# 무선 이동 환경을 지원하는 분산 구조

박 기 현

위덕대학교 정보통신공학과

# A Distributed System Architecture for Supporting Wireless Mobile Environments

Kihyun Park

Dept. of Information and Communication Eng., Uiduk University

요약

본 논문은 기존의 유선 환경에서의 서비스 환경을 이동 환경에서도 지원하기 위한 분산 구조를 다루고 있다. 이를 위하여 이동 단말기를 논리적으로 대신하는 이동 AP의 위치를 추적하는 추적 AP를 두어 기존의 유선 응용 환경에서 수행되던 응용 프로그램들을 무선 이동 환경에서도 수용할 수 있도록 하였다.이동 단말기의 위치 변경에 따른 이동 AP, 추적 AP사이에서 발생되는 핸드오프 처리 알고리즘이 소개되었으며, 이 알고리즘은 패킷 손실이나 순서의 변경과 같은 흐름제어 기능을 포함하고 있다. 또한 무선 환경에서 발생 가능한 하드 핸드오프와 소프트 핸드오프를 함께 고려하였다. 시뮬레이션의 결과는 핸드오프 과정에서의 추가 지연 시간에 대한 내용을 다루고 있다.

## 1. 서론

이동 AP (M-AP: Mobile Agent Process)는 MS (MS: Mobile Station)가 유선망에 위치하는 응용 서버 프로그램 (SO: Service Object)으로 부터 응용 서비스를 받을 때 MS를 논리적으로 대신하는 프로세스이다. 이동 AP는 MS와 SO의 사이에서 연결 설정 유지, QoS 관리, MS의 처리 능력 향상과 같은 부가적인 기능을 수행한다. 무선 MS가 이동하게 되면 이동 AP도 함께 새로운 위치로 이동하게 된다.

MS가 이동함에 따라서 함께 이동하는 이동 AP의 위치를 SO에서는 계속 추적할 수 있어야 한다. 따라서 기존의 유선 환경에서 수행되던 SO들이 이동 환경을 지원하기 위해서는 이동 AP 위치 추적 기능이 추가되어야 한다. 본 논문에서는 이동 환경에서도 기존의 응용 프로그램들이 그대로 수용될 수 있는 방안에 관한 내용을 다룬다. 이를 위하여 이동 AP의 위치를 추적하는 추적 AP (T-AP: Trace Agent Process)를 두도록 하였다. 그림 1에서 MS와 이동 AP는 그 위치를 변경할 수 있지만 SO와 추적 AP는 유선 환경에서 고정된 위치를 갖는다.

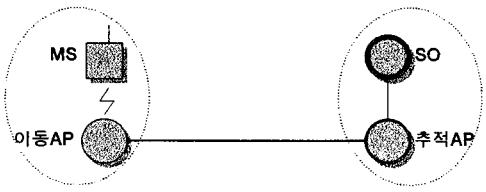


그림 1. 이동 AP의 위치 추적 기능을 갖는 추적 AP

추적 AP는 이동 AP의 현재 위치에 대한 정보를 가지고 있으며, 핸드오프 현상이 발생하는 경우에도 SO와 계속적으로 연결을 유지할 수 있도록 도와 준다. 서비스 진행 중에 MS가 이동되면서 에러 율이나 전송 대역폭이 이전 위치와 다르게 지원되어야 하는 경우에는 QoS에 대한 재협상이 필요한데, 이동 AP에서 하는 것보다는 거리가 짧으므로 빠른 협상이 이루어질 수 있다.

에이전트 구조를 지원하기 위하여 Mobile Domain을 정의하였으며, Mobile Domain은 DPE(Distributed Processing Environment)와 같은 분산 환경을 지원한다. 시스템내에는 다수의 Mobile Domain이 존재하며, 각 Mobile Domain의 관리 영역은 서버의 관리 영역 기준에 의하여 이루어진다. Mobile Domain은 다시 다수의 블럭 구조로 구성이 된다. 블록 구조는 에이전트 프로세스를 효율적으로 관리하기 위한 단위로써 두 종류가 있다. 첫째는 임의의 SO와 관련된 추적들을 관리하는 추적 AP 블록으로, 시스템에 존재하는 SO의 개수만큼 존재한다. MS가 접속 가능한 지역에서는 이동 AP를 관리하기 위한 이동 AP 블럭이 존재한다. 예를 들어서 하나의 MSC (Mobile Switching Center) 에 대하여 다수의 MS가 접속 가능하며 이들을 관리하기 위해서 하나의 이동 AP 블럭이 할당될 수가 있다. 블록 구조내의 AP에 대한 관리는 해당 블록내에 존재하는 블록 관리자가 수행한다. 한편 핸드오프의 발생은 Mobile Domain 영역 내에서 뿐만이 아니라, 도메인 영역간에도 발생될 수 있다.

AP의 이동성을 지원하기 위하여 각각의 Mobile Domain 내에는 여러 종류의 서버 프로세스들이 존재한다. 각 도메인에 있는 서버들은 필요에 따라서 정보를 주고 받을 수 있는 연합 구조를 갖게 된다. 위치 서버 (LS: Location Server)는 MS의 위치 정보를 관리하며, 트레이더(Trader)는 분산 환경에서 트레이딩 기능(Trading Function)을 지원하는 객체이다. 트레이더는 자신의 도메인에 속한 SO의 위치 정보를 제공한다. 저장 서버 (RS: Repository Server)의 역할은 MS의 단말기 타입에 대한 정보를 포함하여 에이전트 프로세스의 컨텍스트에 관한 정보를 보관하는 역할을 한다.

이동 단말기의 이동에 따라서 에이전트 프로세스도 이동할 것인지, 아닌지의 여부는 응용 환경의 특성에 영향을 받는다. 고정 위치 형태의 구조에서는 에이전트 역할을 수행하는 서버가 데이타의 경로를 적절한 위치로 라우팅을 해주며, 응용 환경을 위한 대역폭의 관리, 이동 단말기의 위치 추적과 같은 기능을 담당한다. 에이전트 프로세스를 고정시키는 구조에서는 서비스 영역이 제한된 지역에서만 동작하도록 가정되어 있다. 영역이 커지게 되면 상대적으로 열악한 무선 시스템의 영향이 커지게 되어 지연이 증가하게 되므로 에이전트의 효과가 상쇄되는 것으로 알려져 있다. 이동하는 구조에서는 프로세스 이동 메커니즘이 성능에 중요한 영향을 미치며, 기존의 서비스 객체들이 에이전트 프로세스의 이동 위치에 대한 정보를 계속 추적해야 한다는 부담을 갖게 된다.

핸드오프의 실행은 크게 Hard 핸드오프와 Soft 핸드오프로 구분할 수 있다. Hard 핸드오프는 MS가 셀 위치를 변경하는 경우에 이전 기지국과의 연결 설정을 먼저 해제하고 새로운 기지국과의 연결 설정을 하게 된다. MS와 기지국 사이의 통신 경로는 하나만 존재하게 되므로 핸드오프 과정 중에 MS와 망 사이에는 순간적인 호 절단 현상이 발생하게 되는 단점을 가지고 있다. Soft 핸드오프는 순간적으로 두개 이상의 연결 설정이 MS와 기지국 사이에서 발생할 수 있다. 따라서 핸드오프 과정 중에 두개의 서로 다른 경로를 통하여 정보들이 움직일 수 있다.

## 2. 핸드오프 알고리즘

MS가 시스템에 접속되는 과정은 크게 두 단계로 이루어진다. 첫번째는 MS의 존재가 망에 등록되는 위치 등록 과정이고, 두 번째는 등록된 MS가 서비스 요구를 하는 경우이다. 위치 등록이라 함은 MS가 자신의 존재를 망에 알리는 절차이며, MS가 등록되는 지역을 관장하는 BM에게 통보가 된다. 이 정보를 이용하여 망에서는 각각의 MS의 위치를 유지, 관리할 수가 있다. MS의 위치 등록이 완료되면 MS는 서비스 초기화 요구를 실행할 수가 있다. 이때 트레이더가 원하는 서비스의 위치 정보를 가지고 있다.

핸드오프 처리를 위해서는 핸드오프 초기화 단계에서 새로운 위치에서의 이동 AP 생성, 각 객체에게 핸드오프 발생 통보, 적절한 연결 재설정 등이 필요하다 핸드오프 과정에서 패킷 전달 경로의 순간적인 변경에 따라 원래 위치의 이동 AP 버퍼에 보관된 패킷이 손실되거나, 그 처리가 늦어지게 되어 패킷들의 전달 순서가 변경되는 현상이 발생될 수 있으므로 이에 대한 고려가 필요한다. 처리 과정은 크게 두가지 경우가 가능한데, 하나는 핸드오프 과정에서 이전 경로로의 패킷 전달이 가능한 역방향 핸드오프 실행 방식이고, 다른 하나는 이전 경로를 이용한 패킷 전달이 불가능한 경우를 가정한 순방향 핸드오프 실행 방식이다.

역방향 핸드오프 실행 방식에서는 MS가 현재의 경로를 이용하여 핸드오프의 필요성을 망에 알리고자 하는 경우, 혹은 망에서 핸드오프의 실행을 결정하는 경우 등과 같이 핸드오프에 관련된 정보의 전달이 이전 경로를 이용할 수 있는 경우에 가능하다. 이러한 정보들의 전달이 성공적으로 완료되면 MS는 새로운 기지국으로 연결 설정을 전환하게 된다. 순방향 핸드오프 방식은 MS와 이전 기지국 사이의 연결 설정이 갑자기 끊어지게 되어, 이 경로를 사용한 패킷 전달이 불가능한 경우에 이용된다. 따라서 핸드오프와 관련된 시그날링은 새로운 기지국과의 연결 경로를 통하여 이루어진다

## 3. 성능 분석

시뮬레이션 프로그램은 추적 AP, 무선 단말기, 원래 위치의 이동 AP (OM-AP), 새로운 위치의 이동 AP (NM-AP) 등의 패킷 전송 방식으로 구성된다. 패킷 전송 시스템은 가상 네트워크인 VM(Virtual Media) 에 의하여 구현이 되며, VM 간의 패킷 전송은 UNIX socket 을 이용하였다. 가상 네트워크의 역할은 객체간의 패킷을 전송하는데 걸리는 시간을 조정하기 위한 것이다. 즉, 네 개의 객체에서 전송된 패킷은 일단 각 개체의 VM으로 전달이 되며, VM에서는 이 패킷들을 보관하고 있다가 정해진 시간이 경과하면 수신자 객체에게 전달하게 된다.

그림 2는 각 객체들간의 패킷 전송 시간을 보이고 있다. 예를 들어서 OM-AP 와 T-AP 사이의 패킷 전달 시간은 a 만큼 걸리고, OM-AP 와 NM-AP 사이의 전달 시간은 c 만큼 걸리게 됨을 보이고 있다. I 의 의미는 T-AP 와 MS 에서의 패킷 발신 속도를 의미하는 것으로 전달되는 패킷 사이의 시간 간격을 의미한다. 즉, T-AP 와 MS 에서는 일정한 전송 율의 패킷들이 발신된다고 가정한다. 실험은 UNIX 시스템에서 이루어졌으며, 네 개의 객체들이 각각 하나씩의 프로세스로서 동작하도록 하였다.

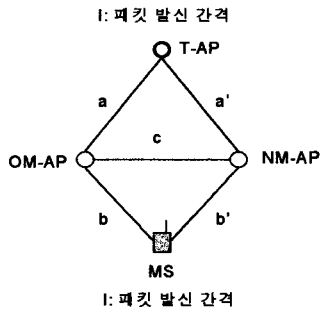


그림 2. 객체들간의 패킷 전송 시간

역방향 흐름 제어 방식에서는 OM-AP 와 MS 사이의 이전 경로가 존재하므로 핸드오프 처리 과정 중에도 패킷 전송이 계속해서 이루어질 수가 있는데, 식 1 과 2 는 각각 하향 패킷과 상향 패킷에서의 추가 지연 시간을 나타낸다. 이때 a=a', b=b' 로 가정한다.

$$T_{back\_down} = \begin{cases} c & (단, a + c > a') \\ 0 & (단, a + c < a') \end{cases} \quad (1)$$

$$T_{back\_up} = \begin{cases} c & (단, b + c > b') \\ 0 & (단, b + c < b') \end{cases} \quad (2)$$

순방향 핸드오프에서의 하향 패킷의 추가 지연 시간도 하향의 경우는 식 3, 상향의 경우는 식 4 로 주어진다

$$T_{forw\_down} = 2a + b + 9c \quad (3)$$

$$T_{forw\_up} = 3b + 9c \quad (4)$$

## 4. 실험 결과

그림 3 은 실험에서 얻어진 결과 그래프에 대한 이해를 돕기 위한 것으로 긴 점선으로 표시된 부분이 핸드오프가 발생하지 않았을 때의 패킷 도착 시간을 의미하며, 실선으로 표시된 부분이 핸드오프가 발생하였을 때의 도착 시간을 의미한다. Y 축의 시간 값은 첫번째 패킷의 도착 시간을 0 으로 하여 이 패킷과의 상대적인 시간 차이를 의미한다. i 번 패킷과 (i + 1) 번 패킷 사이의 핸드오프에 따른 추가 지연 시간이 그림에 표시되어 있으며, 그 이후의 (j - 1) 개 패킷의 경우는 (i + 1) 번 패킷과 거의 동일한 시간에 도착하게 된다.

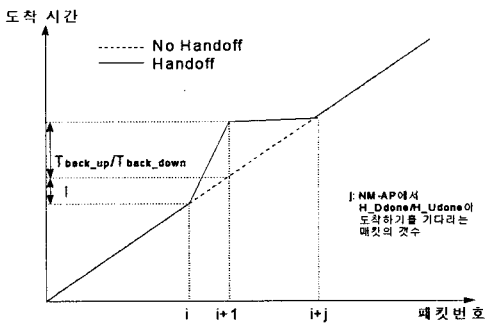


그림 3. 그래프의 의미

실험 결과는 역방향 핸드오프에 대해서 수식과의 비교를 하였다. 그림 4부터 7까지의 결과에는 세가지 종류의 값이 존재하는데, 먼저 점선으로 표시되는 값은 핸드오프가 발생하지 않는 경우의 도착 시간을 의미한다. 사각형이 표시된 실선은 T-AP 에서 수신한 패킷들의 도착 시간을 의미하고, 플러스 표시와 함께 표시된 실선은 MS 에서 수신한 패킷들의 도착 시간을 의미한다. 그림 5와 그림 6은 각각 a 와 b 값을 크게 하여 얻은 결과로써 추가 지연 시간이 그림 3과 동일함을 알 수 있고, 그림 7은 c 값을 크게 하여 얻은 결과로써 두 패킷 사이에는 추가적인 지연이 발생됨을 알 수 있다. 따라서 c 값의 변화가 지연 시간에 영향을 준다는 것을 실험을 통하여 확인할 수가 있다.

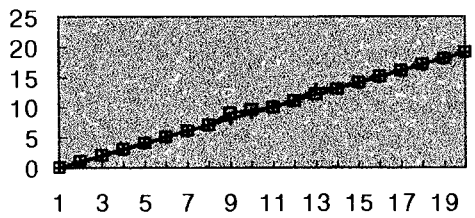


그림 4. 기준 데이터 (I=100ms, a=500ms, b=100ms, c=100ms)

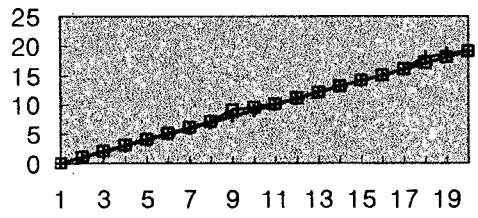


그림 5. a 의 영향 (I=100ms, a=1000ms, b=100ms, c=100ms)

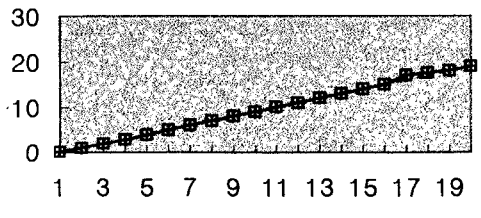


그림 6. b 의 영향 (I=100ms, a=500ms, b=500ms, c=100ms)

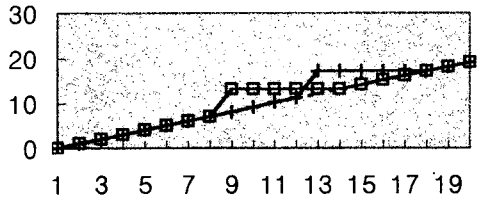


그림 7. c 의 영향 (I=100ms, a=500ms, b=100ms, c=500ms)

## 5. 결론

유/무선 사이의 신뢰성과 대역폭의 차이를 극복하기 위한 이동 에이전트 모델은 둘 사이의 불균형을 해소할 수 있는 하나의 해결책으로 연구되고 있다. 본 논문에서 제안한 구조는 서비스 영역의 제한을 받지 않으면서도 무선 단말기의 이동이 서비스 객체에게 투명하게 보이도록 하여 기존의 응용 환경을 이동 환경에서도 그대로 수용할 수 있다는 장점을 가지고 있다. 시스템 구조는 확장성이 용이하도록 설계되었으며, 이러한 확장성을 효과적으로 지원하기 위한 서버들을 정의하였다. 제시된 모델은 TINA 구조와 같이 기존에 연구된 분산 모델과 쉽게 통합될 수 있도록 기능에 대한 제약을 최소화하여 설계되었다. 즉, 제안 모델의 또 다른 장점은 특정한 시스템 환경을 가정하지 않고, 다양한 환경에 적용이 가능하도록 모델 설정을 했다는 점이다.

## 6. 참고 문헌

[1] Ramachandran Ramjee *et al.*, "The Use of Network-Based Migrating User Agents for Personal Communication Services", IEEE Personal Commun., Dec. 1995

[2] 박기현, 안순신, "이동 에이전트 위치 추적 객체를 이용한 이동 컴퓨팅 시스템", 정보과학회 논문지(A), 제 26권 제 4호, 1999

[3] Kristian Rauhala, "An Introduction to Wireless ATM", MoMuC'97 Tutorial T-2, 1997

[4] George Liu *et al*, "A Mobile Virtual-distributed System Architecture for Supporting Wireless Mobile Computing and Communications", Mobicom'95, 1995

[5] Seshadri Mohan, Ravi Jain, "Two User Location Strategies for Personal Communications Services", IEEE Personal Commun., First Quarter 1994

[6] Randy H. Katz, "Adaptation and Mobility in Wireless Information Systems", IEEE Personal Commun., First Quarter 1994

[7] Larry T.Chen, Tatsuya Suda, "Designing Mobile Computing Systems using Distributed Objects", IEEE Commun. Mag., 1997

[8] Yao-Nan Lien, Chun-Wu R.Leng, "On the Search of Mobile Agents", PIMRC'96, 1996

[9] Tomasz Imielinski, Henry F. Korth, "Mobile Computing", Kluwer Academic Publishers, 1996

[10] 박기현,안순신, "이동성을 지원하는 분산 컴퓨팅 구조와 알고리즘", 정보과학회 논문지 (A), 제 25권 제 5호. 1998