

Proxy Mobile IPv6의 핸드오버 성능 평가

현승일¹, 한연희¹, 공기식², 신명기³, 정상진³

¹한국기술교육대학교 인터넷미디어공학부

{freeprison, yhhan}@kut.ac.kr

²고려대학교 컴퓨터학과

computer@korea.ac.kr

³한국전자통신연구원 표준연구센터

{mkshin, sijeong}@etri.re.kr

Handover Performance Evaluation for Proxy Mobile IPv6

Seung-Il Hyeon¹, Youn-Hee Han¹, Ki-Sik Kong², Myung-Ki Shin³, Sangjin Jeong³

¹Korea University of Technology and Education, School of Internet-Media

²Korea University, Dept. of Computer Science and Engineering

³Protocol Engineering Center, ETRI

요 약

IPv6 이동성 관리를 위한 기존의 호스트 기반 프로토콜인 Mobile IPv6는 오랜 기간 동안 많은 연구에 의해 표준화되어진 안정적인 기술이지만 실제 상용망에서는 잘 사용이 안되고 있다. 그 이유는 단말의 수정 요구, 과도한 유/무선 자원 사용, 느린 핸드오버 지연시간 등이 있다. 이와 같은 Mobile IPv6의 단점을 극복하기 위하여 개발된 네트워크 기반 이동성 관리 프로토콜인 Proxy Mobile IPv6가 최근에 주목받고 있다. Proxy Mobile IPv6는 단말에서의 이동성 지원을 위한 수정이 필요하지 않으며, 유/무선 자원도 효율적으로 사용하며, 핸드오버 지연시간도 단축된다는 장점을 지니고 있다. 본 논문에서는 Proxy Mobile IPv6와 기존 호스트 기반 프로토콜인 Mobile IPv6, Hierarchical Mobile IPv6를 체계적으로 분석하고 핸드오버 지연시간을 비교 분석한다. 비교 분석 결과로서 Proxy Mobile IPv6는 기존 기법들보다 약 84~93% 정도의 핸드오버 지연시간 단축이 있음을 알 수 있다.

1. 서 론

최근의 통신 사업은 유선 네트워크의 시장 포화와 함께 무선 네트워크의 시장이 급속도로 증가하고 있다. 특히 최근에는 휴대폰, PDA (Personal Digital Assistant), 노트북 등 무선 네트워크를 지원하는 다양한 휴대용 기기들이 널리 보급되고 있다. 무선 네트워크를 통한 서비스는 보통 동영상이나 화상 전화와 같은 실시간 광대역 서비스가 주를 이루고 있으며 한정된 자원의 무선 네트워크상에서 이를 끊임없이 원활하게 서비스 할 수 있는 기술이 필요하다.

차세대 이동 통신 기술이 가져야 할 가장 중요한 기능 중 하나는 이동성 관리 (Mobility Management)이다. 특히 IP 서비스 관점에서 이동성 관리란 모바일 단말이 임의의 IP 네트워크 접속 지점 (Access Point, 이하 AP)에 붙었을 때 단말에게 향하는 패킷을 올바르게 전달하고, 단말이 다른 IP 네트워크의 AP로 옮길 때 패킷 지연을 최소화하여 연결이 지속되도록 하는 기능이다. 이러한 IP 이동성 관리를 수행하기 위하여 가장 널리 알려진 기술은 IETF (Internet Engineering Task Force)에서 표준으로 제정한 Mobile IPv6 (MIPv6)이다 [1].

MIPv6는 오랜 기간 동안 많은 연구에 의해 표준화된 안정된 기술이다. 그럼에도 불구하고 지난 몇 년 동안 실제 상용망에서 많이 활용되지 못한 이유는 대략 다음과 같다 [2].

- 제한된 자원의 모바일 단말기에 구현하기에는 기술이 너무 방대하다.
- MIPv6의 많은 메시지 교환에 따른 단말기의 배터리 부담이 크다.
- 많은 메시지 교환 때문에 네트워크 자원에 부하가 생긴다.
- 모바일 단말기 내에 MIPv6 프로토콜을 가지고 있어야 하지만 상용 임베디드 운영체제들에 구현이 되어 있지 않고 향후 구현 계획 또한 불확실하다.

이러한 MIPv6의 단점들로 인해 FMIPv6 (Fast Handover for Mobile IPv6)[3]와 HMIPv6 (Hierarchical Mobile IPv6)[4]과 같은 개선된 IPv6 이동성 프로토콜들이 개발되었다. 이러한 프로토콜들은 핸드오버 지연 단축과 효율적인 네트워크 자원 활용 등의 목적으로 MIPv6보다 더 나은 이동성을 제공한다. 하지만, 여전히 이동 단말 (Mobile Node, 이하 MN)에 이동성 관련 스택 변화를 요구하고 있으며 과도한 무선 자원 낭비의 단점을 극복하지 못하고 있다.

그래서, IETF NETLMM Working Group은 최근에 Proxy Mobile IPv6 (PMIPv6) [5]라 불리는 네트워크 기반 이동성 관리 프로토콜 표준을 제정하기 시작하였으며, PMIPv6는 MIPv6와 달리 실제 상용망에 바로 적용

하여 사용되는 것을 목표로 만들어진 프로토콜이다. 그래서, PMIPv6는 표준으로 제정되기 이전인 현재 시점에도 많은 이동 통신 사업자 및 관련 업계에서 주목을 받고 있다.

PMIPv6의 가장 큰 장점은 MN에 이동성을 위한 어떠한 수정도 필요치 않다는 것이다. 즉, MN이 IPv6 프로토콜만 가지고 있다면 핸드오버에 필요한 시그널링에 전혀 관계할 필요 없이 이동성 서비스를 받을 수 있다. 게다가, 이동성 처리 시그널링 메시지에 의한 유/무선 네트워크 부하가 줄어드는 등 많은 장점을 지닌다.

따라서 본 논문에서는 가장 최근에 주목받고 있는 네트워크 기반의 PMIPv6와 호스트 기반의 기존 IP 이동성 프로토콜들을 체계적으로 분석하고 핸드오버 지연시간 측면에서의 성능을 비교 평가한다. 그래서, 네트워크 기반 프로토콜인 PMIPv6가 상용망에 적용될 때 얻을 수 있는 잇점을 정량적으로 보여준다.

본 논문의 순서는 다음과 같다. 우선 2장에서는 호스트 기반 및 네트워크 기반 프로토콜들에 대한 기본 동작 원리를 소개한다. 3장에서는 위의 기술들에 대한 핸드오버 지연시간 분석과 이의 성능평가를 위한 시스템 모델 도출 및 수치적인 비교 성능 평가를 할 것이다. 4장에서는 3장의 성능 평가 분석에 기반한 결론을 내린다.

2. 관련연구

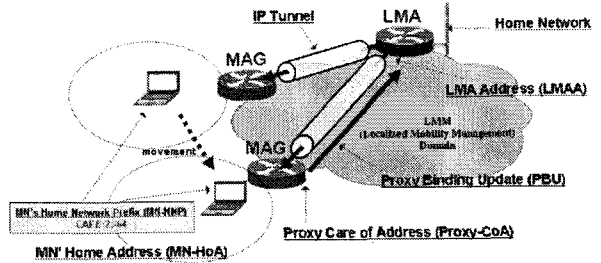
2.1 MIPv6

Mobile IPv6는 가장 널리 알려진 IP 이동성 프로토콜로써 이원화된 주소 체계를 통해 MN의 고유 주소인 HoA (Home Address)와 이동한 네트워크에서 생성한 새로운 주소 CoA (Care-of Address)를 HA (Home Agent)에게 바인딩 함으로써 이동성을 지원한다. 특히, 통신 대상 노드 CN (Correspondent Node)이 MIPv6 프로토콜을 수용하고 있다면 CN에게도 그러한 바인딩을 보내어서 데이터 패킷의 최적화된 라우팅 경로를 지원할 수 있는 장점을 지닌다.

그러나 이런 시그널링으로 인해 MN이 빈번히 이동할 경우 과도한 네트워크 자원 소모와 함께 바인딩 지연에 따른 핸드오버 지연 증가의 문제가 생긴다. 결국 MIPv6는 단지 MN의 위치 등록 및 현재 통신 중인 세션의 데이터 패킷들에 대한 경로 재설정에 관련된 프로토콜로 VoIP (Voice over IP)와 같은 실시간 통신을 만족시킬 수준의 이동성을 지원하기에는 여러 문제점이 존재한다. 또한, 많은 수정을 요구하는 MIPv6를 MN에 구현해야 하기 때문에 현실에 바로 적용시키기에는 무리가 있어 근래까지도 상용망에 효과적으로 활용하지 못하고 있다.

2.2 HMIPv6

MIPv6에서의 MN의 위치 등록을 위한 과도한 시그널링은 MN의 공간적 지역성 (Spatial Locality)을 고려함으로써 완화될 수 있다. HMIPv6는 도메인 네트워크를 관리하는 MAP (Mobile Anchor Point)이라는 이동성 관리 에이전트를 도메인 내에 둔다. 도메인은 임의의 ISP 망이 될 수도 있고 캠퍼스 또는 회사 네트워크 망이 될 수



[그림 1] PMIPv6의 시스템 구성도

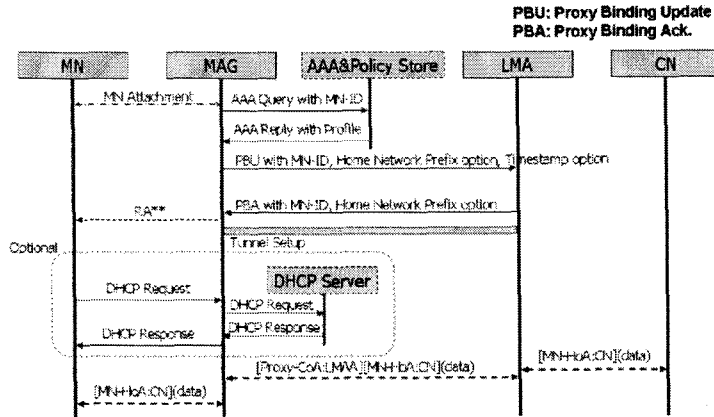
도 있다. MAP은 해당 도메인에서 MN의 HA와 같은 역할을 수행한다. 새로운 네트워크에 들어간 MN은 두 개의 CoA를 할당받는다. 하나는 연결된 서브넷 위치를 나타내는 LCoA (Local CoA)이고 다른 하나는 MN이 위치한 도메인 네트워크를 나타내는 RCoA (Regional CoA)이다. RCoA는 동일 MAP 도메인 내에서는 변하지 않는다. 만일 MN이 동일 MAP 도메인 내에서 이동한다면, MN의 MAP에게 자신의 변한 위치 정보를 갱신하게 된다. 이때, MN의 RCoA는 변하지 않으므로 도메인 외부에 존재하는 CN이나 HA에는 위치 정보를 수정할 필요가 없다. 즉, 동일 MAP내에서는 MN의 위치에 대한 투명성을 CN과 HA에게 제공할 수 있으며, MN이 이동시마다 HA와 CN들에게 갱신하지 않으므로 위치 등록을 위한 시그널링 비용을 줄일 수 있다. 또한, MAP의 도입으로 인하여 지역성을 갖는 MN에 대해서 MIPv6보다 빠른 핸드오버를 제공할 수 있다.

2.3 PMIPv6

MIPv6와 HMIPv6가 MN의 수정을 요구하는 호스트 기반 이동성 관리 프로토콜들이라면 PMIPv6는 네트워크 측 장비의 수정만을 요구하는 네트워크 기반 이동성 관리 프로토콜이다. PMIPv6는 기본적으로 MIPv6의 개념을 확장하였으나 네트워크 기반이라는 점에서 많은 차이가 존재한다. 현재 PMIPv6는 도메인 네트워크 내부에서만 동작하는 조건으로 설계되어 있다. 즉, 도메인 간 이동성 관리는 현재 설계되어 있지 않기 때문에 이후 본 논문에서의 성능평가는 도메인 내부의 MN 이동만을 대상으로 수행할 것이다.

PMIPv6의 시스템 구성도는 [그림 1]과 같다. PMIPv6는 MAG (Mobile Access Gateway)와 LMA (Local Mobility Anchor)라 불리는 새로운 기능 요소를 필요로 한다. MAG는 보통 AR (Access Router)에 그 기능이 올라가며 MAG의 가장 중요한 역할은 MN을 대신하여 이동성 지원 시그널링을 수행하는 것이다. LMA의 역할은 이 시그널링을 받아 MN의 이동성을 관리하는 것이다. LMA는 보통 도메인의 게이트웨이 또는 그 근처에 배치된다. PMIPv6의 기본 동작은 다음 [그림 2]와 같다.

MN은 Link 계층상의 핸드오버가 종료되면 자신이 이동했음을 알리기 위해 MN-Identifier를 새로운 MAG에게 보내 인증(authentication)을 거친다. 이때 MAG는



[그림 2] PMIPv6의 메시지 흐름도

AAA (Authentication Authorization Accounting) 서버를 이용하여 인증을 확인하고 MN의 프로파일을 획득한다. 이 프로파일 안에는 LMA 주소, 주소 구성 모드 등의 정보가 포함되어 있다.

프로파일을 획득한 MAG는 MN의 위치등록을 위하여 LMA에게 PBU (Proxy Binding Update) 메시지를 보낸다. LMA는 PBU를 받고서 이것이 올바른 MAG에게서 온 것인지 AAA서버를 통해 확인하고 확인이 되었다면 위치등록 메시지를 저장하고 PBA (Proxy Binding Acknowledgement)를 MAG에게 보내준다. MAG가 PBA를 받으면 MAG와 LMA간에 데이터 전송을 위한 터널이 생기고 이를 통해 MN의 데이터 전송 또한 가능해진다. 이후, 도메인 외부에서 도메인 내부의 MN에게 보내어지는 패킷은 LMA가 받게 된다. 그 패킷들은 이후 MAG로 터널링을 통하여 MN에게 보내어진다. 한편, MN이 도메인 외부로 보내는 모든 패킷은 MAG에서 터널링을 하여 LMA로 전달되며 LMA에서 터널을 벗겨 내어 외부로 보내게 된다.

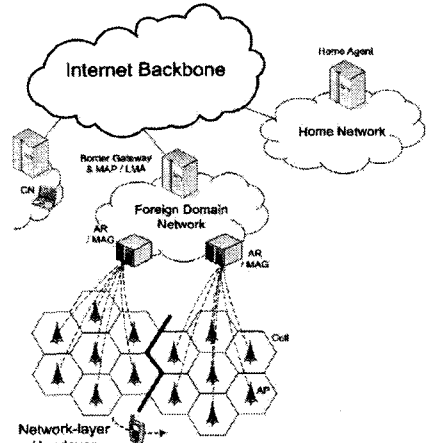
주목할 점은 MN은 새로운 MAG로 이동하여도 매번 같은 home network prefix를 할당받기 때문에 MN은 항상 홈 네트워크에 있는 것으로 착각하여 핸드오프 시그널을 발생시키지 않는다는 점이다. 즉, 도메인에 처음 들어올 때 LMA에 의하여 할당된 임의의 MN에게 할당된 home network prefix는 그 LMA에 의하여 계속해서 관리가 된다. 이후 MN이 도메인 내부의 어떠한 AR에 접속되어도 Router Advertisement (RA) 메시지를 받을 때 같은 home network prefix를 받게된다.

3. 시스템 모델 및 성능 평가

3.1 시스템 모델

MIPv6와 HMIPv6 그리고 PMIPv6의 성능 평가를 위한 시스템 모델은 [그림 3]과 같다. PMIPv6와 HMIPv6는 지역적인 이동성 관리를 위한 프로토콜이기 때문에 성능 비교는 도메인 내부(Intra-Domain)에서의 이동만을 대상으로 한정한다. 도메인 외부의 인터넷 백본망에 대

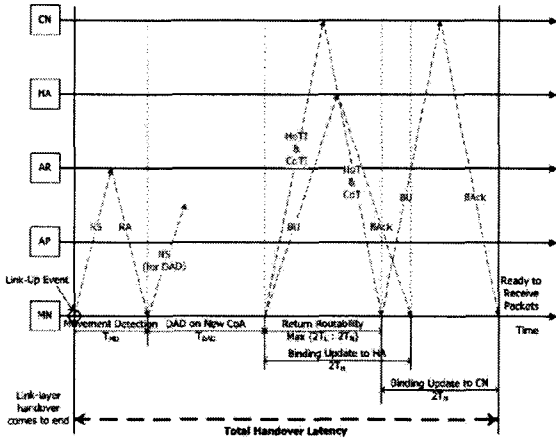
한 게이트웨이 역할을 하는 LMA 또는 MAP는 개별적인 네트워크를 형성한다. 비슷하게 HA와 CN이 속하는 네트워크의 게이트웨이 또한 인터넷 백본망에 연결되어 LMA/MAP과 통신이 가능하다. LMA/MAP 도메인 네트



[그림 3] 성능 평가를 위해 가장한 시스템 모델

[표 1] 성능 평가를 위한 파라미터

기호	설명
t_{α}	MN과 AP간의 패킷 전송 시간
t_{β}	AP와 AR/MAG간의 패킷 전송 시간
t_{γ}	AR/MAG와 MAP/LMA간의 패킷 전송 시간
t_{δ}	AR/MAG와 HA간의 패킷 전송 시간
t_{ϵ}	AR/MAG와 CN간의 패킷 전송 시간
t_{η}	HA와 CN간의 패킷 전송 시간
T_{MD}	L3 단계에서의 IP 이동 감지 소요 시간
T_{DAD}	L3 단계에서의 DAD 소요 시간
T_R	MN <-> AR / MAG = $t_{\alpha} + t_{\beta}$
T_M	MN <-> MAP / LMA = $t_{\alpha} + t_{\beta} + t_{\gamma}$
T_N	MN <-> CN = $t_{\alpha} + t_{\beta} + t_{\epsilon} + t_{\epsilon}$
T_L	MN <-> CN via HA = $t_{\alpha} + t_{\beta} + t_{\gamma} + t_{\delta} + t_{\eta}$
T_H	MN <-> HA = $t_{\alpha} + t_{\beta} + t_{\delta}$



[그림 4] MIPv6의 핸드오버 지연시간

워크 내부에 AR/MAG가 존재하고 AR/MAG는 여러 AP와 연결되어 있으며 그 중 하나의 AP가 MN과 연결된다. 단말기가 서로 다른 AR/MAG에 연결된 AP로 이동하였을 경우 IP 핸드오버가 발생하며 MIPv6, HMIPv6, 또는 PMIPv6 과 같은 IP 이동성 관리 프로토콜이 동작하게 된다.

[표 1]은 각 프로토콜에서 핸드오버 지연시간 분석 시에 사용될 파라미터들이다. 핸드오버 지연 원인으로는 메시지를 생성하기 위한 프로세싱 시간, IP스택에 대한 Encapsulation과 Decapsulation 시간, 큐잉 지연 등등 다양한 인자들이 있다. 하지만, 그러한 인자들은 전체 지연시간에서 차지하는 비중이 매우 작기 때문에 주로 네트워크 상의 패킷 전송 시간을 주요 인자 (Dominant Factor)로 결정한다. 따라서, [표 1]에 제시한 각 파라미터는 [그림 3]에서 소개한 네트워크 모델을 이루는 각 개체들 간의 평균 패킷전송시간과 연관된다.

3.2 핸드오버 지연시간 비교

본 논문에서는 이동성 프로토콜에 대한 가장 중요한 성능 지표인 핸드오버 지연시간을 분석하여 각 프로토콜에 대한 평가를 하였다. 이동성 프로토콜에 대한 핸드오버 지연시간의 기준을 정의하기 위해 본 논문에서는 그 시기를 Link 계층 단계에서 핸드오버가 끝난 시점부터 첫 번째 패킷을 받을 수 있는 준비가 완료되는 시점까지를 이동성 프로토콜의 핸드오버 지연시간이라고 정의한다.

[그림 4]는 MIPv6의 핸드오버 프로시저의 타이밍 다이어그램 (Timing Diagram) 이다. MN은 Link 계층에서의 핸드오버가 종료되면 이동성 관리를 위해 Network 계층에서의 핸드오버를 시작한다. 본 논문에서는 MIPv6 핸드오버 시간을 단축하기 위하여 단말이 Link-Up Event를 받으면 바로 RS (Router Solicitation) 메시지를 보낸다고 가정한다. MN이 AR에게 RS 메시지를 보내면 AR은 MN에게 RA(Router Advertisement) 메시지를 보내어 응답한다. MN이 RA를 받게 되면 새로운 IP 네트워크로 이동했음을 알 수 있으며, 이 과정을 위하여

소요되는 시간을 T_{MD} 라고 표현하고 그 시간은 $T_{MD} = 2T_R$ 이다.

이동을 감지한 이후 MN은 새로운 CoA를 만들고 동일 주소 존재여부 검사인 DAD (Duplicate Address Detection) [6] 과정을 위하여 필요한 시간은 $T_{DAD} = R * D$ 로 표현하며, 여기서 R과 D는 각각 NS(Neighbor Solicitation) 메시지에 대한 중복을 알리는 응답인 NA(Neighbor Advertisement) 메시지를 기다리는 시간과 중복주소 검사에 대한 시도 횟수이다. 본 논문에서는 R과 D를 RFC 2462 [6] 에서의 기본값인 1000ms와 1의 값을 주는 것으로 가정한다. 따라서, $T_{DAD} = 1000ms$ 가 된다.

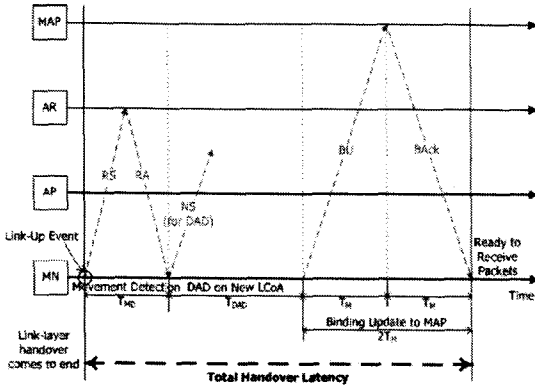
DAD과정까지 끝났다면 Registration 과정이 남는다. 그 과정은 최종적으로 CN에게 바뀐 CoA에 대한 바인딩 업데이트를 하는 것이며, 이 과정으로서 모든 핸드오버가 종료된다. HA로 바인딩 업데이트는 꼭 수행해야 하지만 CN에게 Route Optimization이 이미 수행되는 상황을 가정하기 때문에 전체 핸드오버 지연시간에는 영향을 미치지 않는다. 그리고 Route Optimization을 위한 Return Routability를 위한 시간은 $2T_L$ 이라고 가정하는데 그 이유는 대개의 경우 HA를 거쳐 CN과 통신하는 $2T_L$ 이 일반적으로 더 크기 때문이다. 따라서 MIPv6의 핸드오버 지연시간의 수식을 세워보면 다음과 같다.

$$\begin{aligned}
 T^{MIP} &= T_{MD} + T_{DAD} + 2T_L + 2T_N \\
 &= 2(t_\alpha + t_\beta) + T_{DAD} + 2(t_\alpha + t_\beta + t_\gamma + t_\delta + t_\eta) \quad (\text{식 1}) \\
 &\quad + 2(t_\alpha + t_\beta + t_\gamma + t_\epsilon) \\
 &= T_{DAD} + 6(t_\alpha + t_\beta) + 4t_\gamma + 2(t_\delta + t_\epsilon + t_\eta)
 \end{aligned}$$

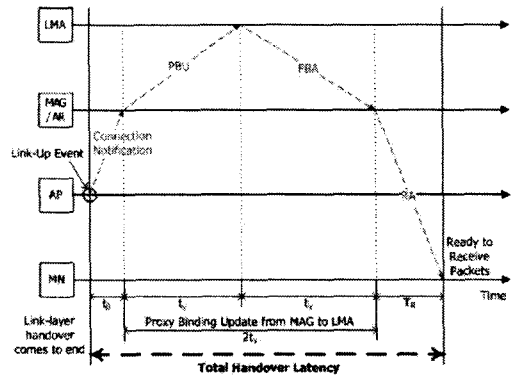
[그림 5]는 HMIPv6에 대한 핸드오버 프로시저의 타이밍 다이어그램이다. HMIPv6의 한정된 지역 내에서의 이동성 관리의 특성상 MIPv6보다 전체 시그널링 비용이 감소됨을 알 수 있다. MN의 이동은 도메인 내부 (Intra-domain)로 가정하였기 때문에 HMIPv6의 도메인 내부 이동에 대한 지연시간만을 핸드오버 지연시간으로 정한다. 전체 핸드오버 절차에 대해 DAD과정까지는 MIPv6와 다르지 않으며 이후 MAP에 Registration하는 과정만 MIPv6과 다르다. HMIPv6의 핸드오버 지연시간 수식은 다음과 같다.

$$\begin{aligned}
 T^{HMIP} &= T_{MD} + T_{DAD} + 2T_M \\
 &= 2(t_\alpha + t_\beta) + T_{DAD} + 2(t_\alpha + t_\beta + t_\gamma) \quad (\text{식 2}) \\
 &= T_{DAD} + 4(t_\alpha + t_\beta) + 2t_\gamma
 \end{aligned}$$

[그림 6]은 PMIPv6 핸드오버 프로시저의 타이밍 다이어그램이다. MIPv6나 HMIPv6와 다른 점은 크게 두 가지로 하나는 이동감지와 DAD과정이 프로토콜 특성상 생략된다는 점과 다른 하나는 Link 계층 핸드오버 종료 후 이동성 관리를 위한 Registration 절차가 MAG에서 시작된다는 것이다. AP가 MN의 연결여부를 MAG에게 통보함으로써 이동감지가 되고 PMIPv6 프로토콜 특성상 단말은 이동을 해도 항상 Home Network으로서 인식하



[그림 5] HMIPv6의 핸드오버 지연시간



[그림 6] PMIPv6의 핸드오버 지연시간

게 되며 자신이 사용하던 주소를 계속해서 사용하게 되므로 DAD과정이 필요치 않게 되었다. MN은 자신이 IP 핸드오버를 했는지에 대한 판단을 내리지 못하며 어떠한 이동성 관련 시그널링을 발생시키지 않으며 관련 시그널링은 MAG와 LMA간의 통신에 의해서만 발생된다. 단지 MN은 새로운 AR/MAG에서 전달해주는 RA 메시지를 받고 자신의 Default Gateway만 적절히 새롭게 설정만 하면 IP 핸드오버 절차가 끝나게 된다. 따라서 PMIPv6의 핸드오버 지연시간 수식은 다음과 같이 간단하게 표현할 수 있다.

$$\begin{aligned}
 T^{PMIP} &= t_{\beta} + 2t_{\gamma} + T_R \\
 &= t_{\beta} + 2t_{\gamma} + t_{\alpha} + t_{\beta} \quad (\text{식 3}) \\
 &= 2(t_{\beta} + t_{\gamma}) + t_{\alpha}
 \end{aligned}$$

3.3 성능 비교

본 절에서는 주어진 시스템 구성 모델에 대해 도메인 내부에서의 MIPv6, HMIPv6, PMIPv6의 핸드오버 지연시간을 비교 평가하였다. 우선 [표 1]에서 제시한 각 파라미터들에 대한 기본 가정 값으로서 $t_{\alpha} = 50ms$, $t_{\beta} = 10ms$, $t_{\gamma} = 50ms$, $t_{\delta} = 100ms$, $t_{\epsilon} = 70ms$, $t_{\eta} = 70ms$ 을 설정한다.

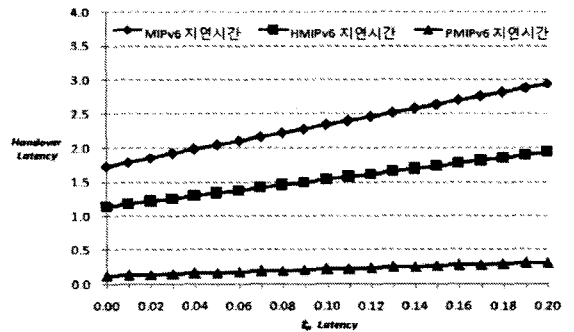
먼저 무선으로 이루어진 MN과 AP간 구간인 t_{α} 인자의 변화에 따른 성능 비교를 보여주면 [표 2]와 [그림 7]과

[표 2] t_{α} 의 변화에 따른 핸드오버 지연시간

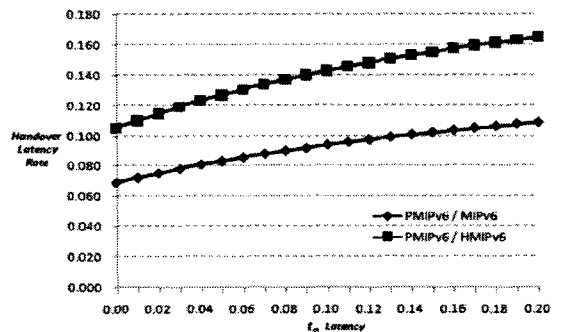
t_{α} Latency	MIPv6	HMIPv6	PMIPv6
0.000	1.54	1.14	0.12
0.010	1.6	1.18	0.13
0.020	1.66	1.22	0.14
0.030	1.72	1.26	0.15
0.040	1.78	1.30	0.16
0.050	1.84	1.34	0.17
0.060	1.9	1.38	0.18
0.070	1.96	1.42	0.19
0.080	2.02	1.46	0.20
0.090	2.08	1.50	0.21
0.100	2.14	1.54	0.22

같다. [표 2]에서 알 수 있듯이 각 프로토콜의 핸드오버 지연시간은 $MIPv6 > HMIPv6 > PMIPv6$ 라는 관계를 보이며, PMIPv6는 기존의 MIPv6과 HMIPv6보다 약 85~90% 정도의 핸드오버 지연시간 단축이 있음을 알 수 있다. 즉, PMIPv6의 핸드오버 지연시간 측면에서의 성능은 매우 우수함을 알 수 있다.

또한, [그림 7]에서 알 수 있듯이 MIPv6와 HMIPv6의 경우 무선 구간에서의 시그널링이 많이 존재함으로써 t_{α} 인자에 대한 영향을 크게 받는다. 반면, PMIPv6의 경우 핸드오버 절차 중 무선 구간에서의 시그널링이 RA 메시지 밖에 없으므로 t_{α} 인자에 큰 영향을 받지 않는다.



[그림 7] t_{α} 의 변화에 따른 핸드오버 지연시간 그래프



[그림 8] t_{α} 의 변화에 따른 PMIPv6와 타 프로토콜 간 핸드오버 지연 시간 비율을 나타낸 그래프

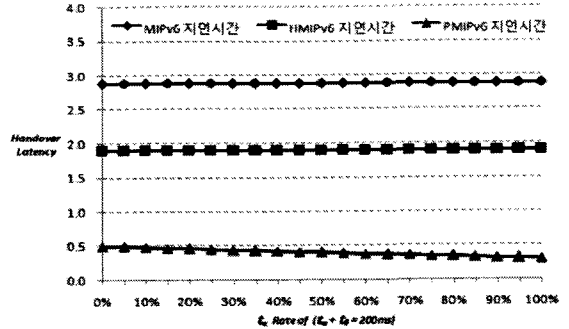
[표 3] t_α 와 t_β 의 비율에 따른 PMIPv6의 핸드오버 지연시간
단위 : 초

t_α 의 비율	t_α	t_β	PMIPv6
0%	0.000	0.200	0.5
5%	0.010	0.190	0.49
10%	0.020	0.180	0.48
15%	0.030	0.170	0.47
20%	0.040	0.160	0.46
25%	0.050	0.150	0.45
30%	0.060	0.140	0.44
35%	0.070	0.130	0.43
40%	0.080	0.120	0.42
45%	0.090	0.110	0.41
50%	0.100	0.100	0.4
55%	0.110	0.090	0.39
60%	0.120	0.080	0.38
65%	0.130	0.070	0.37
70%	0.140	0.060	0.36
75%	0.150	0.050	0.35
80%	0.160	0.040	0.34
85%	0.170	0.030	0.33
90%	0.180	0.020	0.32
95%	0.190	0.010	0.31
100%	0.200	0.000	0.3

상대적으로 안정적인 유선 네트워크에 비해 무선 네트워크는 환경과 상황에 따라 불안정함을 고려할 때, PMIPv6이 무선 네트워크의 영향을 덜 받는다는 점은 PMIPv6이 다른 프로토콜보다 더 안정적인 서비스를 제공해 줄 수 있다는 점을 시사한다.

[그림 8]은 t_α 인자의 변화에 따른 MIPv6와 HMIPv6의 핸드오버 지연시간에 대한 PMIPv6 핸드오버 지연시간 비율이다. 그림이 나타내듯이 $t_\alpha = 0.02$ 즉, 20ms일 때 MIPv6에 대한 PMIPv6의 비율은 0.075이다. 이는 PMIPv6가 MIPv6의 핸드오버 지연시간을 약 92.5% 정도 단축시킴을 의미한다. 반면, 동일 조건에서 HMIPv6의 핸드오버 지연시간을 약 88% 정도 단축시킴을 알 수 있다. 반면 흥미로운 사실은 [그림 8]에서 무선 구간의 t_α 지연시간이 늘어날수록 MIPv6와 HMIPv6 대비 PMIPv6의 성능차이가 줄어들다는 점이다. 그 원인은 PMIPv6의 경우 전체 핸드오버 절차에서 무선 구간의 시그널링 절차가 차지하는 비율이 다른 프로토콜보다 크기 때문이다. 즉, PMIPv6이 다른 프로토콜보다 무선 구간의 변화에 좀 더 민감하게 반응한다는 점을 알 수 있다. 하지만 다른 프로토콜들에 비해 PMIPv6의 핸드오버 지연시간이 매우 작아서 MIPv6와 HMIPv6 대비 PMIPv6의 성능차이가 줄어들다고 해도 여전히 PMIPv6가 절대적으로 우월한 성능을 보여준다.

[표 3]과 [그림 9]는 단말과 AP 사이의 구간과 AP와 AR/MAG 사이 구간 각각이 각 프로토콜의 핸드오버 지연시간에 미치는 영향을 보여준다. [표 3]에서 보이듯이 이번 실험에서는 t_α 와 t_β 의 지연시간의 합을 200ms로 일정하게 유지하였다. 다만 두 인자의 비율이 대칭적으로 증가/감소하도록 변화를 주었으며, 결과적으로 [그림 9]와 같은 그래프를 보여준다. 이 그림에서 보이듯이 MIPv6와 HMIPv6와는 달리 PMIPv6의 경우 t_α 의 비율이 늘어날수록 더욱 좋은 성능향상을 보이고 있다. 이는



[그림 9] t_α 와 t_β 의 비율에 따른 PMIPv6의 핸드오버 지연시간 그래프

달리 말하면 t_β 에서의 지연시간이 작을수록 전체 성능을 좋게 유지한다는 말이며, t_α 보다는 t_β 가 PMIPv6의 핸드오버 지연시간을 결정하는 더 중요한 파라미터임을 알려준다. 그 이유는 PMIPv6에서 AP와 AR/MAG사이의 시그널링이 전체 핸드오버 절차에 많은 부분을 차지하기 때문이다.

4. 결론

본 논문에서는 최근 IP 이동성 관리 프로토콜에서 주목받고 있는 네트워크 기반의 PMIPv6를 기존의 호스트 기반 IP 이동성 관리 프로토콜들인 MIPv6와 HMIPv6와 성능 비교 평가를 하였다. 본 논문의 분석과 실험을 통하여 확인한 사실은 PMIPv6는 단말기의 수정 불필요, 유/무선 네트워크 시그널링 비용 감소, 짧은 핸드오버 지연시간과 같은 많은 장점을 지니고 있어 실제 상용망에 적용하기 좋은 프로토콜임을 알 수 있었다. 특히 PMIPv6는 HMIPv6과 MIPv6보다 84%~93%정도의 핸드오버 지연 시간 단축이 있음을 알 수 있었다.

참고문헌

- [1] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6", RFC 3775, June 2004
- [2] Proxy Mobile IPv6, KRNet (Korea Internet Conference) 2007, 2007년 6월 29일
- [3] R. Koodli (Ed.), "Fast Handovers for Mobile IPv6", IETF RFC 4068, July 2005.
- [4] H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)," IETF RFC 4140, Aug. 2005.
- [5] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, B. Patil, "Proxy Mobile IPv6," draft-ietf-netlmm-proxymp6-01.txt, June 18, 2007.
- [6] S. Thomson and T. Nartan, "IPv6 Stateless Address Autoconfiguration," IETF RFC 2462, Dec. 1998.