핸드오버 지연 시간을 줄일 수 있는 심리스 핸드오버 프로토콜 제안 및 성능 분석

장종민* · 이유태* · 이승희** · 김동일* · 최삼길*

Seamless handover protocol to reduce handover latency

Jong Min Jang* · Yutae Lee* · Soong-Hee Lee** · Dong-il Kim* · Sam-gil Choi*

이 논문은 2008 년도 동의대학교 교내연구비(과제번호 2008AA171)와 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업(IITA-2009-C1090-0801-0013)의 연구결과로 수행되었음.

요 약

무선 이동 통신기술과 이동 단말의 발달로 이종망간 핸드오버에 대한 요구가 증가하고 있다. **IETF**는 **All-IP** 기반 네트워크에서 이종 엑세스 네트워크 사이의 핸드오버를 위한 표준을 정의하는 **MIPv6**와 **FMIPv6**를 개발하였다. **MIPv6**는 단말의 핸드오버가 수행되는 동안 긴 지연 시간과 패킷 손실을 유발함으로써 시간에 민감한 서비스의 세션 연속성을 보장하지 못했다. **FMIPv6**는 핸드오버 준비 과정에서 주소를 미리 구성하고 엑세스 라우터가 패킷을 버퍼링 하도록 함으로써 패킷 손실의 가능성을 줄였다. 그러나 **FMIPv6**는 여전히 핸드오버의 마지막 과정으로 바인딩 갱신을 수행해야 하는데, 이로 인한 지연 시간은 실시간 애플리케이션에 지연을 유발한다. 본 논문에서는 **FMIPv6**와 호환가능하며 바인딩 갱신에 필요한 시간을 줄일 수 있는 새로운 프로토콜을 제안하고, 성능분석을 통해 기존 프로토콜의 핸드오버 지연시간과 비교한다.

ABSTRACT

With significant development in wireless communication technologies, the needs to handover from one network to another have grown dramatically. Recently IETF developed MIPv6 and FMIPv6 which are used to handover between different access networks. These handover mechanisms have some known weaknesses. MIPv6 can not serve time-critical application because it causes long handover latency and packet loss. By these reasons, MIPv6 does not guarantee session continuity. FMIPv6 uses handover preparation phase to reduce DAD latency. However FMIPv6 still suffers from long handover latency, because it must perform binding update at the end of handover procedure. In this paper we propose new handover mechanism FMIPv6-PBU which eliminates binding update latency. The proposed mechanism has compatability with lagacy FMIPv6. In this paper propose FMIPv6-PBU handover procedure and explain its procedures and evaluate it with MIPv6 and FMIPv6.

키워드

MIPv6, FMIPv6, L3 Handover procedure

* 동의대학교
** 인제대학교

접수일자 2008. 09. 25

Ⅰ. 서 론

이동성 지원 기능은 현대의 무선 통신 네트워크에서 가장 중요한 부분 중 하나이다. 핸드오버관리 기능은 이동 단말이 여러 링크 사이를 이동하는 동안 물리적인 위치 이동에 투명하게 통신 할 수 있도록 한다. 최근 다양한 기술을 사용하는 네트워크에 접속할 수 있는 다중 모드 단말의 수가 늘어나면서 다중망간 핸드오버에 대한 수요를 증가시켰다. 서로 다른 기술과 장비를 사용하는 다중망간 핸드오버는 표준화된 절차가 정의되지 않았으므로 이종망간 핸드오버는 달성하기 매우 어려운 과제이다. 다행히 많은 무선 통신 네트워크와 단말이 IP 계층과 융합하면서 이종망간 핸드오버 과정을 링크 계층 핸드오버와 IP 계층 핸드오버로 분리해서 다룰 수 있게 되었다. 핸드오버 과정이 두 계층에서 이루어지기 때문에 이 두 계층의 핸드오버를 조정하는 기능이 네트워크 스택의 특정한 계층에 삽입되어야 한다. All-IP 기반 네트워크에서 핸드오버는 IP 계층에서 더 잘 다룰 수 있으므로 핸드오버를 위한 기능은 IP 계층에 삽입된다[1].

All-IP 네트워크에서 IP 계층의 핸드오버를 다루는 대표적인 핸드오버 프로토콜은 MIPv6 (Mobile IPv6) [2] 이다. 하지만 이 프로토콜은 몇몇 단점 때문에 이를 보완하는 여러 확장들이 제안되고 있다. MIPv6의 대표적인 단점은 핸드오버 과정에서 발생하는 긴 지연시간이다. 이동 단말이 PoA (Point of Attachement)를 변경하면 MIPv6 프로토콜은 단말이 이동한 PoA에서 IP 핸드오버가 필요한지를 확인해야 한다. MIPv6가 IP 핸드오버가 필요한지 판단하기 위해서는 L2 계층의 핸드오버가 선행되어야 하는데 이는 MIPv6의 이동 확인 메커니즘이 단말이 기존에 연결되어 있던 라우터와의 직접적인 연결이 가능한지를 기준으로 삼기 때문이다. 단말이 새로운 서브넷에서 사용할 주소를 획득하여 유일성을 검증하는 Duplicate Address Detection (DAD) 과정도 매우 긴 지연을 발생한다. MIPv6에서 핸드오버를 수행한 단말은 삼각라우팅으로 인한 지연을 제거하기 위해 단말이 이동한 서브넷 네트워크에서 사용할 주소를 등록하는 Correspondent Registration (CR) 절차를 수행해야 한다. CR 과정은 최소 세 번의 네트워크 왕복이 필요하므로 DAD와 더불어 추가적인 지연을 유발한다. 긴 지연시간은 실시간 애플리케이션에 심각한 지연이나 세션의 종료를 유발한다.

MIPv6의 긴 핸드오버 지연시간을 보완하기 위해 FMIPv6 (Mobile IPv6 Fast Handover) [3] 프로토콜이 제안되었다. FMIPv6 는 네트워크 발견과 같은 링크 계층 정보를 획득하여 사용함으로써 핸드오버 과정에서 긴 처리시간을 필요로 하는 작업들을 미리 수행한다. 단말이 이동 중 새로운 네트워크를 발견하고 IP계층 핸드오버가 필요하다고 판단하면 FMIPv6는 단말의 현재 링크의 연결이 종료되기 전에 새로운 서브넷에서 사용할 IP 주소를 미리 구성하므로 DAD 과정에 소요되는 지연 시간을 제거한다. 그러나 FMIPv6는 여전히 CR 과정을 수행해야 하므로 MIPv6의 모든 지연 요소를 제거하지 못하였다.

본 논문에서는 FMIPv6 프로토콜을 개선하여 단말이 새로운 서브넷 네트워크로의 이동을 감지하고 핸드오버를 준비하는 과정에서 CR을 미리 수행하도록 요청하는 FMIPv6-PBU (Proxy Binding Update) 프로토콜을 제안한다. 이것은 FMIPv6의 확장으로서 기존의 FMIPv6와 호환 가능하다. 이 프로토콜은 단말이 새로운 서브넷을 발견하면 단말은 이동할 네트워크의 Access Router (nAR) 가 단말을 가장하여 CR 과정을 수행하도록 요청한다. 이를 통해 단말은 새로운 네트워크로 접속한 직후 새로운 주소를 사용하여 상대 단말과 직접 통신할 수 있다. 이렇게 함으로써 FMIPv6에서 CR이 완료되기 전까지 유지되어야 했던 터널링으로 인한 패킷 왕복 지연과 터널링 유지를 위한 부하를 줄일 수 있다. 또한 단말이 새로운 서브넷에 접속한 직후 패킷을 송신 할 수 있으므로 단말의 상향 링크 패킷 버퍼링 줄인다.

2절에서는 본 논문의 관련연구로 대표적인 핸드오버 프로토콜인 MIPv6와 FMIPv6의 핸드오버 과정을 기술한다. 3절에서는 핸드오버 지연시간을 줄일 수 있는 새로운 프로토콜을 제안하고, 4절에서 제안한 프로토콜의 성능을 정량적으로 분석한다. 또한 제안된 프로토콜과 기존 프로토콜과 비교 제시하여 제안된 프로토콜의 우수함을 제시한다.

Ⅱ. 관련 연구

2.1. MIPv6

MIPv6 프로토콜은 이동 단말 (MN)이 현재 연결된 Access Router (oAR) 에서 새로운 Access Router (nAR) 로

이동하는 동안 MN이 상대 단말 (CN) 과 네트워크 계층 (L3) 연결성을 보장하기 위해 개발 되었다. MIPv6 프로토콜은 MN이 두 개의 주소 사용하도록 한다. 하나는 MN이 홈링크에 접속해 있을 때 사용하는 주소인 Home Address (HoA)이며 이 주소는 MN의 전역 주소로 사용된다. 다른 하나는 MN이 홈 링크 이외의 외부 링크에 접속되어 있을 때 MN의 실제 접속 위치를 나타내는 주소인 Care-of Address (CoA)이다. MN이 홈 링크에 접속되어 있을 때 MN과 CN은 일반적인 IP 라우팅 메커니즘을 통해 패킷을 교환한다. MN이 외부 링크로 이동하면 MIPv6는 서브넷의 변경으로 인한 L3 핸드오버가 필요한지 확인하기 위한 절차를 수행한다. MIPv6는 MN의 이동 확인을 위해 일반적으로 Neighbor Unreachability Detection [5] 을 사용한다. 이를 통해 기존 라우터가 도달 불가능함을 판단함으로써 새로운 서브넷으로 이동했음을 감지한다.

MN이 L3 핸드오버를 감지하면 새로운 서브넷에서 사용할 CoA를 구성해야 한다. CoA 구성에는 일반적인 IPv6 메커니즘이 사용된다 [4][5]. 새로운 CoA (nCoA) 를 구성하면 이 주소가 외부 링크에서 유일한지 확인하기 위해 nCoA에 대한 DAD가 수행된다. DAD 과정이 성공적으로 끝나도 MN은 nCoA를 송신자로 설정하여 CN으로 전달 할 수 없다. 왜냐하면 CN은 MN의 주소 변경을 알지 못 하므로 알려지지 않은 주소로부터의 패킷을 폐기할 것이다. MN은 자신의 새로운 위치를 알리기 위해 CR 과정을 수행한다. 이 과정은 CN이 MIPv6를 지원하는지 여부에 따라 두 가지 모드로 동작한다.

CN이 MIPv6 프로토콜을 사용하지 않는다면 MN은 BU를 홈링크에 위치한 Home Agent (HA)에게 전송함으로서 HoA와 nCoA를 연관시킨다. CN은 MN의 새로운 주소에 대해 알지 못하므로 여전히 패킷의 목적지를 HoA로 설정하여 전달한다. 홈링크의 HA는 목적지가 HoA인 패킷들을 수신하면 MN의 현재 위치인 nCoA로 패킷들을 터널링한다. 또한 MN은 송신자를 CoA로 설정하여 CN에게 직접 패킷을 전달 할 수 없다. 따라서 MN은 CN으로의 패킷을 HA로 전달하고 HA는 패킷을 CN 으로 역터널링 한다. 이러한 터널링은 불필요한 라우팅 비용과 지연 시간을 발생시킨다.

CN이 MIPv6를 지원한다면 이러한 터널링은 불필요하다. MN은 CN에게 직접적으로 BU를 전송함으로써 MN의 nCoA를 알려줄 수 있다. 이 과정은 보안등의 이유로 return routability (RR) 과정 [2] 이 선행되어야 한다. RR 과정은 일반적으로 두 번의 네트워크 왕복이 필요하다. RR 과정이 끝나면 MN과 CN은 안전하게 BU를 수행할 수 있다. CN은 nCoA를 MN의 주소와 연관시킨다. BU 과정이 끝나면 MN과 CN은 MN의 실제 위치를 사용하여 통신하므로 터널링으로 인한 지연이나 삼각 라우팅 문제 해결할 수 있다.

2.2. IETF FMIPv6

IETF FMIPv6는 기존 MIPv6가 정의한 핸드오버 절차에 따른 다소 긴 서비스 중단 시간과 패킷 손실 문제를 해결하기 위해 제안 되었다. MIPv6와 달리 FMIPv6는 링크 계층 정보를 사용하여 핸드오버의 가능성을 미리 알 수 있다. MN이 L3 핸드오버가 필요함을 감지하면 oAR과의 연결이 끊어지기 전에 핸드오버 준비 과정을 수행한다. 핸드오버 준비 과정은 MIPv6에서 핸드오버 지연 시간 중 많은 부분을 차지하는 주소 구성 및 DAD를 미리 수행하므로 핸드오버로 인한 패킷 손실과 시간 지연을 줄인다. 핸드오버 준비 과정 덕분에 MN은 nAR로 이동하는 즉시 버퍼링된 패킷을 수신하고 패킷을 전달할 수 있다.

그림 1은 FMIPv6의 핸드오버 절차를 나타내고 있다. 그림 1의 핸드오버 과정을 간략히 정리하면 다음과 같다.

1. MN이 새로운 서브넷 네트워크를 감지하면 MN은 신호 강도등의 물리 계층 정보를 바탕으로 핸드오버를 수행할지를 결정한다.
2. 새로운 네트워크에 대한 서브넷 접두사 정보를 얻기 위해 단말은 Router Solicitation for Proxy (RtSolPr) 메시지를 oAR에게 전송한다. RtSolPr을 수신한 oAR은 해당 nAR에 대한 서브넷 접두사 정보를 포함하는 Proxy Router Advertisement (PrAdv) 메시지를 전송한다. PrAdv 메시지로부터 새로운 서브넷 접두사를 수신한 MN은 이 값으로부터 nCoA 를 구성한다.
3. MN은 핸드오버를 시작하기 위해 oAR에게 Fast Binding Update (FBU) 메시지를 전송한다. FBU 메시지를 수신한 oAR은 핸드오버 과정에서 패킷 손실을 막기 위해 MN으로의 패킷을 버퍼링한다. 이 패킷은 MN이 핸드오버를 완료하면 MN에게 전달된다.
4. oAR은 nAR로 Handover Initiation (HI) 를 전달한다. HI 를 수신한 nAR은 MN의 nCoA에 대한 DAD를 수행하

고 해당 주소에 대한 새로운 사용 요청을 막는다. nAR은 HI를 승인하기 위해 oAR에게 HAck를 전달한다. nAR로부터 HAck를 수신한 oAR은 MN에게 FBAck 메시지를 전송한다. 이때 MN이 oAR로부터 FBAck를 전달받지 못할 경우를 대비해 nAR로도 FBAck를 전달한다. FBAck을 전송한 oAR은 버퍼링한 데이터를 nAR에게 전송하고 nAR은 이 데이터를 MN이 수신 가능 해질 때까지 버퍼링 한다.

5. MN은 FBAck 메시지를 수신하면 가능한 빨리 L2 핸드오버를 수행해야 한다. MN은 oAR로 부터의 전파 수신강도가 갑자기 매우 낮아져 oAR로 부터 FBAck을 수신하지 못하고 접속이 끊어질 수 있다. 이 경우 FBAck 은 nAR이 전달해 준다.
6. nAR 로 이동한 MN은 자신이 도달 가능함을 알리기 위해 nAR에게 Unsolicited Neighbor Advertisement (UNA) 메시지를 전송한다. UNA를 수신한 nAR은 버퍼에 들어있는 패킷을 MN으로 전달한다.
7. MN은 CN이 MN의 새로운 위치인 nCoA를 등록할 수 있도록 Binding Update (BU) 메시지를 CN에게 전송한다.

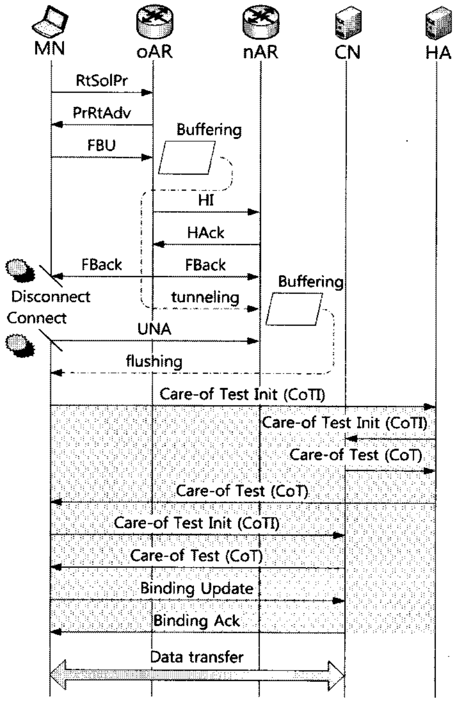


그림 1. FMIPv6 핸드오버 과정
Fig 1. FMIPv6 Handover procedure

2.3. MIPv6와 FMIPv6의 핸드오버 지연 시간 분석

MIPv6에는 단말이 nAR로 이동한 후 이를 감지하기 위한 방법이 존재하지 않는다. 이 때문에 MIPv6가 전송할 패킷이 가지고 있지 않다면 MN이 새로운 서브넷을 발견하기 위해서는 Router Advertisements (RA) 메시지를 감지해야한다. [5]에서 RA의 주기를 RETRANS_TIMER로 정의하고 이것의 기본값은 1000ms로 정의된다. 따라서 단말은 평균적으로 L2 핸드오버가 완료 된 뒤 500ms가 지나야 주소 구성을 시작한다. 또한 RA를 통해 L3 핸드오버를 감지한 후에도 새로운 주소를 구성하여 유일성을 보장하기 위한 DAD 과정과 BU 과정의 지연시간이 발생한다. MIPv6는 핸드오버과정에서 CN의 패킷을 버퍼링하지 않으므로 핸드오버 과정에서 CN이 전송한 패킷은 모두 유실된다.

FMIPv6는 MIPv6의 긴 핸드오버 지연 시간과 패킷 유실을 해결하기 위해 MIPv6의 핸드오버 과정을 변형시켰다. FMIPv6에서 L3 핸드오버는 L2 핸드오버 이전에 시작된다. FMIPv6 프로토콜은 핸드오버 준비 과정에서 FBU 메시지를 oAR에게 전송함으로써 CN으로부터의 패킷을 버퍼링한다. 또한 oAR과 nAR은 버퍼링된 패킷을 교환하고 MN이 BU가 완료될 때까지 MN의 패킷을 터널링 함으로써 핸드오버 지연으로 인한 패킷 유실이 발생하지 않는다. 각 프로토콜의 핸드오버 지연시간과 패킷 유실 시간은 [1] 에서 자세히 다루고 있다. 그림 2 [6] 는 MIPv6와 FMIPv6의 핸드오버 지연 시간을 비교한다.

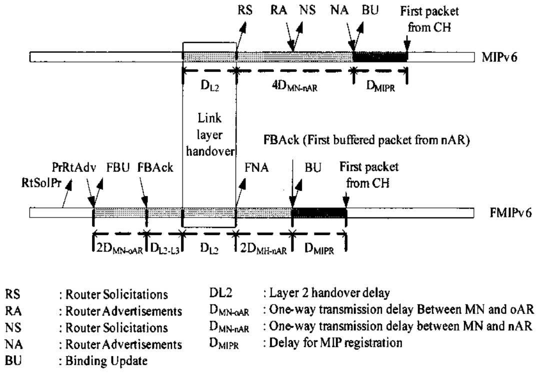


그림 2. MIPv6와 FMIPv6 핸드오버 지연 시간 비교[6]
Fig2. Comparison of Handover Latency between MIPv6 and FMIPv6 [6]

FMIPv6가 핸드오버를 위한 준비 과정을 수행함에도 불구하고 FMIPv6는 여전히 핸드오버 지연 요소를 포함하고 있다. MN이 L2 핸드오버 이후 즉시 사용할 수 있는 nCoA 구성하고 있더라도 CN은 nCoA와 바인딩 되어있지 않기 때문에 MN은 BU 과정 없이 이 주소를 송신자로 설정하여 상대 단말에게 전송할 수 없다. 이 때문에 MN과 CN 사이의 패킷은 oAR과 nAR사이에서 터널링되어야 하며, 이때문에 BU 과정이 완료되기 전까지 MN과 CN사이에 패킷 지연이 발생한다. 사용중인 서비스의 성격에 따라 이러한 지연시간은 서비스에 심각한 지연을 유발하거나 세션을 종료 시킬 수 있다.

Ⅲ. FMIPv6-PBU 핸드오버 과정 제안

MIPv6와 FMIPv6 프로토콜은 MN이 L3 핸드오버를 수행한 이후 CR 과정이 수행되므로 이를 위한 BU 과정을 위한 추가적인 핸드오버 지연을 겪어야 한다. BU 과정은 일반적으로 RR 과정을 동반하므로 MN이 CN으로부터 직접적으로 패킷을 전송받기까지 몇 번의 네트워크 왕복 시간이 필요하다. 이러한 지연은 음성이나 동화상 등의 실시간 애플리케이션에게 치명적인 서비스 품질의 저하나 세션의 단절을 발생시킨다.

본 절에서는 CR 로 인한 핸드오버 지연을 제거하기 위해 이동 단말이 새로운 서브넷 네트워크로 핸드오버 할 경우 BU를 미리 수행하는 FMIPv6-PBU를 제안한다. FMIPv6에서 MN의 핸드오버는 준비, L2 핸드오버, 바인딩 갱신의 세 단계로 이루어졌다. FMIPv6-PBU는 MN이 L2 핸드오버를 수행하는 동안 이동할 네트워크의 nAR이 MN을 가장하여 해당 MN이 새로운 네트워크로 이동한 후 수행할 BU과정을 미리 수행한다. 그림 4 는 제안한 프로토콜과 기존 프로토콜의 핸드오버 지연 시간을 비교한다.

제안된 프로토콜은 FMIPv6 프로토콜의 메시지의 모든 의미를 유지하며 새로운 프로토콜을 인식할 수 있는 네트워크 엔티티들의 추가적인 동작을 지시하기 위해 FMIPv6의 HI, HAck, FBAck 메시지 헤더의 할당되지 않은 영역에 해당 메시지가 추가적인 동작을 수행하도록 지시하는 PBU 옵션을 포함한다. 이 새로운 옵션을 인식하지 못하는 라우터들은 이 메시지의 옵션에 지정된 동작을 수행하지 않고 PBU가 포함된 Ack를 전달 받지 못한다. 이 경우 MN은 FMIPv6 모드로 동작한다. FMIPv6-PBU 프로토콜은 기본적으로 FMIPv6과 동일하게 동작하며 일부 메시지만 차이가 있으므로 FMIPv6 와의 차이점을 중심으로 기술한다.

1. MN이 새로운 서브넷 네트워크를 발견하고 핸드오버를 준비하면 MN은 원형 FBU 메시지 내부에 새로운 옵션인 PBU 가 1로 설정된 FBU-PBU 메시지를 oAR에게 전달한다.
2. oAR은 nAR에게 HI 메시지를 전달 할 때 새로운 옵션을 확인하고 옵션이 1이라면 HI 메시지에 새로운 옵션인 PBU 에 1을 설정하여 전달한다.
3. 만약 nAR이 PBU 를 인식하면 nAR은 MN을 가장하여 CN에게 바인딩 업데이트를 요청할 것이다. nAR은 PBU 옵션이 설정된 HAck-PBU를 oAR에게 전달한다. oAR은 MN으로의 패킷을 버퍼링하고 MN이 nAR로 이동한 뒤 BU를 수행할 필요가 없다는 것을 알리기 위해 PBU 옵션을 설정한 HAck-PBU 메시지를 oAR에게 전달한다.
4. oAR은 HAck-PBU의 옵션을 검사하고 PBU동작이 수행될 것임을 MN에게 알리기 위하여 FBAck-PBU를 전달한다. 이때 oAR은 MN의 갑작스런 연결 단절을 대비하여 FBAck-PBU를 nAR에게도 전송함으로써 MN이 nAR에 도착한 직후 해당 메시직를 전달 받을 수 있도록 한다.
5. MN이 nAR로 이동한 뒤 FBAck-PBU를 받으면 nAR은 즉시 oAR 에게 Proxy Binding Update Success 옵션이 1로 설정된 FBAck-PBU를 전달함으로써 oAR과 nAR사이에 설정된 터널을 종료한다. 이렇게 하면 oAR이 속한 네트워크에 터널링으로 인한 부담을 줄일 수 있다.

MN이 nAR의 링크에 접속하여 UNA를 전송한 시점에 nAR과 CN 사이의 PBU 가 완료되면 nAR은 oAR과의 터널을 종료한다. 만약 MN이 nAR의 링크에 접속한 후 지정된 시간동안 FBAck-PBU를 받지 못하면 MN은 CR 과정을 수동으로 실행할 수 있다. 이 과정은 FMIPv6와 동일하다.

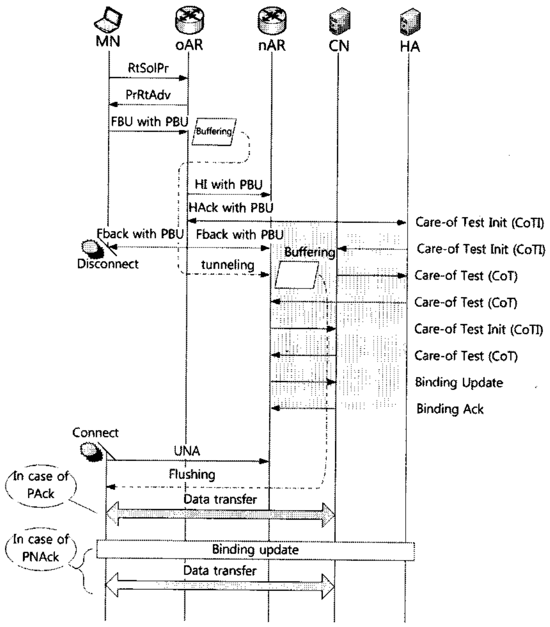


그림 3. FMIPv6-PBU 핸드오버 과정
Fig 3. FMIPv6-PBU Handover procedure

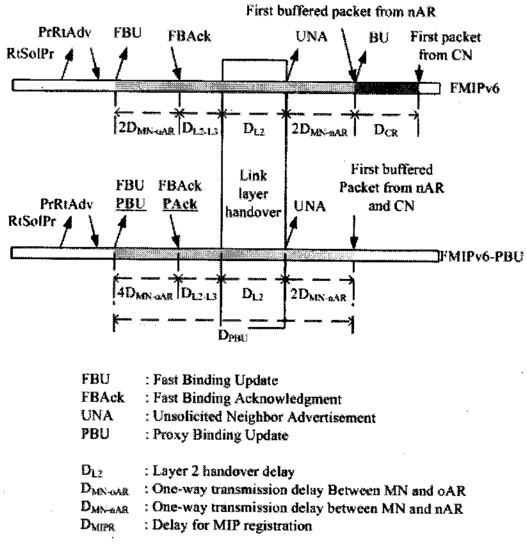


그림 4. FMIPv6과 FMIPv6-PBU의 핸드오버 지연시간 비교
Fig 4. Comparison of handover latency between FMIPv6 and FMIPv6-PBU

Ⅳ. 핸드오버 성능 분석

본 절에서는 제안한 프로토콜과 기존의 프로토콜의 핸드오버 지연 시간을 정량 분석한다. 본 절에서는 어떤 서브넷에서 서비스를 받는 단말이 다른 서브넷으로 이동하는 경우를 고려하며, 이때 사용된 각 단계별 지연/처리 시간은 표 1 과 같이 가정하였다[7].

표 1. 지연/처리 시간.
Table. 1. Delay/Processing Time.

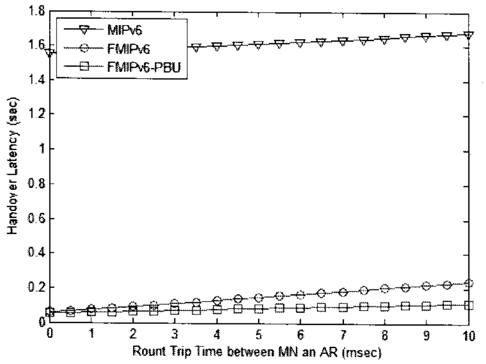
Type	Time
$D_{wireless}$	0 ms ~ 10 ms
D_{wire}	2 ms
D_{L2}	50 ms
D_{init}	$RTT_{MN-oAR} + RTT_{AR}$
D_{UNA}	$D_{MN-nAR} = D_{wireless}$
D_{RD}	500 ms
D_{DAD}	1000 ms
D_{CR}	$RTT_{MN-HA} + RTT_{CN-HA} + 2RTT_{MN-CN}$
D_{PBU}	$RTT_{nAR-HA} + RTT_{CN-HA} + 2RTT_{nAR-CN}$

여기서 $D_{wireless}$는 MN과 AR사이와 같은 무선채널의 단방향 패킷 전송 지연시간이며 D_{wire}은 AR과 CN 등 유선 채널의 패킷 전송 지연 시간이다. D_{L2}는 L2 핸드오버에 필요한 시간이며 D_{init}는 FMIPv6에서 FBU 및 HI 메시지 교환에 필요한 시간이다. RTT_{MN-oAR}은 MN과 oAR 사이의 패킷 왕복 전송 지연 시간이며 RTT_{AR}은 oAR과 nAR 사이의 패킷 왕복 전송 지연 시간이다. D_{RD} 와 D_{DAD} 는 각각 라우터 발견 과 DAD 과정에 필요한 시간이다. 이는 rtAdvInterval/2 과 RetransTimer 의 기본값을 사용한다 [4][5]. D_{CR}과 D_{PBU}는 각각 CR 과 PBU 동작이 수행되는데 걸리는 시간이다. 이때 RTT_{MN-HA} 및 RTT_{CN-HA} 은 D_{wire}와 같고 RTT_{MN-CN} 및 RTT_{nAR-HA}, RTT_{nAR-CN} 은 $D_{wire}+D_{wireless}$와 같다.

표 1의 시간을 사용하여 세 프로토콜 MIPv6, FMIPv6, FMIPv6_PBU의 지연 시간을 구하면 다음과 같다:

$$MIPv6 \quad = D_{L2} + D_{RD} + D_{DAD} + D_{CR}$$
$$FMIPv6 \quad = D_{init} + D_{L2} + D_{UNA} + D_{CR}$$
$$FMIPv6_{PBU} = D_{init} + D_{PBU} + D_{UNA}$$

FMIPv6-PBU는 L2 핸드오버가 수행되기 전에 시작되어 MN의 개입없이 수행되므로 MN의 핸드오버 지연 시간은 L2 핸드오버 시간과 PBU 교환 시간에 영향을 받는다. 일반적으로 L2 핸드오버 시간은 D_{PBU}보다 매우 크다.



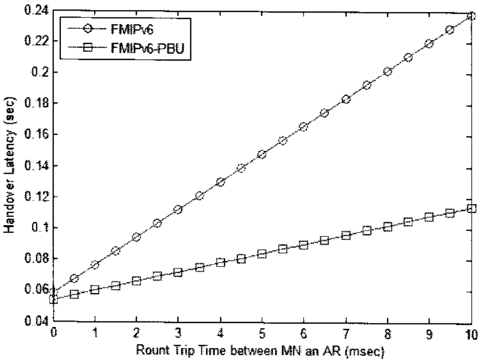


그림 5. MIPv6, FMIPv6, FMIPv6-PBU 프로토콜의 핸드오버 지연시간 비교
Fig. 5. Handover latency comparison of Handover protocols

그림 5 는 각 프로토콜의 핸드오버 지연 시간을 나타낸다. 제안한 프로토콜은 MIPv6와 FMIPv6의 핸드오버 과정에 필요한 CR 과정을 미리 수행함으로써 CR로 인한 핸드오버 지연을 없앴다. CR은 최소 3회의 네트워크 왕복 지연시간을 발생시키므로 MN과 nAR사이의 채널 상황에 따라 매우 큰 차이를 보임을 알 수 있다. 또한 MN이 nAR로 이동한 직후에 CN과 즉시 직접적으로 통신할 수 있으므로 터널링으로 인한 패킷의 지연 시간을 줄일 수 있다.

V. 결론

본 논문에서는 MIPv6와 FMIPv6의 핸드오버의 지연 요소를 분석하였다. FMIPv6는 MIPv6 핸드오버 과정에서 긴 시간이 필요한 DAD 과정을 핸드오버 준비 과정에서 수행함으로써 핸드오버 준비 시간을 줄였다. 그러나 FMIPv6는 MIPv6와 마찬가지로 CR 과정을 수행해야 하므로 MN의 상향 및 하향 링크 지연 시간이 필요하다. 이 지연 시간을 없애기 위해 본 논문에서는 CR을 미리 수행하는 FMIPv6-PBU 핸드오버 프로토콜을 제안하였고 이 프로토콜의 동작을 상세히 기술하였다. 또한 FMIPv6-PBU의 성능을 MIPv6와 FMIPv6와 정량적으로 비교하여 기존 핸드오버 프로토콜 보다 우수함을 보였다.

참고문헌

[1] L. Dimopoulou, G. Leoleis, I.O. Venieris, "Fast Handover Support in a WLAN Environment: Challenges and Perspectives", Vol 19, Issue 3, IEEE Network, May/June 2005.

[2] D. Jhonson, C. Perkins, J. Arkko, "Mobility Support in IPv6", IETF RTC 3775, June. 2004

[3] R. Koodli, Ed, "Mobile IPv6 Fast Handovers", IETF RFC 5268, June. 2008

[4] S. Thomson, T. Narten, T. Jinmei, "IPv6 Stateless Address Autoconfiguration", IETF RFC 4862, Sep. 2007

[5] T. Narten, E. Normark, W. Simpson, H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", IETF RFC 4861, Sep. 2007

[6] Y. Y. An, B. H. Yae, K. W. Lee, Y. Z. Cho, W. Y. Jung, "Reduction of Handover Latency Using MIH Services in MIPv6," aina, pp. 229-234, 20th International Conference on Advanced Information Networking and

Applications - Volume 2 (AINA'06), 2006
[7] H. V. Nguyen, S. Ro, J. Ryu, Y-G. Hong, "A Temporary Binding Update in Fast Handover for Mobile IPv6", International Conference on Convergence Information Technology 2007, Nov. 2007.

저자소개

장종민(Jong Min Jang)

2006년 동의대학교 정보통신공학과 공학사
2006년~현재 동의대학교 대학원 정보통신공학과 석사과정
※관심분야 : MAC, 네트워크 성능 분석

이유태(Yutae Lee)

1994년 한국과학기술원 수학과 이학석사
1997년 한국과학기술원 수학과 이학박사
2001년~현재 동의대학교 정보통신공학과 부교수
※관심분야 : Opportunistic Spectrum Access

이 숭 희(Soong-hee Lee)

1987년 경북대 전자공학과 공학사
1990년 경북대 전자공학과 공학석사
1995년 경북대 전자공학과 공학박사
1987 ~ 1997 한국전자통신연구원 선임연구원
1997 ~ 현재 인제대학교 정보통신공학과 부교수
※관심분야 : NGN, MANET, AAA, OBS

김동일(Dong-il Kim)

1983년 광운대학교 전자통신공학과 공학석사
1992년 광운대학교 전자통신공학과 공학박사
1991년~현재 동의대학교 정보통신공학과 교수
※관심분야 : 통신망 성능분석, 무선망 프로토콜, IT네트워크 기술 표준화

최삼길(Sam-gil Choi)

1969년 광운대학교 무선통신공학과 공학사
1988년 동아대학교 전자공학과 공학석사
2001년 동아대학교 전자공학과 공학박사
1973년 ~ 1985년 동의과학대학 통신공학과 교수
1985년 ~ 현재 동의대학교 정보통신공학과 교수
※관심분야 : 디지털 신호처리, 네트워크 프로토콜