캐쉬 에이전트를 이용한 신뢰성 있는 Mobile IP 핸드오프 기법

김래영° 송주석
연세대학교 컴퓨터과학과
{leon, jssong}@emerald.yonsei.ac.kr

A Reliable Mobile IP Handoff Mechanism using Cache Agent

Lae-Young Kim° Joo-Seok Song
Dept. of Computer Science, Yonsei University

요 약

본 논문에서는 Mobile IP에서 핸드오프 중에 발생하는 패킷손실을 제거하기 위한 신뢰성 있는 핸드오프 기법을 제안하였다. 기존에 제시된 핸드오프 기법은 핸드오프 중에 발생하는 패킷손실을 줄여주지만 여러 가지 문제점을 가지고 있었다. 본 논문에서는 캐쉬 에이전트(Cache Agent)가 대응 노드(Correspondent Node)를 대신하여 바인딩 캐쉬를 관리하며 대응 노드가 이동 노드(Mobile Node)에게 전송한 패킷을 버퍼링함으로써 핸드오프 중에 발생하는 패킷손실을 제거한다.

1. 서 론

최근 컴퓨터 하드웨어의 급속한 발전으로 인해 단말이 소형화, 경량화되고 있으며, 단말에 이동성을 부여하는 것이 점점 보편화되고 있다. 인터넷의 근간을 이루는 TCP/IP는 유선망과 고정 단말을 염두에 두고 만들어진 프로토콜이라 단말의 이동성을 지원하기가 어렵다. 이에 IETF에서는 단말의 이동성을 지원하기 위한 프로토콜로 Mobile IP를 제안하였다[1]. Mobile IP의 기본적인 동작을 살펴보면 각각의 이동 노드(MN: Mobile Node)는 홈 네트워크에서 할당받은 홈 주소(home address)를 가지며 홈 네트워크에는 MN의 이동을 관리하는 홈 에이전트(HA: Home Agent)가 있다. MN이 자신의 HA가 아닌 다른 외부 에이전트(FA: Foreign Agent)가 관리하는 네트워크로 이동했을 경우 MN에게 새로운 IP 주소, care-of address(COA)가 할당되어 이를 FA를 통해 HA로 등록하게 된다. HA는 MN의 현재 위치를 나타내는 COA를 유지함으로써 MN에게 전송되는 모든 패킷을 인터셉트하여 IP-within-IP encapsulation[2]을 이용하여 MN이 위치한 FA로 패킷을 터널링(tunneling)한다. FA는 받은 패킷을 디캡슐레이션(decapsulation)하여 MN에게 전달한다. 따라서 MN과 통신하는 대응 노드(CN: Correspondent Node)는 MN의 현재 위치를 전혀 알 필요가 없다.

Mobile IP는 이처럼 간단하면서도 확장성 있게 노드의 이동성을 지원하지만 몇몇 문제점을 갖고 있다. 첫째, 패킷이 최적의 경로(optimal path)로 전송되지 않는 트라이앵글 라우팅(triangle routing) 문제를 갖는다[3]. 둘째, MN이 빈번하게 이동하는 경우 그때마다 HA에게 새로운 COA를 등록해야 하기 때문에 시그널링 오버헤드(signaling overhead)와 핸드오프 지연(handoff delay)이 발생하며 핸드오프로 인한 패킷손실의 가능성도 있다.

본 논문에서는 Mobile IP 환경에서 사용되는 경로 최적화 및 핸드오프 기법들을 살펴보고 이에 대한 문제점들을 지적하며 핸드오프 중 발생하는 패킷손실이 제거된 신뢰성 있는 핸드오프 기법을 제시한다. 본 논문의 나머지 구성은 다음과 같다. 2장에서는 관련 연구를 살펴보며 3장에는 신뢰성 있는 핸드오프 기법이 제시되고 4장에서는 결론 및 향후 과제가 제시된다.

2. 관련 연구

2.1 바인딩 캐쉬(Binding Cache)를 이용한 경로 최적화 및 핸드오프 기법

Mobile IP route optimization[3]은 CN과 핸드오프하기 전의 FA에게도 MN의 COA를 알려줌으로써 트라이앵글 라우팅과 핸드오프 중 발생하는 패킷손실을 완화시켰다. MN의 홈 주소와 COA의 조합인 바인딩 정보는 기본적인 Mobile IP에서는 HA만이 유지하지만 [3]의 경우 CN도 바인딩 정보를 가짐으로써 MN에게 패킷을 전송할 때 HA를 경유하지 않고 바로 MN의 COA로 전송하여 트라이앵글 라우팅 문제를 해결한다. CN은 HA가 보낸 MN의 COA 정보를 담은 바인딩 업데이트(Binding Update) 메시지를 가지고 바인딩 캐쉬를 관리한다.

MN이 핸드오프 하는 동안 MN에게 전송된 패킷은 종종 손실될 수 있는데 이는 MN으로의 터널링이 MN의 이동 전 위치정보에 근거하여 이루어지기 때문이다. 만약 핸드오프가 빈번하게 발생한다면 패킷손실은 더욱 악화될 것이다.

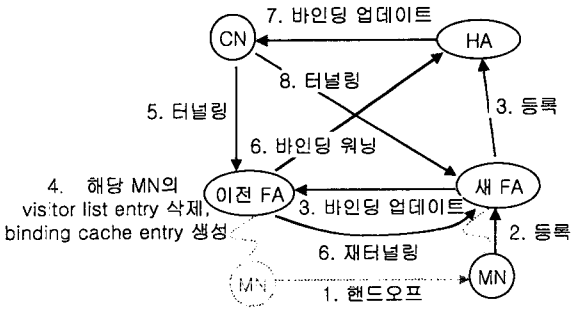

그림 1 스무드 핸드오프(Smooth Handoff)

[3]은 핸드오프 중 발생하는 패킷손실을 줄이기 위한 방안으로 스무드 핸드오프(Smooth Handoff)를 제안했는데 그림 1과 같이 MN이 이동하여 새로운 FA에게 등록을 요청할 때 새로운 FA로 하여금 이전의 FA에게 MN의 현재 위치, 즉 COA를 알리게 함으로써(바인딩 업데이트 메시지 이용) 이전의 FA가 바인딩 캐쉬를 생성하도록 한다. 등록 작업이 완료되는 동안 이전

의 FA에게로 터널링된 패킷은 MN에 대한 바인딩 정보를 가지고 있는 이전의 FA에 의해 새로운 FA로 재터널링된다.

그러나 [3]은 Mobile IP의 중요한 설계 이념을 위배한다. 즉, CN이 바인딩 캐쉬를 유지하기 위해서는 CN의 프로토콜에 수정이 가해져야 하므로 Mobile IP를 투명하게 구현할 수 없다 [4].

2.2 에이전트 기반 경로 최적화 및 핸드오프 기법

CN의 프로토콜에 수정을 가하지 않는, 투명하면서도 경로 최적화를 달성하는 기법으로 Agent-based Route Optimization for Mobile IP가 있다[4]. 여기서는 HA와 FA가 구현된 곳과 동일한 장소(예를 들면 라우터, 기지국 또는 다른 노드들)에 역시 구현할 수 있는 대응 에이전트(Correspondent Agent)라는 것을 두었다. CN으로 향하는 모든 메시지는 대응 에이전트를 거치며 대응 에이전트로 하여금 CN 대신 바인딩 캐쉬를 관리하게 하여 각각의 CN에서 대응 에이전트로 온 MN으로 향하는 메시지를 터널링하는 역할을 하게 했다. 그림 2는 에이전트 기반 경로 최적화 기법을 나타낸다.

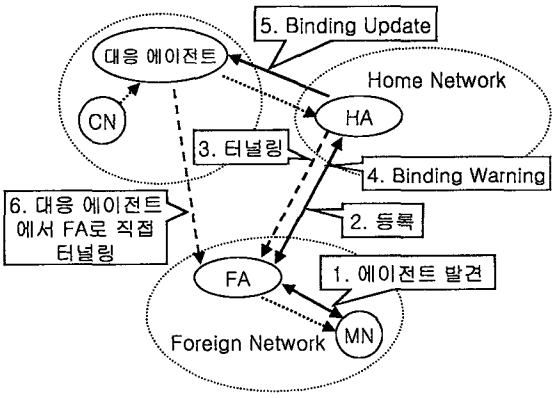

그림 2 에이전트 기반 경로 최적화 기법

MN이 홈 네트워크를 떠나 외부 네트워크로 이동하면 MN은 FA를 발견하여 HA에게로 새로 얻은 COA를 등록하고 그 결과 HA에는 MN의 COA가 저장된다. CN이 MN에게 패킷을 전송하면 HA는 이를 인터셉트하여 MN이 위치한 FA로 터널링한다. 터널링된 패킷을 받은 FA는 이를 디캡슐레이션하여 MN에게 전달한다. 패킷을 전달하기 전에 FA는 받은 패킷의 송신자(sender)를 체크한다. 만약 송신자가 HA라면 FA는 HA에게 MN의 홈 주소 및 현재 위치(COA)를 알리는 바인딩 워닝(Binding Warning) 메시지를 보내고 이를 받은 HA는 대응 에이전트에게 바인딩 업데이트 메시지를 보낸다. 대응 에이전트는 바인딩 업데이트 메시지를 가지고 자신이 지원하는 CN을 대신하여 MN에 대한 바인딩 정보를 관리한다. 이후에 CN이 MN에게 패킷을 전송하면 대응 에이전트는 가지고 있는 바인딩 정보를 사용하여 MN이 위치한 FA로 직접 터널링한다. 따라서 CN의 프로토콜에는 어떠한 수정도 없이 투명하게 경로 최적화를 구현할 수 있다. 물론 대응 에이전트의 기능을 구현해야 하지만 이는 HA, FA와 같은 장소에 구현 가능할 뿐만 아니라 네트워크에서 하나의 대응 에이전트가 많은 CN에 의해 공유될 수 있기 때문에 하나의 대응 에이전트가 모든 CN을 대신해 바인딩 캐쉬를 유지함으로써 좋은 확장성을 가진다. [4]는 핸드오프 중 발생하는 패킷손실을 줄이기 위한 방법으로 [3]과 마찬가지로 핸드오프 이전의 FA가 새로운 네트워크로 이동한 MN에게 전송된 패킷을 포워드(forward)하도록 한다.

[6]의 경우 앞서 살펴본 대응 에이전트와 유사하게 CN을 대신하여 MN에 대한 바인딩 캐쉬를 관리하는 에이전트인 캐쉬 에이전트(Cache Agent)를 이용한 경로 최적화 기법이 제안되었다.

3. 신뢰성 있는 핸드오프 기법

[3], [4]는 핸드오프 중 발생하는 패킷손실을 줄이기는 하지만 결과적으로 이를 방지하지는 못한다. 이는 TCP 혼잡 제어 알고리즘(congestion control algorithm)으로 하여금 패킷의 손실이 핸드오프로 인한 것이 아니라 네트워크의 혼잡에 의한 것이라고 오인하게 만들며 이로 인해 TCP의 성능에 급격한 저하를 가져다 줄 수 있다. 그밖에 문제점으로는 CN 또는 대응 에이전트가 MN의 새로운 위치정보를 얻는 데까지 다수의 컨트롤 메시지가 요구되며 이에 따라 소요되는 시간 또한 문제점이라 할 수 있다. 또한, FA의 프로토콜에 수정이 이루어져야하는데 이는 새 FA가 이전 FA에게 바인딩 업데이트 메시지를 보내는 것과 이전 FA가 바인딩 캐쉬를 관리하는 등의 기능이 필요하기 때문이다. MN과 이전 FA간의 security association이 필요한 것도 구현상의 어려운 점이다. [6]의 경우도 핸드오프 중 발생하는 패킷손실을 막지는 못한다. 본 논문에서는 [4], [6]의 장점은 유지하면서 이러한 문제점들을 해결하는 캐쉬 에이전트를 이용한 신뢰성 있는 핸드오프 기법을 제안한다.

본 논문에서 제시한 캐쉬 에이전트는 [4]에서처럼 CN을 대신하여 MN에 대한 바인딩 캐쉬를 관리할 뿐만 아니라 핸드오프 중 발생하는 패킷손실을 방지하기 위해 CN이 MN에게 보내는 패킷을 버퍼링하는 역할도 한다.

3.1 기본 시나리오

그림 3은 본 논문에서 제안한 캐쉬 에이전트를 이용한 신뢰성 있는 핸드오프 기법의 기본 시나리오를 보여준다.

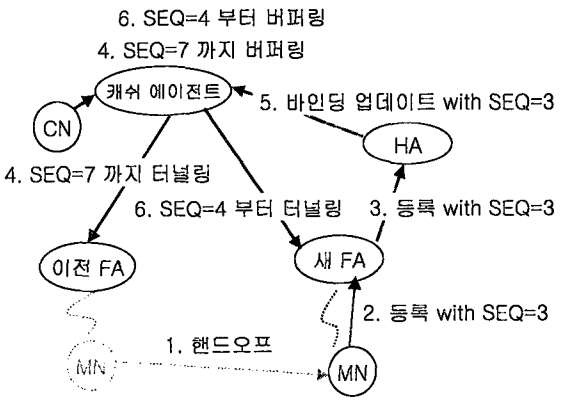

그림 3 캐쉬 에이전트를 이용한 신뢰성 있는 핸드오프

1. MN이 이동하여 새로운 COA를 HA에게 등록하려 한다.
2. MN이 새 FA에게 등록 요청을 보낸다. 이때 MN이 패킷 수신 중에 핸드오프를 했다면 MN이 현재까지 수신한 패킷의 Sequence 번호를 알리는 extension(Cache Agent Notification Extension)도 추가하여 보낸다.
3. 등록 요청을 받은 새 FA는 이를 HA에게로 전달한다.
4. CN으로부터 MN으로 가는 패킷을 받은 캐쉬 에이전트는 MN이 이동한 것을 아직 모르기 때문에 MN의 이전 위치로 패

킷을 전송하고(그림 3의 경우 SEQ=7까지) 이때 전송한 패킷을 버퍼링한다.
5. HA는 새 FA로부터 등록 요청을 전달받아 여기에 Cache Agent Notification Extension이 포함되어 있는 경우 캐쉬 에이전트에게 MN의 새로운 위치정보를 알려주는 바인딩 업데이트 메시지를 전송하는데 이때 Cache Agent Notification Extension도 같이 전송한다. 캐쉬 에이전트는 이 메시지를 받은 후 자신이 관리하는 해당 MN에 대한 바인딩 캐쉬를 업데이트한다.
6. 캐쉬 에이전트는 Cache Agent Notification Extension으로부터 MN이 현재까지 수신한 패킷의 Sequence 번호를 알 수 있으므로 MN이 현재까지 수신한 패킷의 다음 패킷부터(그림 3의 경우 SEQ=4부터) MN의 새로운 위치로 패킷을 전송하며 이때 전송한 패킷의 시작 패킷부터(그림 3의 경우 SEQ=4부터) 보낸 패킷까지를 다시 버퍼링한다. 이때 MN이 수신한 것이 확인된 패킷(그림 3의 경우 SEQ=3까지)은 더 이상 버퍼링할 필요가 없다.

3.2 Cache Agent Notification Extension

본 논문에서 제안한 핸드오프 기법을 구현하기 위해 필요한 Cache Agent Notification Extension은 그림 4와 같은 형태를 가진다.

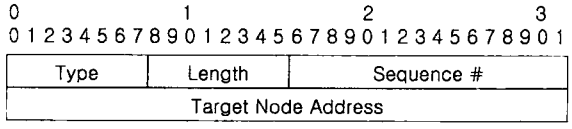

그림 4 Cache Agent Notification Extension

Sequence # 필드는 2 바이트로 MN이 수신한 IP 데이터그램의 IP 헤더[5] 부분 중 Identification 필드에서 그 값을 가져온다. Target Node Address 필드의 경우 4 바이트로 데이터그램을 전송한 CN의 주소이며 이는 MN이 수신한 IP 데이터그램의 IP 헤더에서 Source Address 필드의 값이다.
이 extension은 MN이 어떤 CN으로부터 어느 패킷까지를 수신했는지 알리기 위해 등록 요청을 할 때 추가하여 보낼 수 있으며 이때 필요한 만큼 extension을 추가할 수 있다. 즉, 다수의 CN으로부터 패킷을 수신 중에 MN이 이동한 경우에는 CN의 수만큼 extension을 추가하면 된다. HA가 이 extension이 포함된 등록 요청을 받은 경우 캐쉬 에이전트에게 이 extension을 추가하여 바인딩 업데이트 메시지를 보낸다.

4. 결론 및 향후 과제

핸드오프 중 발생하는 패킷손실을 줄이기 위해 [3]은 스무드 핸드오프를 제안했으나 CN의 프로토콜에 수정을 가함으로써 Mobile IP의 투명성을 해치는 문제점이 있으며 [4]는 CN의 프로토콜에 수정을 가하지 않고 대응 에이전트를 이용함으로써 Mobile IP의 투명성을 유지하지만 모두 여전히 패킷손실의 가능성을 갖고 있다.
이에 본 논문에서는 [4]의 대응 에이전트나 [6]의 캐쉬 에이전트처럼 캐쉬 에이전트로 하여금 CN을 대신하여 MN에 대한 바인딩 캐쉬를 관리하게 할 뿐만 아니라 CN이 MN에게 전송하는 패킷을 버퍼링하게 함으로써 핸드오프 중 발생하는 패킷손실이 제거된 신뢰성 있는 핸드오프 기법을 제안하였다. 핸드오프 중 발생하는 패킷손실이 제거됨으로써 TCP 혼잡 제어 알고리즘이 패킷손실을 핸드오프로 인한 것이 아니라 네트워크의 혼잡에 의한 것이라고 오인하도록 함으로써 발생하는 TCP의 성능 저하를 피할 수 있는 큰 장점이 있으며 그 외의 장점은 다음과 같다. 우선 [4], [6]에서 제안한 에이전트 기반 경로 최적화 기법을 이용함으로써 Mobile IP의 투명성을 해치지 않으면서 확장성도 있는 장점을 그대로 유지했다. 그리고 그림 1에서 볼 수 있듯이 [3], [4]는 CN([4]는 대응 에이전트)이 MN의 새로운 위치정보를 알려주는 바인딩 업데이트 메시지를 수신하기까지 시간 및 컨트롤 메시지가 다소 소요됨을 알 수 있다. 즉, 우선 이전의 FA가 새 FA로부터 바인딩 업데이트 메시지를 받음으로써 MN의 이동을 알며 그 후에 MN을 향하는 패킷을 받은 이전의 FA가 HA에게 바인딩 워닝 메시지를 보냄으로써 CN([4]는 대응 에이전트)이 HA로부터 바인딩 업데이트 메시지를 받게 된다. 이에 비해 본 논문에서 제안한 기법은 HA가 등록 요청을 받은 후 바로 캐쉬 에이전트에게 바인딩 업데이트 메시지를 보냄으로써 더욱 간결하고 신속하게 캐쉬 에이전트가 MN의 새 위치정보를 얻을 수 있다. 또한 새 FA가 이전 FA에게 바인딩 업데이트 메시지를 보내는 것과 이전 FA가 바인딩 캐쉬를 관리하는 것 등의 기능 추가가 필요하지 않고 이에 따라 FA 없이도 본 기법의 구현이 가능하다. 추가적인 장점으로는 [3], [4]의 경우 MN과 이전 FA간의 security association이 요구되나 제안된 기법에서는 이를 필요로 하지 않는다는 점이다.
향후 과제로는 기존에 제시된 기법과 본 논문에서 제안된 기법을 비교하여 성능을 측정하는 시뮬레이션이 행해져야한다.

5. 참고 문헌

[1] C. Perkins, "IP Mobility Support for IPv4", RFC 3220, January 2002.
[2] C. Perkins, "IP Encapsulation Within IP", RFC 2003, October 1996.
[3] C. Perkins and D. Johnson, "Route Optimization in Mobile IP", draft-ietf-mobileip-optim-11.txt, September 2001.
[4] R. Vadali, J. Li, Y. Wu, and G.Cao, "Agent-Based Route Optimization for Mobile IP", IEEE Vehicular Technology Conference (VTC), Volume: 4, Page(s): 2731 -2735, Oct. 2001.
[5] J. Postel, "Internet Protocol, DARPA Internet Program Protocol Specification", RFC 791, September 1981
[6] A. Myles, D. B. Johnson, and C. Perkins, "A Mobile Host Protocol Supporting Route Optimization and Authentication", IEEE Journal on Selected Areas in Communications, vol. 13, pp. 839-849, June 1995.