Mobile IPv6 에서의 향상된 티켓 기반 바인딩 갱신 프로토콜에 관한 연구*

이중희*, 이종혁**, 한영주**, 정태명*
*성균관대학교 정보통신공학부
**성균관대학교 컴퓨터공학과
e-mail : {jhlee00, jhlee, yjhan}@imtl.skku.ac.kr
tmchung@ece.skku.ac.kr

A Study on Enhanced Binding Update based on Ticket for Mobile IPv6*

Joong-Hee Lee*, Jong-Hyouk Lee**, Young-Ju Han**, Tai-Myoung Chung*
*School of Information and Communication Engineering, Sungkyunkwan University
**Dept. of Computer Engineering, Sungkyunkwan University

요 약

Mobile IPv6 의 바인딩 갱신 메커니즘은 이동 노드가 새로운 링크로 이동하여 처음으로 시행되는 바인딩 갱신과 같은 링크 내에 머물고 있을 때, 혹은 바인딩의 수명으로 인하여 시행되는 바인딩 갱신으로 나눌 수 있다. 하지만 이 모든 과정을 모두 동일한 메커니즘으로 시행하기 때문에 비효율적이다. 이러한 단점을 해결하기 위한 방안으로 티켓 기반의 바인딩 갱신 프로토콜[4]이 제안되었다. 그렇지만 이것은 대응 노드가 고정 노드라는 가정 하에 만들어졌기 때문에 대응 노드가 이동 노드일 경우 바인딩 갱신이 비효율적인 경로를 통해 이루어진다. 이에 본 논문에서는 티켓을 이용하여 첫 번째 바인딩 갱신 이후에 이동 노드와 대응 노드가 홈 에이전트의 도움 없이 바인딩을 갱신하며 대응 노드가 이동성을 갖는 경우에도 효과적으로 해결할 수 있는 프로토콜을 제시한다. 본 논문에서 제안하는 프로토콜은 대응 노드가 이동성을 갖는 경우 기존의 티켓 기반의 바인딩 갱신 프로토콜에 비해 적합하며 대응 노드를 고정 노드로 가정하고 진행된 다른 많은 연구에도 적용될 수 있다.

1. 서론

Mobile IPv6(MIPv6)에서 이동 노드(Mobile Node; MN)는 홈 주소(Home Address; HoA)와 의탁 주소(Care-of Address; CoA) 두 개의 주소를 갖는다[1]. 이 두개의 주소를 바인딩 갱신(Binding Update; BU)을 통해 MN 의 현재 위치를 홈 에이전트(Home Agent; HA)와 대응 노드(Correspondent Node; CN)에게 알리는 과정이 필요하다[1,2].

CN 이 MN 의 CoA 를 모를 경우 MN 과 CN 간의 통신은 항상 HA 를 통해서 이루어져야 하고 이것은 삼각 라우팅 문제를 유발한다. 이러한 문제점을 해결하기 위해서 MN 은 CN 에게 BU 를 통해 경로 최적화(Route Optimization) 를 하고, 이로 인해 MN 은 CoA 를 이용하여 CN 과의 직접적인 통신이 가능해진다[1,3].

본 논문에서는 기존의 BU 는 매번 동일한 메커니즘을 통해서 하기 때문에 발생하는 문제점을 해결하며 CN 이 이동성을 갖을 때에도 효과적인 바인딩 갱신을 수행하는 티켓 기반의 바인딩 갱신 프로토콜을 제안한다. 또한 CN 을 고정 노드로 가정하고 진행되어온 기존의 연구들도 보완할 수 있는 방법을 제시한다.

본 논문의 나머지 구성은 아래와 같다. 2 장은 기존의 티켓 기반 바인딩 갱신 프로토콜에 대해 알아보고

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성・지원사업의 연구결과로 수행되었음.

3 장은 기존 방식의 단점을 극복하는 새로운 티켓 기반 바인딩 갱신 프로토콜을 제안한다. 4 장은 제안하는 프로토콜과 기존의 프로토콜의 성능을 비교 분석한다. 마지막으로 5 장에서는 정리 및 향후의 연구 방향을 제시한다.

2. 기존에 제안된 티켓 기반 바인딩 프로토콜의 분석

기존의 BU 메커니즘에서 BU 는 BU 가 일어날 때마다 매번 동일한 메커니즘을 통하여 이루어 진며 이로 인해 두가지 문제점이 발생한다. 첫째로는 새로운 링크로의 이동이 빈번하게 발생하는 MN 에게 있어서 동일한 과정을 반복하는 것은 비효율적이다. 둘째로 바인딩의 수명이 끝났을 경우 MN 이 동일한 네트워크 내에 있더라도 동일한 과정을 통해서 BU 를 하게 된다. 이러한 문제점을 극복하기 위해서 티켓 기반의 BU 프로토콜이 제안되었다[4].

기존의 제안된 티켓 기반의 BU 프로토콜을 간단히 설명하자면 (그림 1)과 같다.

(그림 1) 티켓 기반의 바인딩 갱신 과정

(그림 1)에서 1 번부터 4 번까지의 메시지는 첫 번째 BU 에 이용되고 5 번은 그 이후의 BU 에 이용된다. 첫 번째의 BU 에서 MN 은 자신의 HA 에게로의 BU 와 CN 으로의 BU 를 요청한다. 이에 HA 는 CN 에게 BU 를 요청하고 첫 번째 BU 이후의 갱신에서 이용되어질 티켓 발행을 요청한다. CN 은 HA 로부터의 BU 에 대한 확인 메시지인 바인딩 확인(Binding Acknowledgment; BA)과 티켓을 발행하여 HA 에게 전송한다. HA 는 CN 으로부터 받은 티켓과 MN 이 요청한 BU 에 대한 BA 메시지를 전송하여 준다. 이로써 첫 번째 BU 를 마친다. 첫 번째 BU 이후의 BU 를 하기 위해 MN 은 첫 번째 BU 에서 발급 받은 티켓을 이용하여 HA 의 도움없이 CN 과 직접 BU 를 수행한다.

2.1 기존 티켓 기반 바인딩 갱신 프로토콜의 가정

* MN 과 CN 은 서로의 CGA(Cryptographycally Generated Addresses)[5]에 대해 확신한다.
* MN 과 HA 는 공유된 대칭키를 가지고 있다.
* **CN 은 이동 노드가 아닌 고정 노드이다.**

2.2 표기법

* MN/HA/CN: 이동 노드/이동 노드의 홈 에이전트/대응 노드
* HoA/CoA: MN 의 홈 주소/의탁 주소
* HA_{addr}/CN_{addr}: HA의 주소/CN의 주소
* BU/BA: 바인딩 갱신/바인딩 갱신에 대한 응답
* $T_e/N_e/L$: 노드 e의 타임 스탬프/난수/라이프 타임
* MAC(K,M): 암호키 K 를 이용한 메시지 M 에 대한 MAC 값
* K_{HA-CN}/K_{MN-CN}: HA와 CN사이의 비밀키/MN과 CN 사이의 세션키
* x_{MN}/g^x_{MN}: MN의 Diffie Hellman 개인키/공개키 쌍
* y_{CN}/g^y_{CN}: CN의 Diffie Hellman 개인키/공개키 쌍
* Sig(): 전자서명
* TckMN-CN: MN 과 CN 사이의 티켓. MN 에서 핸드오프 발생 시에 CN 과의 인증을 위해서 사용
* A||B: 메시지 구성요소 A 와 B 의 비트 결합.
* prf(k,m): Pseudo 랜덤 함수(k:키, m:메시지)

2.3 메시지

2.3.1 HA 와의 BU 및 CN 과의 BU 요청

$CoA, HA_{addr}, CN_{addr}, HoA, n_{MN}, T_{MN}, L_{BU},$
$MAC(K_{MN-HA}, CoA||CN_{addr}||HoA||n_{MN}||T_{MN})$

2.3.2 CN 과의 BU 및 티켓 발행 요청

$HoA, CN_{addr}, CoA, n_{MN}, n_{HA}, T_{HA}, g^x_{HA}, L_{BU}, S^{\#}, Cookie_1,$
$Sig(x_{HA}, h(CoA||CN_{addr}||HoA||n_{MN}||n_{HA}||T_{HA}||g^x_{HA}||S^{\#}||L_{BU}))$

2.3.3 BU 에 대한 BA 전송 및 티켓 발행

$CN_{addr}, HoA, CoA, n_{HA}, n_{CN}, T_c, g^y_{CN}, S^{\#}, Tck_{MN-CN}, Cookie_1,$
$Cookie_2, MAC(K_{HA-CN}, CN_{addr}||CoA||HoA||T_c||n_{CN}||g^y_{CN}||$
$Tck_{MN-CN}||S^{\#}||L_{BA})$
$*K_{HA-CN} = H(g^{xy}||n_{HA}||n_{CN}||Cookie_1||Cookie_2)$
$*K_{MN-CN} = prf(K_{HA-CN}, n_{MN}||n_{CN})$
$*Tck_{MN-CN} = \{HoA||n_{CN}||T_{CN}||L_{Tck}||K_{MN-CN}\}y_{CN}$
$*T_c = T_{HA}||T_{CN}$

2.3.4 HA 와 CN 의 BA 전송 및 티켓 전송

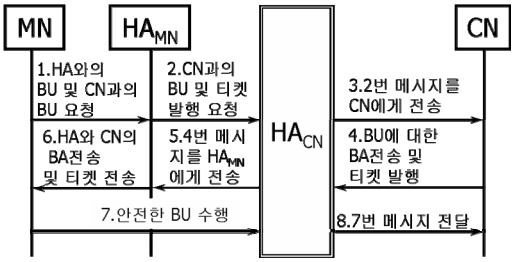
$CN_{addr}, CoA, HoA, n_{MN}, T_a, Tck_{MN-CN}, L_{BA},$
$\{n_{MN}, T_a, K_{MN-CN}, L_{BA}\}K_{MN-HA}$
$*T_a = T_c||T_{MN}$

2.3.5 안전한 BU 수행

$BU : CoA, CN_{addr}, HoA, n_{MN}, T_{MN}, Tck_{MN-CN}, Cookie_1, L_{BU},$
$MAC(K_{MN-CN}, CoA||CN_{addr}||HoA||n_{MN}||T_{MN})$
$BA : CN_{addr}, CoA, n_{MN}, T_c, Cookie_1, Cookie_2, L_{BA},$
$MAC(K_{MN-CN}, CN_{addr}||CoA||n_{MN}||T_c||Cookie_1||Cookie_2||L_{BA})$
$*T_c = T_{MN}||T_{CN}$

2.4 기존의 프로토콜의 단점

CN 이 이동성을 갖을 경우 MN 의 BU 요청 메시지는 CN 의 HA 를 통해서 CN 에게 전해지게 된다. 또한 이후의 BU 도 CN 에게 직접 전해지는 것이 아닌 CN 의 HA 의 도움을 얻어야 한다는 단점이 있다.



(그림 2) 기존 티켓 기반 바인딩 갱신 프로토콜의 단점

(그림 2)에서 3, 5, 8 번 메시지는 CN 이 고정 노드일 때에 비해서 CN 이 이동성을 갖을 경우에 발생하는 오버헤드이다.

3. 제안하는 프로토콜

본 논문에서 제안하는 프로토콜은 [4]에서 제안된 티켓 기반의 바인딩 갱신 프로토콜의 단점을 보완하기 위한 것이다. (그림 2)에서 볼 때, CN 과 통신을 원하는 MN 이 CN 의 CoA 를 알고 있다면 MN 은 CN 을 고정 노드와 같이 다룰 수 있기 때문에 오버헤드인 3, 4, 8 번 메시지가 없어질 수 있다. 따라서 MN 으로 하여금 CN 의 CoA 를 알게하는 과정이 선행된다면 이러한 문제를 해결할 수 있다.

3.1 표기법

*BUN: 바인딩 갱신을 요청하는 노드
*BAN: 바인딩 갱신을 확인하는 노드
*BUH/BAH: BUN 의 홈 에이전트/BAN 의 홈 에이전트
*BUN_{CoA}/BUN_{HoA}: BUN의 의탁 주소/홈 주소
*BAN_{CoA} /BAN_{HoA}: BAN의 의탁 주소/홈 주소
*BUH_{addr}/BAH_{addr}: BUH의 주소/BAH의 주소
*n_{BUN}: BUN이 생성한 난수
*x_{BUN}/g^x_{BUN}: BUN의 Diffie-Hellman 개인키/공개키 쌍
*x_{BAH}/g^x_{BAH}: BAH의 Diffie-Hellman 개인키/공개키 쌍
*K_{BUN}: BUN의 관용암호 키

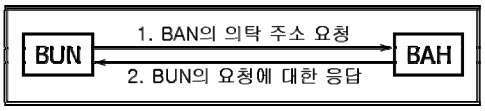
3.2 제안하는 프로토콜

3.2.1 프로토콜의 가정

* BUN 과 BAN 모두 이동성을 갖을 수 있다.
* BAN 의 CoA 는 신용할만한 BU 과정을 통해 BAH 의 Binding Cache[1]에 저장되어 있다.

3.2.2 제안하는 프로토콜의 상세 설명

앞에서 설명한 바와 같이 BAN 이 MN 이 될 수 있을 때 최초의 BU 를 위해서 BUN 로 하여금 BAN 의 CoA 를 알게하는 과정이 선행된다면 기존의 티켓 기반의 바인딩 갱신 프로토콜을 그대로 이용할 수 있다. 또한 이 과정은 한번 이상의 BU 가 이미 이루어진 후라면 BAN 과 BUN 의 CoA 는 이전의 갱신을 통해서 각 노드의 Binding Cache 에 저장되어 있기 때문에 생략될 수 있다. 따라서 본 논문에서는 BU 를 하기에 앞서서 BAN 이 BUN 에게 의탁 주소를 알리는 과정(그림 3)만을 제안하고 나머지 갱신 과정은 [4]에서 제안한 바인딩 갱신 프로토콜을 따른다.



(그림 3) BAN 의 의탁 주소 요청 과정

3.2.2.1 BUN 의 BAN 의탁 주소 요청 메시지

BUN 이 BAN 에게 BU 를 하기 위해서는 BAN 의 CoA 가 선행적으로 필요하다. 따라서 BUN 은 BAN 의 HoA 로 BAN 의 CoA 를 요청하게 된다. 이 메시지는 우선 BAH 에게 전해진다. BAH 의 Binding Cache 에 BAN 의 CoA 가 이미 저장되어 있기 때문에 이 메시지에 대한 응답은 BAH 가 BUN 에게 직접 하면 된다. 따라서 이 메시지는 BAN 에게 전해질 필요가 없다. 메시지의 형식은 다음과 같다.

$$\{ g^x_{BUN}, \{ g^x_{BUN}, BUN_{CoA}, BAN_{HoA}\} x_{BUN} \} g^x_{BAH}$$

BUN 은 메시지를 자신의 Diffie-Hellman[6] 개인키로 암호화 하고 BAH 의 공개키로 다시 한번 암호화 하여 전송한다. BAH 는 자신의 개인키로 복호화 한 다음 BUN 의 공개키로 한 번 더 복호화함으로써 메시지를 인증한다.

3.2.2.2 BUN 의 요청에 대한 BAH 의 응답

3.2.2.1 에서 설명한 바와 같이 BUN 의 BAN CoA 요청 메시지는 BAN 까지 전달 될 필요가 없다. 이에 대한 응답은 BAH 의 Binding Cache 에 저장되어 있는 정보를 통해 BAH 가 직접하게 된다. 메시지의 형식은 다음과 같다.

$$\{ g^x_{BAH}, \{g^x_{BAH}, BAN_{HoA}, BAN_{CoA}, Lifetime\} x_{BAH} \} g^x_{BUN}$$

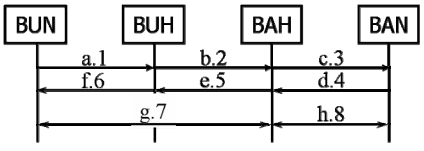
메시지 인증을 위한 암호화는 CoA 요청 메시지와 동일한 방식으로 이루어 진다. 여기에서 Lifetime 은 BAH 의 Binding Cache 에 저장되어 있는 BAN 의 Binding 에 대한 Lifetime 이다. Lifetime 이 초과되면 BUN 과 BAN 은 BU 를 새로 해야하며 이것은 최초의 BU 가 아니므로 2 장에서 설명된 방식대로 티켓을 통한 갱신이 이루어진다.

4. 성능 분석 및 평가

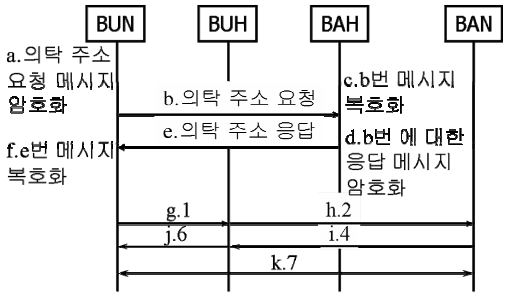
4.1 분석적 모델링을 통한 비교

데이터 전송 흐름에 따라 메시지를 보내고 처리하는 총시간을 기존의 티켓 방식 바인딩 갱신 프로토콜과 본 논문에서 제안하는 프로토콜과 비교하기 위해 (그림 4,5) 의 절차를 고려한다.

(그림 4,5)에서 각 과정을 나타나는 숫자는 (그림 2) 의 각 숫자에 해당하는 메시지와 동일하다.



(그림 4) 기존 티켓 방식 바인딩 갱신 프로토콜 절차



(그림 5) 제안하는 프로토콜의 절차

메시지를 무선에서 전달하는 시간은 두 프로토콜 모두 동일하고 두 프로토콜에서 공통되는 프로세싱 시간 역시 동일하므로 고려하지 않는다.

4.1.1 기존 티켓 방식 바인딩 갱신 프로토콜 절차

제안하는 프로토콜과의 비교를 위해 공통된 부분은 고려하지 않으므로 기존 티켓 방식 바인딩 갱신 프로토콜에서 고려해야 할 사항은 유선 상에서 메시지가 전송되는 시간만 고려하면 된다.

유선 상에서 메시지가 전송되는 경우는 첫 번째 바인딩 갱신을 위해서는 (그림 4)의 a, b, c, d, e, f 단계에서 필요하고 그 이후의 바인딩 갱신에서는 g, h 단계에서 필요하다.

메시지가 전송되는데 걸리는 시간의 합을 구하면 다음과 같다. n 은 첫 번째 BU 이후의 갱신의 횟수이다.

$$L_{total} = M_{BUN,BUH} + M_{BUH,BAH} + M_{BAH,BAN} + M_{BAN,BAH} + M_{BAH,BUH} + M_{BUH,BUN} + 2n(M_{BUN,BAH} + M_{BAH,BAN})$$

여기에서 각 단계에서의 메시지 전송시간이 M^L으로 동일하다면 $L_{total} = 6M^L + 4n*M^L$로 정리된다.

4.1.2 제안하는 프로토콜 절차

제안하는 프로토콜에서 공통된 부분을 뺀 나머지 절차는 CoA 요청과 응답 메시지를 위한 암호화와 복호화에 걸리는 프로세싱 시간과 유선 상에서 메시지가 전송되는 시간의 합으로 표현될 수 있다.

프로세싱 시간은 다음과 같이 표현한다.

$$S_{total} = S_{Erequest} + S_{Drequest} + S_{Ereply} + S_{Dreply}$$

여기에서 각 프로세싱 시간이 S 로 동일하다면, $S_{total} = 4S$로 정리된다.

메시지가 전송되는데 걸리는 시간은 다음과 같다.

$$L_{total} = M_{BUN,BAH} + M_{BAH,BUN} + M_{BUN,BUH} + M_{BUH,BAN} + M_{BAN,BUH} + M_{BUH,BUN} + 2n(M_{BUN,BAN})$$

여기에서 각 단계에서의 메시지 전송시간이 M^L으로 동일하다면 $L_{total} = 6M^L + 2nM^L$ 으로 정리된다.

따라서 제안하는 프로토콜 절차에서 기존의 티켓 기반 바인딩 갱신 프로토콜과 공통된 부분을 뺀 나머지의 총합은 다음과 같다.

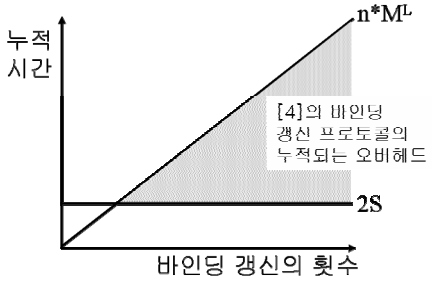
$$T_{total} = 4S + 6M^L + 2n*M^L$$

4.2 결과 분석 및 비교

기존의 티켓 기반 바인딩 프로토콜에 비해 제안하는 프로토콜이 우수하기 위해서는 다음과 같은 식이 성립한다.

$$6M^L + 4nM^L > 4S + 6M^L + 2n*M^L$$

이 식은 $n*M^L > 2S$ 으로 정리할 수 있다. M^L로 표현된 기존 프로토콜에서 바인딩 갱신이 비효율적인 경로를 통해 이루어지기 때문에 발생하는 시간의 손실은 바인딩 갱신의 횟수가 거듭될수록 늘어난다. 그리고 S로 표현된 암호화, 복호화 과정은 가장 처음의 바인딩 갱신을 위해 BAN의 CoA를 요청, 응답하는 과정에서 한번만 이루어진다. 따라서 (그림 6)에서와 같이 바인딩 갱신의 횟수가 늘어남에따라 본 논문에서 제안한 프로토콜이 [4]에서 제안한 프로토콜에 비해 우수하다는 의미를 갖는다.



(그림 6) [4]의 바인딩 갱신 프로토콜과의 성능 비교

5. 결론

본 논문에서는 기존의 티켓 기반 바인딩 갱신 프로토콜이 CN 을 고정 노드로 가정하였기 때문에 생기는 문제를 개선하기 위한 프로토콜을 제시하였다. 또한 여기에서 이용된 방식은 기존의 CN 을 고정 노드로 가정하고 진행된 모든 연구를 개선 시키는데 적용되어질 수 있다. 그리고 제시한 프로토콜에서 전송되는 메시지는 일회성이며 기밀성이 보장될 필요가 없고 또한 메시지의 크기가 작기 때문에 매우 유용하게 이용되어질 수 있다. 향후 연구 과제로 더욱 성능을 향상시키기 위해 계산 능력이 떨어지는 모바일 노드에 적합한 암호화 기법과 인증 기법을 연구할 것이다.

참고문헌

[1] D. Johnson, C.Perkins, J. Arkko, “Mobility Support in IPv6”, IETF RFC 3775, Jun 2004.

[2] C. Perkins, D. B. Johnson, “Route Optimization in Mobile IP”, IETF Internet Draft, Sep 2001.

[3] P. Nikander, J. Arkko, T. Aura, G. Montenegro, E. Nordmark, “Mobile IP Version 6 Route Optimization Security Design Background”, IETF RFC 4225, Dec 2005.

[4] 구중두, 김상진, 오희국, “MIPv6 을 위한 티켓 기반의 바인딩 갱신 프로토콜”, 제 32 회 추계학술발표회 논문집 Vol.32, pp. 4-6, Nov 2005.

[5] T. Aura, “Cryptographically Generated Addresses(CGA)”, IETF RFC 3972, Mar 2005.

[6] W. Diffie, M.E. Hellman, “New directions in cryptography”, IEEE Transactions on Information Theory 22, pp. 644-654, Nov 1976.