된 후 MN에게 전달하여 패킷 손실을 방지한다.

한편 이동성 지원 프로토콜의 중요성이 높아짐에 따라 이동성 지원 프로토콜의 실질적 구현에 대한 연구도 진행되어 왔다. 대표적인 MIPv6 구현 결과로는 GO-CORE 프로젝트 및 USAGI/WIDE group이 함께 개발한 MIPLv2가 있으며, 현재는 UMIP이라는 이름으로 USAGI group이 관리하고 있다. PMIPv6 표준화 이후 UMIP을 기반으로 한 PMIPv6 구현이 시도되었다. 특히 EURECOM이 개발

한 OAI PMIPv6 (OpenAirInterface PMIPv6)는 PMIPv6의 모든 기능을 성공적으로 구현하였다[4].

본 논문에서는 OAI PMIPv6를 바탕으로 PMIPv6의 보완된 버전인 FPMIPv6을 구현하고 MN의 핸드오버를 실험하며, 이를 통해 실제 FPMIPv6 환경에서의 핸드오버 지연 및 패킷 손실 정도를 측정한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 FPMIPv6의 핸드오버 절차 및 OAI PMIPv6의 구조에 대해 설명한다. 구현된 FPMIPv6 테스트베드에 대해서는 3장에서 설명하며, 이어지는 4장을 통해 FPMIPv6 테스트베드 하에서의 핸드오버 실험 결과를 제시하고 결과를 분석한다. 마지막으로 5장에서는 결론 및 향후 연구 진행방향에 대해 논의한다.

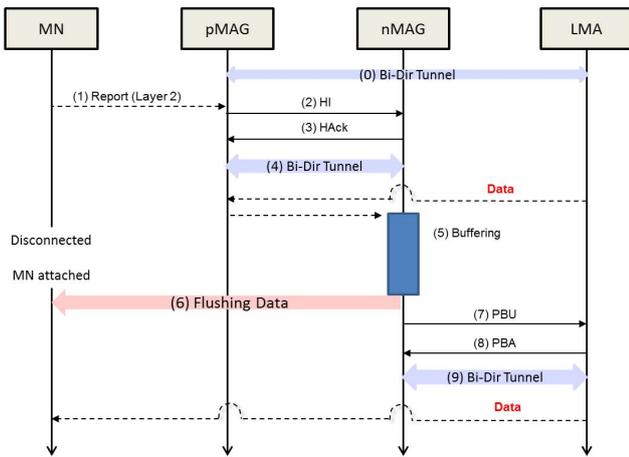
2. 관련연구

2.1 Fast Handover for PMIPv6

FPMIPv6은 PMIPv6의 단점인 핸드오버 시그널링 동안의 지연 및 패킷 손실을 줄이기 위해 제안되었다. 이를 위해 FPMIPv6은 MN이 핸드오버할 대상 MAG (nMAG)로 접속하기 전에, 기존의 MAG (pMAG)와 nMAG간의 터널을 생성하여 핸드오버에 필요한 정보를 미리 전달하고 MN으로 향하는 패킷을 버퍼링한다.

그림 1은 FPMIPv6의 핸드오버 절차를 나타낸다. 핸드오버가 임박함을 감지한 MN은 주변의 MAG를 스캔하여 새로 핸드오버할 nMAG를 찾는다. 다음으로 pMAG에게

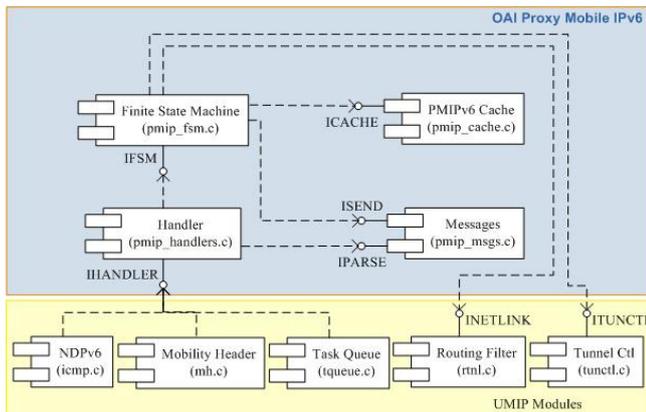
L2 Report 메시지를 보내 MN과 nMAG의 정보를 전달한다. 메시지를 받은 pMAG는 nMAG와 HI (Handover Initiate) 메시지 및 HAcK (Handover Acknowledge) 메시지를 교환하고 두 MAG 사이에 양방향 터널이 생성된다. 이후 LMA로부터 MN으로 전달되는 패킷은 pMAG와 양방향 터널을 거쳐 nMAG로 전달되며 nMAG는 전달된 패킷을 버퍼링한다. MN이 nMAG로 접속하면 nMAG는 버퍼링된 패킷을 MN으로 전송한다. 이와 동시에 LMA와 PBU 및 PBA 메시지를 교환하고 LMA와 nMAG간의 터널을 생성한다. 이후 MN으로 향하는 패킷은 LMA와 nMAG를 거쳐 MN으로 전달된다.



(그림 1) FPMIPv6의 핸드오버 과정

2.2 OpenAirInterface PMIPv6

OAI PMIPv6은 EURECOM에서 RFC 5213을 기반으로 구현한 PMIPv6 오픈소스이다. MIPv6의 오픈소스인 UMIP을 기반으로 개발되었으며 Ubuntu 및 기타 리눅스 운영체제에서 동작한다.



(그림 2) OAI PMIPv6의 구조

그림 2는 OAI PMIPv6의 아키텍처를 보여준다. 하단의 5개 모듈은 UMIP에서 이어받은 모듈로, UMIP 코드의 모듈 중 상당수가 OAI PMIPv6에서도 재사용되고 있다. 상

단의 4개 모듈은 PMIPv6의 기능을 구현하기 위해 개발된 것이다. Handler 모듈은 모든 메시지와 이벤트를 입력받아 Finite State Machine 모듈로 전달한다. Finite State Machine 모듈은 전달된 메시지를 분석하여 수행할 동작을 결정하고 다른 모듈들을 제어한다. PMIPv6 Cache 모듈은 바인딩 캐시의 역할을 담당하며, Message 모듈은 PMIPv6 프로토콜에 따라 입력된 메시지를 분석하고 출력될 메시지를 규격화한다.

3. FPMIPv6 테스트베드의 구현

본 테스트베드는 C 언어로 구현되었으며 운영체제로는 Ubuntu 10.04, 리눅스 커널 버전 2.6.32를 사용한다. 또한 RFC 5949 및 기존에 구현된 OAI PMIPv6를 기반으로 한다. 그림 1의 시그널링 절차를 바탕으로 FPMIPv6 구현의 요소를 설명한다.

(1) L2 Report: syslog 메시지를 이용하여 MN으로부터 pMAG로 L2 Report 메시지를 보낸다. 이때 새로 접속할 nMAG의 정보를 실어서 보낸다.

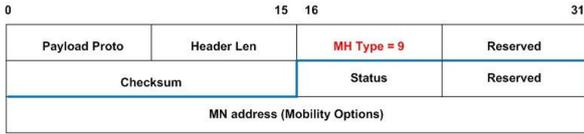
(2) HI 메시지: HI 메시지에는 MN의 주소가 포함되어 있으며, 이 정보는 이후의 터널 설정을 위해 사용된다. 본 구현에서 HI 메시지는 PMIPv6의 새로운 시그널링 메시지로 정의되었으며, 정보 전달을 위해 IPv6 확장헤더 중 하나인 Mobility header를 사용한다[5]. Mobility header는 모바일 노드의 이동성과 관련된 메시지를 주고받기 위한 확장 헤더로 MN, CN(Correspondent Node), HA(Home Agent) 간 바인딩 생성 및 관리를 위해 사용된다.

0		15	16		31
Payload Proto		Header Len		MH Type = 8	Reserved
Checksum			Reserved		
MN address (Mobility Options)					

(그림 3) HI 메시지의 Mobility header 형식

그림 3은 HI 메시지의 Mobility header 형식을 나타낸다. Payload Proto 필드는 다음에 나타날 확장 헤더를 가리키는데, MIPv6 프로토콜에서는 Mobility header가 항상 IPv6 헤더 중 가장 마지막에 위치하도록 정의하였으므로 이 값은 항상 58(IPv6-NONEXT, 다음 헤더가 없음을 나타냄)의 값을 가진다. Header Len 필드는 Mobility header의 총 길이를 나타낸다. MH Type 필드는 메시지의 타입을 나타내며, MIPv6/PMIPv6 프로토콜에서는 0에서 7까지 총 8개의 메시지를 정의하고 있다. 본 구현에서는 HI의 MH Type을 8로 정의하였다.

(3) HAcK 메시지: HAcK 메시지는 HI 메시지와 유사한 방법으로 구현되었다. 그림 4는 HAcK 메시지의 Mobility header 형식이다. MH Type으로는 9를 사용하며, Status 필드는 시그널링 과정의 성공/실패 여부를 나타낸다. 그 외 필드의 의미는 HI 메시지와 동일하다.



(그림 4) HAcK 메시지의 Mobility header 형식

(4) 터널링: pMAG와 nMAG 사이의 터널을 생성한다. 이때 pMAG에서는 CN에서 MN으로 향하는 패킷{source addr = any, destination addr = MN addr}이 AP1 대신 터널로 전송될 수 있도록 라우팅 경로 및 규칙을 설정한다. 반대로 nMAG에서는 MN에서 CN으로 향하는 패킷{source addr = MN addr, destination addr = any}이 터널로 전송될 수 있도록 라우팅 경로 및 규칙을 설정한다.

(5) 버퍼링: MN으로 향하는 패킷이 pMAG에서 터널을 통해 nMAG로 전달되면, nMAG는 MN이 접속하기 전까지 해당되는 패킷을 버퍼링한다. 패킷 버퍼링 모듈은 총 3 파트로 나뉜다.

- 패킷 버퍼링 모듈: PMIPv6 소스코드에 새로 추가된 모듈이다.

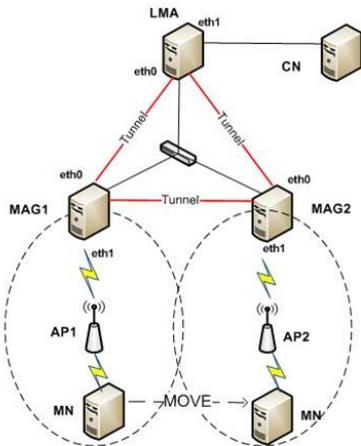
- IPTables tool: 유저 스페이스에 위치하며 Netfilter framework와 패킷 버퍼링 모듈이 서로 통신할 수 있도록 한다. 또한 IPTables의 일부인 Libipq 라이브러리는 IPv6 규약의 통신을 담당하여 버퍼링될 패킷을 전달받고 버퍼링이 끝난 패킷을 다시 전달한다.

- Netfilter framework: 커널 스페이스에 위치하며 실제 네트워크 계층에서 주고받는 패킷을 분석하고 버퍼링할 패킷을 걸러낸다.

(6) 플러싱: MN의 접속을 감지한 nMAG는 MN으로 향할 패킷{source addr = any, destination addr = MN addr}이 AP2를 거쳐 MN으로 전달될 수 있도록 라우팅 규칙을 변경하고, 버퍼에 저장된 패킷을 모두 전송한다. 본 구현상에서 플러싱은 버퍼링 모듈에서 함께 담당한다.

4. 실험 결과 및 분석

4.1 실험 환경



(그림 5) FPMIPv6 테스트베드의 구성

본 테스트베드의 구조는 그림 5와 같이 구성되며 실험에 필요한 파라미터는 표 1에 정의되어 있다. 실험은 $r_1 > r_2$ 인 경우(시나리오 1)와 $r_1 = r_2$ 인 경우(시나리오 2)로 나누어서 진행하며 각 실험에서 사용된 파라미터의 값은 표 2에 나타나 있다.

<표 1> 실험에 사용된 파라미터

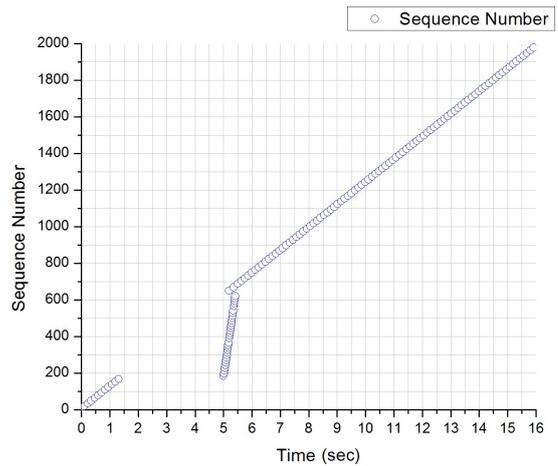
이름	정의
r_1	플러싱을 사용할 경우의 패킷 전송률
r_2	일반적인 경우의 패킷 전송률
p	전송된 패킷의 양

<표 2> 각 실험에 사용된 파라미터의 값

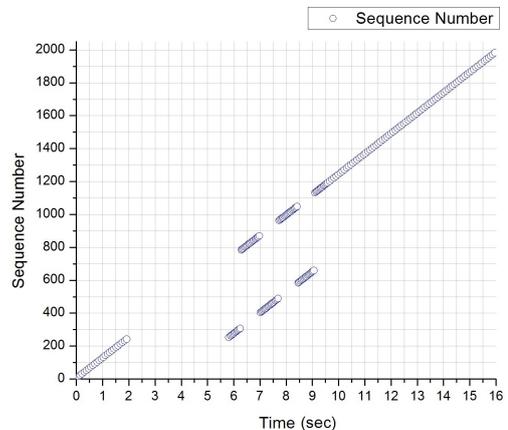
이름	시나리오 1	시나리오 2
r_1	1280pkg/s	128pkg/s
r_2	128pkg/s	128pkg/s
p	2048	2048

4.2 실험 결과

각 시나리오의 실험 결과는 그림 6과 그림 7에 나타나 있다.



(그림 6) 시나리오 1의 결과 그래프



(그림 7) 시나리오 2의 결과 그래프

참고문헌

실험 결과 두 시나리오 모두에서 핸드오버 이후 버퍼링된 패킷 및 새로운 패킷이 잘 전송되고 있음을 볼 수 있다. 그러나 일정 구간에서 out of sequence 문제가 발생하는데, 버퍼링되었던 패킷과 새로 전송되는 패킷이 교대로 전송되고 있다. 이는 버퍼링된 패킷이 충분한 시간 안에 전달되지 못하였기 때문이며, 시나리오 1에 비해 시나리오 2에서 out of sequence 현상이 더 심각하게 일어난다는 점에서 확인할 수 있다.

FPMIPv6에서는 버퍼링된 패킷을 포워딩하기 시작함과 동시에 nMAG와 LMA 사이의 양방향 터널을 생성하며 이때 버퍼링된 패킷이 터널 생성 이전에 모두 전달되는지에 대해서는 고려하지 않고 있다. 가장 간단한 해결책으로는 플러싱을 하는 방법이 있으나, 플러싱은 네트워크에 부하를 가져오므로 적절한 수준으로 조절할 필요가 있다. 또한 일부 응용 프로그램의 경우 패킷 전송률이 과도하게 증가하면 패킷을 제대로 처리하지 못할 수 있다. 따라서 FPMIPv6 테스트베드를 실제로 구현 및 운영하는 경우 버퍼링으로 인한 out of sequence 문제를 해결하여야 한다.

5. 결론 및 향후 연구 과제

본 논문에서는 OAI PMIPv6를 바탕으로 한 FPMIPv6 구현을 소개하고 실제 MN의 핸드오버 실험 결과를 제시하였다. 그러나 실험 결과 버퍼링된 패킷의 포워딩이 nMAG와 LMA 사이의 양방향 터널 생성 이전에 끝나지 않을 경우 out of sequence 문제가 발생하였다. 향후 테스트베드의 개선을 통해 이러한 문제점을 보완하여야 할 것이다.

Acknowledgement

본 연구는 지식경제부(정보통신산업진흥원) 대학 ITRC, 교육과학기술부(한국연구재단)의 차세대정보컴퓨팅기술개발사업 및 중점연구소지원사업의 일부지원으로 수행되었음.(NIPA-2012-(H0301-12-3001),2012-0005861, 2012-0006420). 교신저자: 추현승

[1] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6," IETF RFC 5213, August 2008.

[2] K. Kong, W. Lee, Y. Han, M. Shin, "Mobility Management for All-IP Mobile Networks: Mobile IPv6 vs. Proxy Mobile IPv6," IEEE Wireless Communications, June 2008.

[3] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, and F. Xia, "Fast Handovers for Proxy Mobile IPv6," IETF RFC 5949, September 2010.

[4] OpenAirInterface, OpenAirInterface PMIPv6 (OAI PMIPv6) [Internet], <http://www.openairinterface.org/openairinterface-proxy-mobile-ipv6-oai-pmipv6>

[5] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," IETF RFC 3775, June 2004