

## 이동노드의 속도를 고려한 선택적 MAP 등록 기법

이우엽 조인휘<sup>o</sup>

한양대학교 전자컴퓨터통신공학과

matias12@hanyang.ac.kr, iwjoe@hanyang.ac.kr

## A Selective MAP Registration Algorithm based on the Mobile Speed

Wooyeob Lee Inwhee Joe<sup>o</sup>

Department of Electronics Computer Engineering, Hanyang University

## 요 약

기존의 네트워크 환경에 이동성을 지원하기 위하여 Internet Engineering Task Force (IETF)에서는 Mobile IP(MIP)와 Mobile IPv6 (MIPv6)를 순차적으로 제안하였다. 그러나 MIP나 MIPv6는 유저가 요구하는 이동성을 지원하기에 많은 문제점을 나타내었고 이런 문제를 해결하기 위해 Hierarchical MIPv6(HMIPv6)를 제안하였다. HMIPv6는 Home Agent (HA)를 향한 Binding Update (BU)를 줄이기 위해 Mobility Anchor Point (MAP) 라는 새로운 개념을 제안했다. 그러나 HMIPv6 역시 이동노드가 빠르게 이동하는 환경에서는 잦은 MAP BU로 인해 큰 오버헤드가 발생하는 문제점이 나타났다.

본 논문에서는 HMIPv6 환경에서 노드가 빠르게 이동할 때 발생하는 불필요한 MAP BU를 줄이기 위한 선택적 MAP 등록 기법을 제안한다. 이 기법은 기존에 제안되었던 계층적인 MAP 선택 기법들과는 다르게 상황에 따라 불필요한 MAP BU를 회피하는데서 차별성을 갖는다. 또한 OPNET을 이용한 시뮬레이션을 통해 기존의 HMIPv6에 비해 제안된 기법의 성능을 비교한다.

## 1. 서 론

최근 노드의 이동성에 대한 필요성이 증가함으로 인해 IETF에서는 IP 환경에서의 노드의 끊임 없는 이동성을 지원하기 위해서 MIP를 제안하였다. 그러나 빠르게 증가하는 무선 이동 네트워크 환경에서의 이동성을 지원하기에는 충분하지 못했기 때문에 MIPv6를 제안하였다.

MIPv6는 IPv6를 기반으로 하여 기존의 MIP보다 진화된 이동성을 제공하기 위해 제안되었다 그러나 MIPv6 역시 기존의 MIP와 마찬가지로 노드가 자주 이동하는 환경에서는 등록과정에 따른 오버헤드의 측면에서 취약함을 드러냈다. 또한 잦은 등록과정에 따른 등록비용과 배터리 소모 역시 크게 증가했기 때문에 효율적이지 못했고 이를 해결하기 위해서 지역적으로 이동노드를 관리하는 HMIPv6가 제안되었다.

HMIPv6는 MAP라는 새로운 개념을 제안하였다 MAP은 이동노드의 등록과정을 지역적으로 처리하게 하는 중간 노드의 역할을 수행한다 각각의 이동노드는 가장 적합한 MAP을 선택하여 해당 MAP에 local BU 메시지를 이용하여 등록과정을 수행한다 만약 이동노드가 같은 MAP 영역에 속해 있는 Access Router (AR) 사이를 이동한다면, 이동노드는 HA에 등록할 필요 없이 자신이 속한 MAP에만 등록을 하게 된다. HMIPv6의 문제점은 하나의 MAP 영역 안에 많은 이동노드가 집중되어 이동하고 있을 경우 큰 부하가 집중된다는 점이다 큰 부하가 발생하게 되면 전체 네트워크의 성능이 저하되는 물

론이고, 과부하가 걸린 MAP이 이동노드로부터 전송된 등록 메시지를 블록함에 따라 큰 시그널링 오버헤드가 발생하게 된다 이러한 문제를 해결하기 위해서 계층적 MAP 환경을 기반으로 하는 다양한 MAP 선택 기법이 제안되었다. 일반적으로 이 기법들은 이동노드의 속도 MAP까지의 거리 등을 고려하여 이동노드가 빠르게 움직일 경우에는 상위 계층에 위치한 MAP을 선택하고 그렇지 않을 경우에는 하위 계층에 위치한 MAP을 선택하게 함으로써 과부하를 분산시키는데 목적을 뒀고 기법의 특성상 무조건 하나 이상의 MAP을 선택하는 것을 가정했다.

본 논문에서 제안하는 선택적 MAP 등록 기법은 노드가 하나의 MAP 영역 안에서 빠르게 이동할 경우에 발생하는 불필요한 MAP BU를 줄이는데 목적을 둔다 이 기법은 기존에 제안된 계층적 MAP 환경에서의 MAP 선택 기법이 아닌 하나의 MAP 영역에서 이동노드의 속도를 고려하여 MAP BU를 수행할 것인지 혹은 수행하지 않고 넘어갈 것인지를 선택하는데 초점을 둔다

본 논문의 구성은 다음과 같다 2장에서는 기존에 연구된 이동성 관리 방법에 대해서 설명하고 3장에서는 본 논문에서 제안하는 기법에 대해서 설명한다 4장에서는 OPNET을 이용한 시뮬레이션을 통해 성능을 비교하고 5장에서 결론을 내린다

## 2. 관련 연구

MIPv6 환경에서는 이동노드가 새로운 AR 영역에 들어설 때 마다 HA에 BU 메시지를 전송해야 한다 이동노드

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음(ITA-2008(C1090-0801-0047))

와 HA 간의 거리가 멀면 멀수록 BU에 따른 비용과 지연시간도 같이 증가한다 거기가 이동노드가 빠른 속도로 이동하는 상황에서는 AR에 상주하는 시간이 극단적으로 짧아지기 때문에 잦은 핸드오프가 발생하게 되고 그에 따라 BU의 수행 횟수도 증가한다.

HMIPv6는 MIPv6의 취약점을 보완하기 위해서 제안되었다. HMIPv6에서는 HA로 향하는 BU 메시지를 줄이기 위해서 MAP이라는 새로운 개념을 제안하였다. MAP은 MAP 영역 안에 있는 이동노드들을 관리하고 그 이동노드들로부터 전송되는 local BU 메시지를 통해 이동노드의 이동성을 지원하는 기능을 한다.

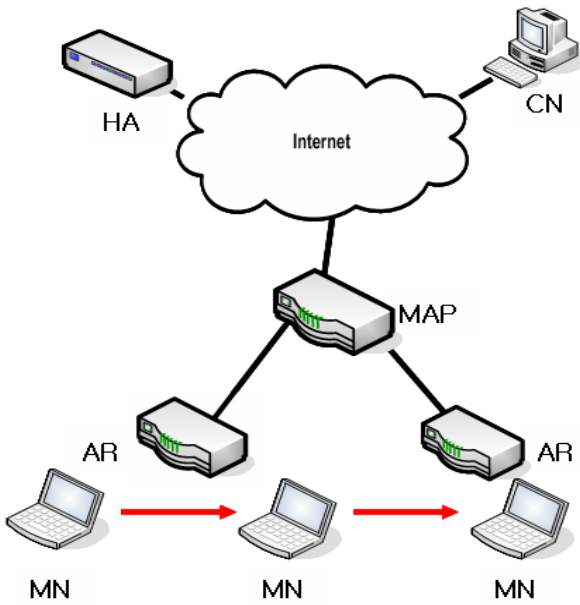


그림 1 HMIPv6 네트워크 구성도

그림 1은 HMIPv6의 네트워크 구성도를 나타낸다. 이동노드가 새로운 MAP 영역에 들어서게 되면, 이동노드는 Router Solicitation (RS) 메시지를 이용하여 Router Advertisement (RA)를 요청한다. MAP이 RS 메시지를 받으면 RA 메시지의 M 필드를 1로 설정한 후 이동노드에게 RA 메시지를 전송하고 자신의 MAP routing table에 해당 이동노드를 등록한다. 이동노드는 RA 메시지를 받음으로써 현재 자신이 속해 있는 MAP의 RCoA를 알게 되고, 그 후에 HA와 Correspondent Node (CN)에게 BU 메시지를 전송하게 되는데 이 때 BU 메시지에 자신이 속한 MAP의 RCoA가 포함된다. 등록과정이 모두 종료된 후에는 이동노드가 하나의 MAP 영역에 속한 AR 사이를 이동할 경우에는 HA에 BU 메시지를 전송하지 않고 MAP에 local BU 메시지를 전송하여 등록과정을 수행하게 된다. MAP은 HA에 비해서 상대적으로 MN와의 거리가 짧기 때문에 BU 비용과 지연시간이 효과적으로 감소된다. 또한 기존의 MIPv6와는 다르게 두 개의 터널이 생성된다. 기존에는 HA와 이동노드 사이에 터널이 있었지만 HMIPv6에서는 HA와 MAP사이, 그리고 MAP과 이동노드 사이에 터널이 생성된다.

그러나 앞서 언급한 것처럼 하나의 MAP 영역안에 많은 이동노드가 존재하고 있으면 잦은 local BU 수행으로

인해 해당 MAP에 과중한 부하가 발생하게 된다. 이런 취약점을 보완하기 위해서 MAP 선택 기법이나 핸드오프 결정 기법 등과 같은 많은 해결책들이 제안되었다.

### 3. 선택적 MAP 등록 기법

본 논문에서 제안하는 선택적 MAP 등록 기법은 잦은 핸드오프로 인해 발생하는 불필요한 local BU 수행을 줄이기 위해 이동노드의 속도를 고려하여 상황에 따라 불필요한 BU를 회피하는데 목적을 두고 있다.

이동노드가 새로운 MAP 영역에 진입하게 되면, 먼저 이동노드가 자신의 이동속도를 확인하고 HIGH SPEED인지 LOW SPEED인지를 결정한다. 만약 이동속도가 HIGH SPEED로 결정되면 MN은 local BU를 수행하지 않고 지나가고, 이동속도가 LOW SPEED로 결정되면 기존의 HMIPv6와 같은 수행을 하게 된다.

#### 3.1 이동속도 결정 기법

앞에서 언급한 것처럼 본 논문은 이동노드의 속도를 이용한다. 본 논문에서는 속도를 측정하기 위한 방법으로 GPS 장치를 사용하는 것을 가정하고 있다. GPS 장치를 이용하여 얻은 이동속도는 기본적으로 HIGH SPEED와 LOW SPEED로 나뉘게 된다.

이동속도를 나누는 척도는 그림 2와 같다.  $T_B$ 는 BU를 수행하는데 소요되는 시간으로 BU의 시작부터 끝까지의 시간을 나타낸다.  $T_P$ 는 하나의 MAP 영역을  $v$ 라는 속도를 이동할 때 소요되는 최대 상주시간을 나타낸다. 위의 두 가지 시간 값을 이용하여 속도를 나누는 임계값을 정하게 된다.

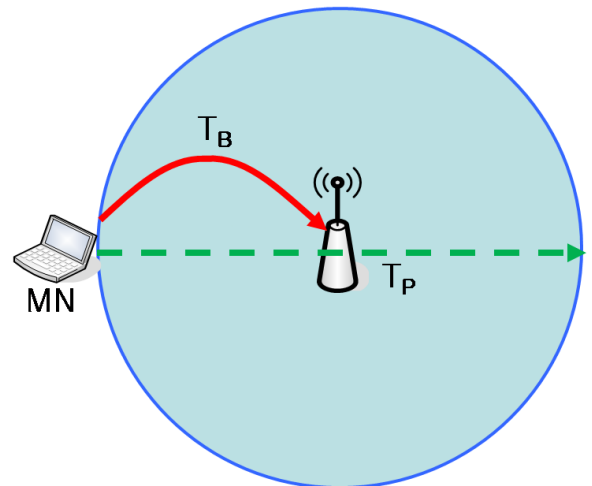


그림 2 이동속도 결정 요소

만약  $T_P$  값이  $T_B$  값 보다 크면 이동노드가 HIGH SPEED로 이동한다고 결정한다. 만약  $T_P$  값이  $T_B$  값 보다 작거나 혹은 같으면 이동노드가 LOW SPEED로 이동한다고 결정한다.

$$T_P > T_B$$

### 3.2 선택적 등록 기법

이동노드의 속도를 결정하고 나면 local BU를 수행할 것인지 아닌지를 결정한다

표 1은 선택적 MAP 등록 기법을 pseudo 코드로 나타낸 것이다. 본 논문에서는 current timer와 accumulated timer라는 두 가지의 counter를 제안한다. Current timer는 이동노드가 새로운 RA 메시지를 받게 되면 0부터 초단위로 증가하게 된다. Current timer 값이  $T_B$  값 보다 작은 동안에는 local BU를 하지 않고 회피하는데 아무런 오버헤드나 지연시간이 발생하지 않고 이는 핸드오버 횟수의 측면에서 봤을 때도 불필요한 핸드오버를 줄이는 것과 같은 효과를 얻게 된다. Accumulated timer는 연속으로 회피할 수 있는 최대 local BU 횟수를 제한하기 위해서 사용한다. 표 1에 보이는 것처럼 accumulated timer는  $T_B$  값을 초과할 수 없다. 그 이유는  $T_B$  값을 초과한다는 것은 local BU의 회피를 결정하는데 사용되는 임계값을 초과하는 것과 같은 의미로 오버헤드와 지연시간이 발생한다는 의미와도 같기 때문이다

만약 current timer의 값이  $T_B$  값을 초과하게 되면, 이동노드는 현재 자신이 속한 MAP으로 local BU를 수행하게 되는데 이는 이동노드가 아직 해당AR의 영역에 있기 때문이다.

표 1 제안된 기법의 Pseudo 코드

```

if (High Speed)
  if (Current_timer < TB)
    increase current_timer
    increase accumulated_timer
    if (new RA_RECV)
      set current_timer to 0
      break

  if (accumulated_timer = TB)
    set current_timer to 0
    set accumulated_timer to 0
    perform HO process
    break
else
  set current timer to 0
  perform HO process
  break
else
  same process as HMIPv6
    
```

### 4. 시뮬레이션

이 장에서는 제안하는 기법의 네트워크 모델 시뮬레이

션과 그 성능을 비교한다. 본 논문에서는 이동노드가 HA로부터 MAP까지 이동하는 네트워크 모델을 구성하여 시뮬레이션을 수행한다

네트워크 모델은 하나의 HA와 MAP을 기준으로 HA와 MAP 사이에 MAP 영역에 속하지 않는 AR이 하나 존재하고, MAP 영역 안에는 3개의 AR이 있도록 구성하였다. 그림 3은 시뮬레이션에 사용된 네트워크 모델이다

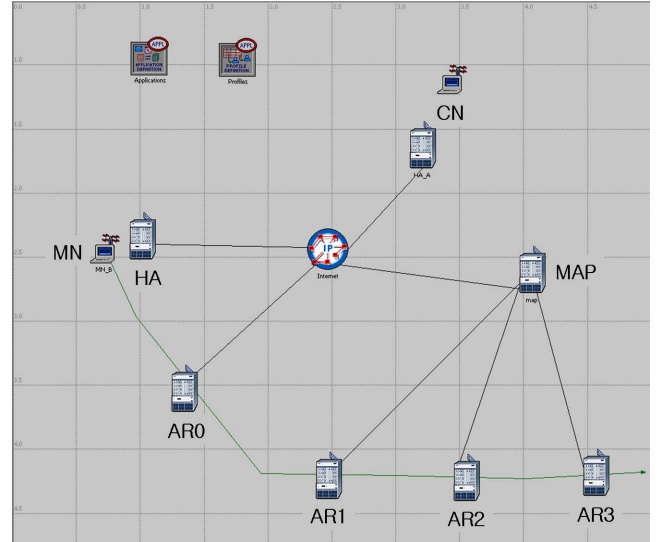


그림 3 네트워크 모델

이동노드는 MAP에 속해 있는 AR을 향해 이동한다. 이동노드가 MAP 영역에 진입하게 되면, 이동노드는 속도를 변화시키면서 MAP 영역에 속한 AR 사이를 이동하도록 설정되었다. 본 논문에서는 이동노드가 AR2의 영역을 제외한 모든 영역에서 고정된 속도로 이동한다고 가정한다.

시뮬레이션에 사용된 파라미터 값 들은 표2에 명시되어 있다. 또한  $T_B$  값은 원활한 시뮬레이션을 위해서 1분으로 고정했다.

표 2 시뮬레이션 파라미터

고정 속도	2.778 m/s
HIGH SPEED	33.333 m/s
송신 파워	0.8 mW

송신 파워는 OPNET에서 사용되는 거리 공식에 의해서 AR의 범위를 500m로 설정하기 위해서 결정되었다. 사용된 수식은 다음과 같다

$$\left( \frac{4\pi D}{0.12476} \right)^2 \times 10^{-12.5}$$

그림 4는 시뮬레이션의 결과를 나타낸다. HMIPv6 환경에서는 이동노드가 HIGH SPEED로 AR2를 통과할 때도 AR2에 local BU를 수행하는 것을 볼 수 있다. 상주 시간을 고려하면 local BU가 종료되기 이전에 이동노드

가 AR2를 통과하기 때문에 오버헤드가 발생한다 본 논문에서 제안한 기법 환경에서는 이동노드가 자신의 이동 속도를 확인하고 local BU에 소요되는 시간보다 통과하는 시간이 짧기 때문에 AR2 구간에서 local BU를 수행하지 않고 지나가는 것을 확인할 수 있다

그림 5는 제안된 기법과 HMIPv6간의 local BU의 횟수를 비교한 그래프이다 HMIPv6 환경에서는 이동노드는 자신이 통과하는 모든 AR의 수만큼 local BU를 수행하게 된다.

5. 결론

본 논문은 local BU의 수행을 결정하는 선택적 MAP 등록 기법을 제안하였다. 이동노드의 속도를 구분하는 방법을 제시하고 local BU 수행을 결정하는 기법을 제안했다. 또한 OPNET을 이용하여 시뮬레이션을 수행하였고, 결론적으로 제안한 기법과 HMIPv6의 local BU 수행 횟수를 비교함으로써 제안한 기법이 기존의 HMIPv6에 비해 좋은 결과를 나타냄을 보였다

참고문헌

[1] D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6", IETF RFC 3775, June 2004.  
 [2] H. Soliman, C. Castelluccia, K. El Malki, L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)", IETF RFC 4140, August 2005.

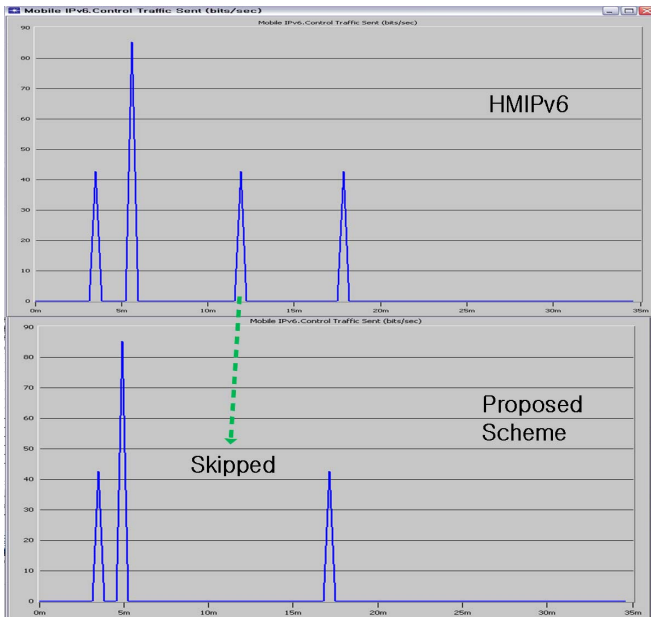


그림 4 시뮬레이션 결과

HMIPv6와는 다르게, 제안된 기법 환경에서는 local BU 횟수가 효과적으로 줄어든 것을 확인할 수 있다 이전 장에서 설명한 것처럼  $T_p$  값이  $T_B$  값에 도달하게 되면 오버헤드를 발생시키지 않기 위해서 강제적으로 local BU를 수행하게 했기 때문에 두 개의 AR을 통과하면 한번의 local BU가 증가하는 것이다

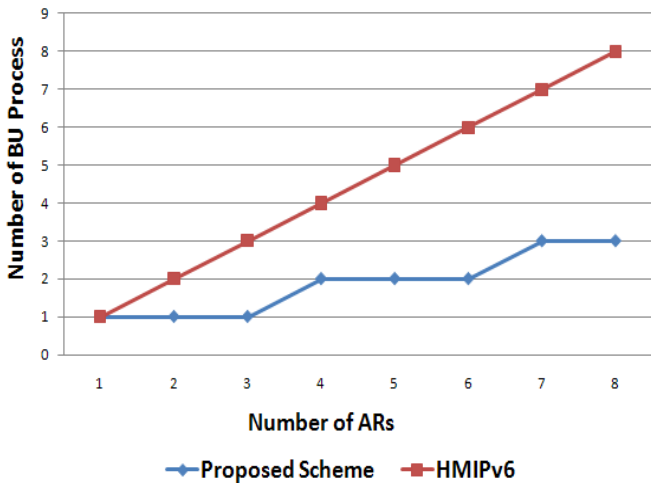


그림 5 Local BU 횟수 비교