

분산 HLR 환경에서 분기포인터를 이용한 위치 관리 기법

(Location Management Scheme by using Jumping Pointer in Distributed HLR Environment)

이 보 경 [†] 황 종 선 ^{**}
(Bo-Kyung Lee) (Chong-Sun Hwang)

요 약 본 논문에서는 분산 HLR 환경에서 이동 단말의 위치 추적을 위한 비용을 줄이기 위한 분기포인터 기법(Jumping Pointer Scheme)을 제시한다. 이 기법은 분산 HLR 환경에서 사용되는 포인터 포워딩 기법(Pointer Forwarding Scheme)의 변형으로, 최근에 갱신된 HLR이 아닌 다른 HLR로 부터 착신호가 유발되어 질 경우 포인터 포워딩 기법에서는 착신 이동 단말을 찾기 위해서 경유해야 하는 VLR의 개수가 많아질 수 있는데 이러한 현상을 막고자 최근에 갱신된 VLR과 현재 이동 단말이 위치하고 있는 등록 영역의 VLR간에 분기포인터(Jumping Pointer)로 연결하여 위치 추적을 보다 빠르고 신속하게 하고자 한다. 본 논문에서는 분기포인터를 사용할 경우 위치 추적을 위한 "find" 비용과 HLR 및 VLR 데이터베이스의 자료 갱신을 위한 "update" 비용 관점에서 수학적 모델을 제시하고 이에 대한 성능을 평가한다. 또한 기존의 포인터 포워딩 기법의 수학적 모델을 이용하여 두 기법간의 성능을 비교 분석한다. 분기포인터를 이용한 위치 관리 기법은 착신호와 등록 영역을 이동하는 횟수에 대한 비율이 상대적으로 낮은 경우 즉 호가 자주 발생하지는 않으나 등록 영역간 이동이 많은 단말의 경우 포인터 포워딩 기법 대비 약 17%의 성능 향상을 나타낸다.

Abstract In distributed HLR environment, the pointer forwarding with distributed HLR(Pointer Forwarding Scheme) has been proposed for eliminating the cost of updating multiple HLRs. In Pointer Forwarding Scheme, the find cost can be changed according to from which HLR a call originates. If a user changes PCS registration areas frequently, but receives calls relatively infrequently, and call for such a user originates from any other HLRs than the recently updated HLR, many VLRs should be traversed in order to find that user. In this paper, location tracking with jumping pointer in distributed HLR(Jumping Pointer Scheme) is proposed. The jumping pointer can be connected between the recently updated VLR and the VLR in which the mobile host currently resides. In case of call delivery, the mobile host can be found sooner by using the jumping pointer. In terms of "find" cost, Jumping Pointer scheme improves about 17% over Forwarding Pointer scheme. Otherwise, Jumping Pointer scheme takes additional update charge in order to manage the jumping pointer. However the total costs of "find" cost and "update" cost for Jumping Pointer scheme are cheaper than them for Forwarding Pointer scheme when call-to-mobility ratio is smaller.

1. 서 론

PCS(Personal Communication Services)에서 이동성(Mobility)을 지원하기 위하여 EIA/TIA IS-41과 ETSI GSM MAP 등의 프로토콜이 제안되었다. 이들 프로토콜에서는 이동성 관리를 위한 위치 추적(location tracking) 및 등록(registration) 등의 2가지 기능을 수행한다. 그러나 이러한 기능을 수행하기 위해서는 많은 비용을 부담해야하기 때문에 이들 비용을 줄이기 위한

[†] 비 회 원 : 고려대학교 컴퓨터학과
mkjung@chollian.net

^{**} 종 신 회 원 : 고려대학교 컴퓨터학과 교수
hwang@disys.korea.ac.kr

논문접수 : 1998년 9월 14일
심사완료 : 2000년 1월 13일

노력들이 많이 연구되고 있다. IS-41과 GSM 방식 모두 HLR(Home location register) 및 VLR(Visitor location register)의 두 계층(two-tier) 구조를 갖는다. IS-41 및 GSM 두 방식에서는 다음의 절차들을 수행함으로써 이동성을 지원하게 된다. 만일 이동 단말이 새로운 등록 영역(Registration Area)을 방문하게 되면 등록 영역의 VLR 데이터베이스에 등록 작업을 하게된다. VLR은 등록된 메시지를 HLR에 보내 데이터베이스의 정보를 갱신하게 된다. 이동 단말에 대한 착신호가 발생되면 HLR은 VLR에 착신 단말의 정보를 보내 이동 단말의 위치를 추적하게 된다. 이동 단말의 위치를 발견하면 비로소 호의 전달이 이루어 지게된다. 이러한 일련의 작업을 수행하기 위하여 많은 비용이 부담되어진다. 이러한 비용을 위치 관리 비용이라 하며, 이 위치 관리 비용을 줄이기 위한 연구가 많이 진행되고 있다.[1]-[6]

특히 빈번하게 등록 영역을 이동하는 단말의 경우, 쓸모없는 VLR 및 HLR 데이터베이스에 등록 작업을 수행해야 한다. 이러한 문제점을 보완하고자 포인터 포워딩 기법(Pointer Forwarding Scheme)[2]이 제안되었다. 이는 등록 영역을 이동할 때, HLR의 데이터 베이스는 갱신하지 않고 VLR 간에 포워딩 포인터로 연결하는 기법이다. 만일 착신호가 발생되면 이동 단말이 등록되어 있는 HLR에서 VLR로 이동 단말의 위치 파악을 위한 메시지를 보낸다. 만일 그 VLR 데이터베이스에서 이동 단말의 정보를 찾지 못하면 포워딩 포인터를 따라 추적을 하게되어 이동 단말의 위치를 파악할 수 있게 된다. 이렇게 찾은 이동 단말의 위치를 HLR에 알려주게 되며 발신자와의 호 접속이 이루어지게 된다. 이때 비로소 HLR 데이터베이스의 정보는 갱신된다.

그러나 단일 HLR 환경에서는 오로지 하나밖에 존재하지 않는 HLR로 모든 트래픽이 집중되므로 HLR의 병목 현상은 불가피한 것이다. HLR을 분산시킴으로써 이러한 병목 현상을 막기 위한 분산 HLR 기법이 도입되었으며, 분산 HLR 환경에서 포인터 포워딩 기법을 적용하는 방안[6]이 제안되었다. 분산 HLR 환경에서의 포인터 포워딩 기법은 이동 단말의 등록 영역간 이동이 빈번하게 발생하고 상대적으로 착신호의 발생은 적은 경우 상당한 효율을 나타낸다. 그러나 분산 HLR 환경에서 포인터 포워딩 기법은 착신호가 여러 지역의 HLR로 부터 유발되는 경우, 특히 최근에 갱신된 HLR이 아닌 다른 HLR로부터 호가 발생되는 경우 이동 단말의 위치를 파악하기 위하여 포워딩 포인터로 연결된 수많은 VLR을 추적해야 하는 문제가 발생될 수 있다. 이러한 문제점을 해결하기 위하여 본 논문에서는 최근에 갱

신된 VLR과 현재 이동 단말이 위치하고 있는 VLR간에 분기포인트(Jumping Pointer)라는 개념을 제안하여 두 VLR간 연결을 하게 된다. 분기포인트를 두어 두 VLR간 직접 연결되어지므로 호의 전달 시 보다 빠르게 접속할 수 있도록 한다. 본 논문에서는 기존의 포인터 포워딩 기법에서 제안된 수학적 모델을 활용하여, 분기포인트 기법의 위치 추적 시 부담되는 "find"비용과 등록 작업 시 부담되는 "update" 비용에 대한 수학적 모델을 설정한다. 또한 포인터 포워딩 기법 및 분기포인트 기법의 수학적 모델을 이용하여 두 기법 간의 성능을 비교 분석한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구로서 [6]에서 제안된 포인터 포워딩 기법을 자세히 소개하며, 제 3장에서는 본 논문에서 제안된 분산 HLR 환경에서의 분기포인트 기법에 대하여 설명한다. 제 4장에서는 포인터 포워딩 기법과 분기포인트 기법에 대한 "find" 비용 및 "update" 비용에 대한 수학적 모델을 설정한다. 제 5장에서는 제 4장에서 설정된 수학적 모델을 근거로 하여 두 기법간의 성능을 분석하고 평가한다. 마지막으로 제 6장에서는 본 논문의 결론으로 성능 분석의 결과 및 향후 연구 과제 등을 제시한다.

2. 관련 연구

본 장에서는 분산 HLR 환경에서 이동 단말의 위치 추적을 위한 비용인 "find" 비용을 줄이기 위한 기법중의 하나인 분산 HLR 환경에서의 포인터 포워딩 기법[6]에 대하여 소개한다. 포인터 포워딩 기법은 원래 단일 HLR 환경의 PCS 네트워크에서 HLR의 위치 갱신 비용을 줄이기 위하여 제안되었다[2]. 단일 HLR 환경에서의 포인터 포워딩 기법에서, 이동 단말이 새로운 등록 영역(Registration Area)으로 이동할 경우, HLR 데이터베이스의 내용은 갱신되지 않고, 대신 새로운 등록 영역에서 기존의 등록 영역으로 단말의 이동을 알리는 메시지를 보낸다. 이 메시지를 받은 기존 등록 영역의 VLR은 VLR 데이터베이스에 있는 이동 단말에 대한 등록 자료를 삭제하고 새로운 등록 영역의 VLR로 포워딩 포인터를 설정한다. 만일 착신호가 발생되면, VLR내에 설정된 포워딩 포인터를 따라 이동 단말의 실제 위치를 찾기 위하여 포워딩 포인터가 추적된다. 단일 HLR 환경에서의 포인터 포워딩 기법은 위치 갱신을 위한 비용은 절감되었으나, HLR이 하나 존재하므로 HLR의 병목 현상은 여전히 문제로 남아 있다. 즉, 단말의 위치를 갱신하기 위하여 단일의 HLR로 모든 트래픽이 집중될 수 있다. 이러한 문제점을 해결하기 위하여 HLR을 여

러 곳에 분산시키는 방안이 제시되었는데 이것이 분산 HLR 환경에서의 포인터 포워딩 기법이다.[6] 이 기법에서 HLR은 원격의 PSTN(Public Switched Telephone Network)에 분산되어 있으며, 분산 HLR들은 STP(Signal Transfer Point)와 근접해 있다. STP는 GTT(Global Title Translation)라 불리는 테이블 조사(Table Lookup) 기법을 이용하여 이동 단말의 위치 정보를 가지고 있는 HLR 주소를 인식하게 된다. 각각의 HLR은 이동 단말이 속해 있거나 또는 이전에 방문하였던 VLR을 가리키고 있다. 그림 1은 분산 HLR 환경에서의 포인터 포워딩 기법을 나타낸다.

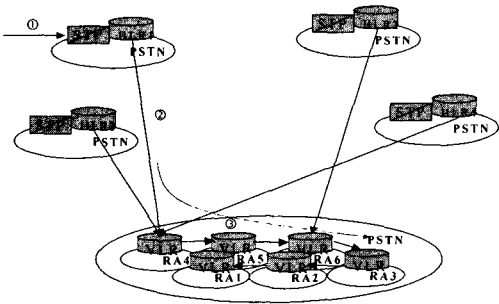


그림 1 분산 HLR환경에서의 포인터 포워딩 기법

이동 단말이 등록 영역을 이동할 때는 기존의 VLR과 새로운 VLR간에는 포워딩 포인터만을 유지하고 HLR의 데이터베이스는 갱신되지 않는다. 그러나 착신호가 도착되었을 경우, PSTN의 HLR이 조회되며, "find" 명령을 통하여 HLR이 가리키고 VLR을 검색한다. 만일 그 등록 영역에 이동 단말이 없으면 VLR간의 포워딩 포인터를 추적하여 현재 이동 단말이 있는 등록 영역을 찾게된다. 그래서 착신 단말의 위치가 발견되어 지며, 호의 접속이 성립된다. 그 후, 이동 단말은 자신이 현재 머무르고 있는 등록 영역을 HLR에 가르켜 주면, HLR은 현재 이동 단말이 위치하고 있는 등록 영역의 VLR을 가리키도록 데이터베이스를 갱신하게 된다. 즉 착신호가 발생될 때, 포워딩 포인터를 이용하여 이동 단말이 위치하고 있는 등록 영역을 찾게되며, 분산 HLR은 현재 이동 단말이 위치하고 있는 등록 영역의 VLR을 가리키도록 데이터베이스를 갱신하게 된다. 포인터 포워딩 기법은 등록 영역간 이동이 빈번히 발생하나, 상대적으로 착신호가 적게 발생할 때 효율적이다. 즉, 착신호 발생 빈도수 대비 등록 영역간 이동 횟수의 비율이 적은 경우 효율적이다. 그러나, 포인터 포워딩 기법의 경우,

이동 단말로의 착신호가 여러 지역에서 발생하는 경우 다시 말하면 최근에 갱신된 HLR이 아닌 다른 HLR로부터 호 접속 요구가 발생되면, VLR간의 포워딩 포인터의 연결이 길어지므로 이동 단말을 찾기 위한 시간 및 비용이 많이 소요된다.

3. 분산 HLR 환경에서 분기포인터 기법

분산 HLR 환경에서의 포인터 포워딩 기법은 이동 단말의 등록 영역간 이동이 많고 상대적으로 착신호가 적게 발생하는 경우 성능의 향상을 나타낸다. 그러나 이동 단말로의 착신호가 여러 지역에서 발생하는 경우, 즉, 최근에 갱신된 HLR이 아닌 다른 HLR로 부터 착신호가 발생되면, VLR간의 포워딩 포인터의 연결이 길어지므로 이동 단말을 찾기 위한 시간 및 비용이 많이 들게 된다. 분산 HLR 환경에서의 포인터 포워딩 기법의 이러한 문제점을 해결하고자 본 논문에서는 분기포인터 기법(Jumping Pointer Scheme)을 제안하고자 한다. 분기포인터 기법은 포인터 포워딩 기법과 유사하나, 착신호 도착 시 등록 작업에서 차이를 나타낸다. 착신호가 도착되면 "find" 명령을 통하여 접속을 위한 이동 단말의 위치가 찾아지게 되며, HLR은 이동 단말이 현재 속해 있는 등록 영역의 VLR로 포인터를 변경하게 된다. 여기 까지의 절차는 포인터 포워딩 기법과 동일하다. 그러나 분기포인터 기법에서는 이동 단말이 현재 속한 등록 영역의 VLR과 이전의 VLR간에 분기포인터로 연결되며, 포워딩 포인터는 삭제된다. 그 다음 최근에 갱신되지 않은 또 다른 HLR로 부터 착신호가 발생되면 분기포인터를 통하여 이동 단말의 위치가 추적되게 된다. 그림 2는 분산 HLR 환경에서의 분기 포인터를 보여준다.

분산 HLR 환경에서의 분기포인터 기법은 초기 상태, 포워딩 포인터 생성단계, 착신호 발생 단계, 분기포인터

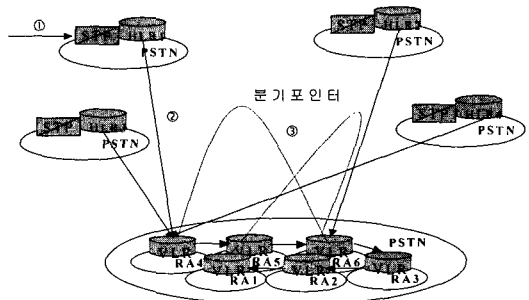


그림 2 분기포인터

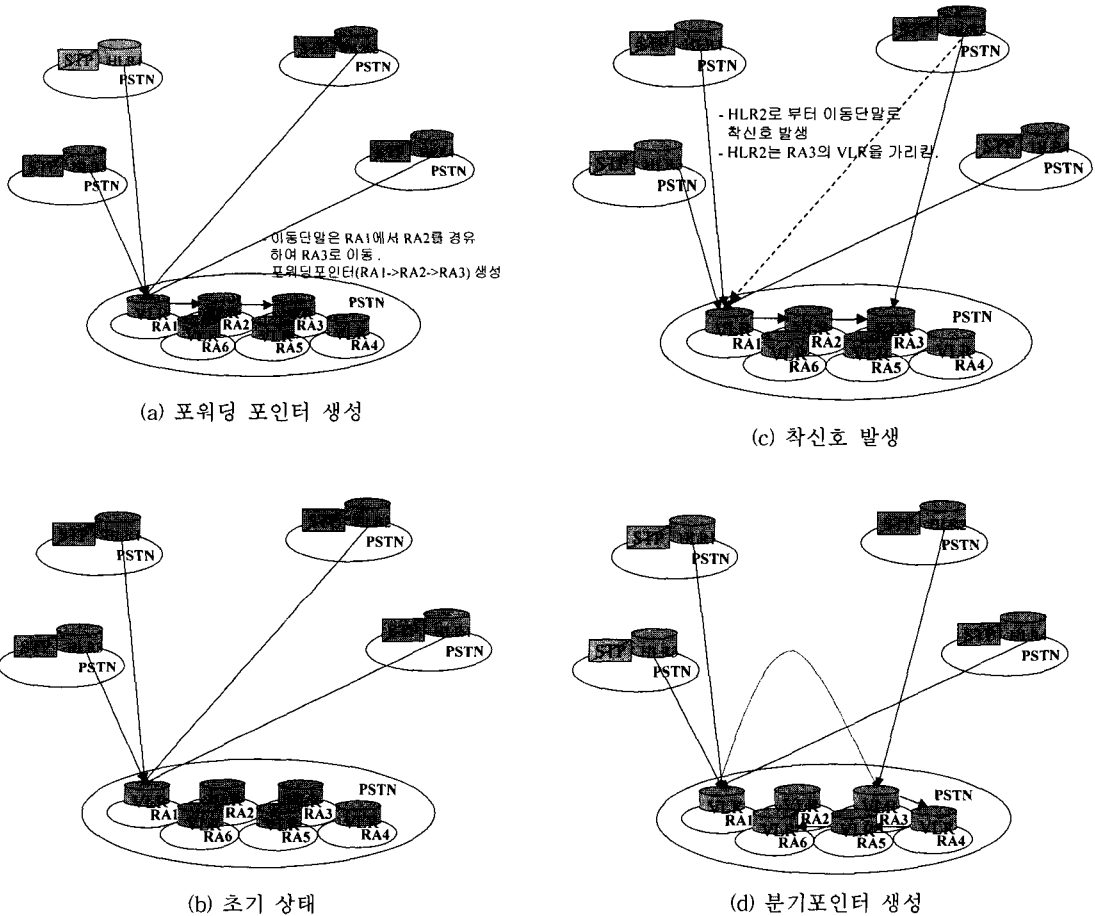


그림 3 분산 HLR 환경에서 분기포인트 기법 수행 절차

생성 등의 4가지의 절차로 구성된다. 그림 3에서는 분기포인트 기법에 대한 수행 절차를 나타낸다. 이동 단말이 등록 영역 1에서 등록 영역 2를 경유하여 현재 등록 영역 3에 머무르고 있으며, 이때 HLR 2가 속한 PSTN의 가입자가 이동 단말과의 접속을 시도한다고 가정하자. 초기상태에서 모든 HLR은 등록 영역 1의 VLR을 가리키고 있다고 하자. 포워딩 포인트 생성 단계에서는 이동 단말이 등록 영역을 이동함에 따라 등록 영역내의 VLR 간 포워딩 포인트가 생성된다.

PSTN 가입자의 호 접속 요구는 HLR을 통하여 이동 단말이 최근에 등록한 VLR을 찾으며 VLR의 포워딩 포인트 생성 단계에서 생성된 포워딩 포인트를 따라서 이동 단말이 현재 머무르고 있는 등록 영역 3에 까지

도달하게 된다. 이러한 과정을 통하여 “find” 작업이 수행되며, 이동 단말을 찾으면 HLR 2는 이동 단말이 현재 머무르고 있는 등록 영역 3의 VLR을 가리키도록 데이터베이스 정보를 갱신한다. 반면 등록 영역 1의 VLR은 분기포인트를 사용하여 곧바로 등록 영역 3의 VLR을 가리키며, 등록 영역 1의 VLR은 등록 영역3과의 포워딩 포인트 설정을 해제한다. 그후 최근에 갱신된 HLR인 HLR2가 아닌 다른 HLR로 부터 착신호가 발생되면, 이 분기포인트를 이용하여 이동 단말을 보다 신속하게 찾을 수가 있다. 또한 이동 단말이 등록 영역 4을 거쳐 등록 영역 5로 이동하였으며, 이동 단말과의 접속을 원하는 착신호가 HLR4가 속한 PSTN 가입자로부터 발생한다고 할 경우, “find”를 위한 경로는 HLR4가 가

리키고 있는 VLR1으로 메시지를 보내며 VLR1과 VLR3간의 분기포인트를 이용하여 등록 영역 5에 있는 이동 단말을 찾을 수 있다. 이때 이동 단말을 찾기 위한 경로는 HLR4 -> 등록 영역 1 -> 등록 영역 3 -> 등록 영역 4 -> 등록 영역 5이다. 반면, 분기포인트를 사용하지 않을 경우 "find"의 경로는 HLR4가 가리키고 있는 등록 영역 1로 가서 포워딩 포인트의 연결 고리를 찾아 등록 영역 2, 등록 영역 3, 등록 영역 4을 거쳐 등록 영역 5에 있는 이동 단말을 찾게된다. 이때의 이동 경로는 HLR4 -> 등록 영역 1 -> 등록 영역 2 -> 등록 영역 3-> 등록 영역 4 -> 등록 영역 5이다. 즉, 분기포인트를 사용 할 경우는 포워딩 포인트를 사용하는 것 보다 "find"를 위한 경로를 줄일 수 있으며, 이를 위한 비용 역시 줄일 수 있다.

4. 수학적 모델

본 장에서는 [6]에서 제시된 분산 HLR 환경에서의 포인트 포워딩 기법에 대한 "find" 비용과 "update" 비용 산정을 위한 수학적 모델을 기반으로 하여, 분기포인트 기법에 대한 "find" 비용 및 "update" 비용에 대한 수학적 모델을 제시하고자 한다.

4.1 Find 비용

"find" 비용은 호 접속 요구가 있을 때, 착신 단말이 어디에 위치해 있는지를 추적하는데 소요되는 비용으로, "find" 비용을 줄임으로써 보다 빠르고 신속하게 착신 단말과 접속할 수 있도록 해준다. 본 논문에서 고려한 환경은 분산 HLR 환경인데, 일반적으로 분산 HLR들은 GTT STP와 근접하여 있거나, 같은 곳에 위치된다고 할 수 있다. 수학적 모델 설정을 위하여 다음의 비용 추정값을 사용하며, 그림 4는 분산 HLR 환경에서의 포인트 포워딩 기법 및 분기포인트 기법에서 사용되는 비용 추정값을 보여준다.

- 가) 발신자로부터 분산 HLR까지 접속 요청을 위한 비용은 1로 일반화된다.
- 나) HLR로부터 HLR이 가리키고 있는 VLR을 조회하는 비용은 1이다.
- 다) 하나의 VLR로 부터 다른 VLR로 경유하는 비용, 즉 포워딩 포인트를 경유하는 비용은 δ 이며, $\delta \ll 1$ 로 기대되어 진다.
- 라) 하나의 VLR로 부터 분기포인트를 경유하는 비용은 δ' 이며, δ' 역시 $\delta \ll 1$ 으로 기대되어진다.

분기포인트 기법에서 "find" 비용을 구하기 위한 수학적 모델을 제시하고자 다음의 공식들이 정의된다. 2개의

연속된 착신호 발생 동안 등록 영역을 K 번 가로 지를 경우, 포워딩 포인트의 개수는 k 이며 포워딩 포인트의 개수 k 에 대한 기대값으로 정의되는 $E[k]$ 는 다음의 식으로 유도되어진다.[6]

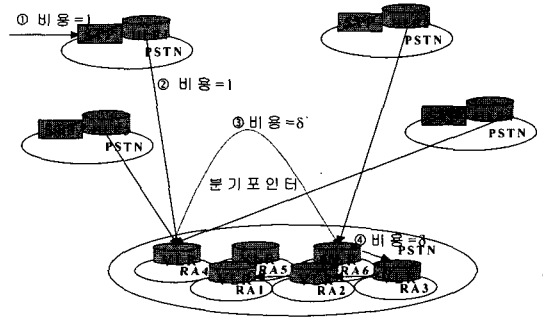


그림 4 비용 추정값

$$E[k] = \sum_{k=0}^{\infty} E[k | K] \alpha(K) \tag{1}$$

이때 k 는 K 와 항상 같다고 할 수 없으며, 일반적으로 $K \geq k$ 이다. $\alpha(K)$ 는 두 개의 연속된 착신호 사이에 등록 영역을 K 번 가로지를 확률을 말한다. $\alpha(K)$ 는 다음과 같이 정의된다.[2] [6]

$$\alpha(K) = \begin{cases} 1 - \frac{1 - f_m^*(\lambda_c)}{\theta} & K = 0 \\ \frac{1}{\theta} [1 - f_m^*(\lambda_c)]^2 [f_m^*(\lambda_c)]^{K-1} & K > 0 \end{cases} \tag{2}$$

$f_m^* \lambda_c$, λ_m 은 각각 호 도착률, 등록 영역을 가로지를 빈도수를 나타내며, $\theta = (\lambda_c / \lambda_m)$ 은 착신호 발생 빈도수 대비 등록 영역을 가로지를 횟수에 대한 비율 (Call-to-Mobility Ratio)로 정의된다.

f_m^* 은 이동 단말이 한 등록 영역에서 머무를 시간에 대한 Laplace Transform으로서, 등록 영역에서의 거주 시간은 Gamma 분포를 갖는다고 가정한다. 평균이 $1/\lambda_m$ 이고 분산이 V 인 Gamma Laplace-Stieltjes Transform은 다음과 같다.[2] [6]

$$f_m^*(s) = \left(\frac{\lambda_m \gamma}{s + \lambda_m \gamma} \right)^\gamma \quad \text{where } \gamma = \frac{1}{V \lambda_m^2} \tag{3}$$

두 연속된 착신호 간 포워딩 포인트의 개수에 대한 기대값을 나타내는 $E[k]$ 는 착신호 발생 빈도수 대비 등록 영역을 가로지를 횟수에 대한 비율인 θ 의 함수이며, 분산된 HLR로 부터 일정하게 (uniformly) 호가 발생된

다고 가정할 수 있다. 그러므로 분산 HLR 환경에서 포인트 포워딩 기법을 적용할 경우 "find" 비용은 식(4)와 같이 정의 된다.[6] 이때 N 은 분산 HLR의 개수를 말한다.

$$FindCost_{\text{포인트포워딩}} = 2 + E[k | \frac{\theta}{N}] \delta \quad (4)$$

본 논문에서는 최근에 갱신된 VLR을 가리키는 HLR이 아닌 다른 HLR에 접속된 가입자로 부터 호가 발생하는 경우에 대한 수학적 모델을 설정하고자 한다. $n+2$ 번의 착신호 발생 후 동일한 HLR에서 호가 발생하였다고 가정해 보자. 위의 식(4)는 두 연속된 착신호가 발생할 경우, 이동 단말을 찾기 위한 비용이므로 $n+2$ 번의 착신호 발생 후 이동 단말에 대한 "find" 비용은 식(5)로 제시될 수 있다.

$$FindCost_{\text{포인트포워딩}} = 2 + nE[k' | \frac{\theta}{N}] \delta + E[k'' | \frac{\theta}{N}] \delta \quad (5)$$

k' 은 임의의 i 번째 호 접속 요구가 들어온 HLR이 가리키고 있는 VLR에서 이동 단말이 $i+1$ 번째 호를 받은 후, 갱신된 HLR이 가리키고 있는 VLR까지의 포워딩 포인트 개수에 대한 기대값이다. 반면, k'' 은 이동 단말이 최근에 갱신한 HLR이 가리키고 있는 VLR로 부터 이동 단말이 등록 영역을 이동하였을 경우 포워딩 포인트의 개수에 대한 기대값을 말한다. $E[k']$ 과 $E[k'']$ 모두 두 연속된 착신호 사이의 포워딩 포인트에 대한 기대값이므로 결국 $E[k]$ 와 같다고 볼 수 있다. 그러므로 포워딩 포인트 기법에서 동일한 HLR로부터 호를 받을 경우 "find" 비용은 식(6)으로 유도될 수 있다.

$$FindCost_{\text{포인트포워딩}} = 2 + (n+1)E[k | \frac{\theta}{N}] \delta \quad (6)$$

분기포인트 기법에서의 "find" 비용에 대한 수학적 모델은 포인트 포워딩 기법에서 제시된 식(4)와 위에서 유도된 식(5),(6)을 기반으로 하여 설정될 수 있다. 이는 포인트 포워딩 기법과 유사하게, 호가 어떤 HLR에서 발생되느냐에 따라 다음과 같이 정의될 수 있다. 식 (7)은 최근에 갱신된 HLR에서 호가 발생된 경우 "find" 비용을 나타내는 것으로, 포인트 포워딩 기법과 동일하다.

$$FindCost_{\text{분기포인트}} = 2 + E[k | \frac{\theta}{N}] \delta \quad (7)$$

왜냐하면, 최근에 갱신된 HLR에서 호가 발생되므로 이동 단말의 위치를 파악하기 위하여 분기포인트를 추적할 필요 없이 포워딩 포인트를 따라 위치 추적을 하

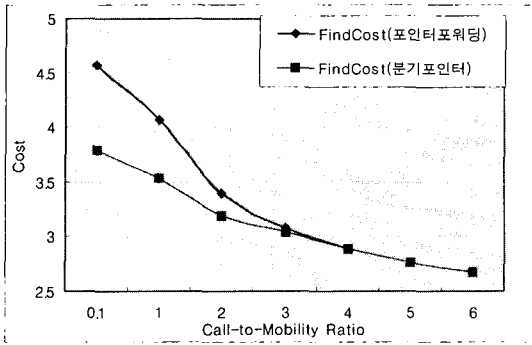
기 때문이다. 반면, 최근에 갱신된 HLR이 아닌 다른 HLR로부터 호가 발생되면, 이동 단말의 위치를 추적하기 위하여 분기포인트가 추적된다. 포인트 포워딩 기법에서처럼, $n+2$ 번의 착신호 발생 후 동일한 HLR에서 호가 발생하였다고 가정해 보자. 이때, 분기포인트의 개수는 착신호 발생 횟수보다 2 적은 값 n 을 갖는다. 즉, 분기 포인트 기법에서 이동 단말의 위치를 추적하기 위하여 n 번의 분기포인트를 추적하고, 그런 후 포워딩 포인트를 추적하여야만 비로소 이동 단말의 위치를 파악할 수 있다. 그러므로 포인트 포워딩 기법의 식(5)로 부터 분기포인트 기법에 대한 "find" 비용을 추정할 수 있다. 즉, 최근에 갱신된 HLR이 아닌 다른 HLR로 부터 호가 발생하는 경우의 "find" 비용은 아래의 식 (8)과 같이 유도할 수 있다.

$$FindCost_{\text{분기포인트}} = \begin{cases} 2 + n\delta + E[k | \frac{\theta}{N}] \delta, & E[k | \frac{\theta}{N}] \geq 1 \quad n > 0 \\ 2 + E[k | \frac{\theta}{N}] \delta + E[k | \frac{\theta}{N}] \delta, & E[k | \frac{\theta}{N}] < 1 \end{cases} \quad (8)$$

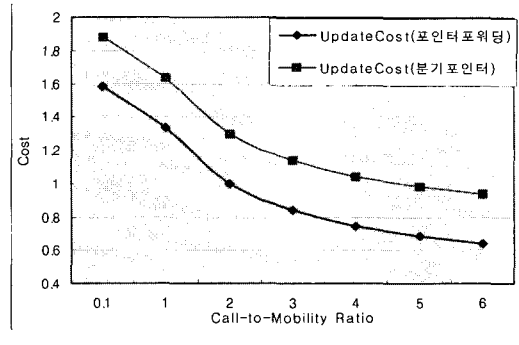
두 연속된 착신호 간 포워딩 포인트의 개수에 대한 기대치가 1보다 적어지면, 분기포인트 기법은 포인트 포워딩 기법을 따르게 된다. 두 연속된 착신호 간 포워딩 포인트의 개수에 대한 기대치가 1보다 적다면, 이동 단말이 등록 영역을 이동하는 횟수가 매우 적으므로 분기포인트의 설정이 불필요하기 때문이다.

4.2 Update 비용

포인트 포워딩 기법과 분기포인트 기법 간 갱신 비용은 매우 유사하나, 분기포인트 기법의 경우 분기포인트를 관리하는 비용이 추가로 부담된다. 포인트 포워딩 기법과 분기포인트 기법을 비교하면 이동 단말이 등록 영역을 가로지를 경우는 두 기법이 동일하게 작동한다. 그러나 일단 호 접속 요구가 발생되어 착신 이동 단말의 위치를 발견하여 호 접속이 완료되면, HLR 및 VLR 데이터베이스를 갱신하게 되는데 이때 두 기법간에 차이가 발생한다. 포인트 포워딩 기법에서는 호 접속 요구가 발생한 HLR이 이동 단말이 현재 머무르고 있는 등록 영역의 VLR을 가리키도록 갱신하는 반면, 분기포인트 기법에서는 추가적인 갱신 작업을 더 수행한다. 그것은 호 접속 요구가 발생하기 전의 HLR이 가리키고 있는 VLR이 현재 이동 단말이 머무르고 있는 VLR을 가리키도록 분기포인트를 설정하도록 하는 작업이다. 그러므로 분기포인트 기법의 "update" 비용에 대한 수학적 모델을 설정하기 위하여 포인트 포워딩 기법과는 달리 분기포인트를 생성하는 비용이 추가적으로 고려되어야 한다. 다음의 식(7),(8)은 포인트 포워딩 기법과 분기



(a) "find" 비용



(b) "update" 비용

그림 5 포인터 포워딩 기법과 분기포인터 기법간 비용 분석

포인터 기법에 대한 등록 작업 시 고려되어야하는 "update"비용에 대한 수학적 모델을 설정한 것이다. f, j, H 는 각각 포워딩 포인터 생성비용, 분기포인터 생성비용, HLR 갱신 비용을 나타낸다.

$$UpdateCost_{\text{포인터포워딩}} = E[k | \frac{\theta}{N}]f + H \quad (7)$$

$$UpdateCost_{\text{분기포인터}} = E[k | \frac{\theta}{N}]f + H + j \quad (8)$$

5. 성능 분석

제 4장에서 설정된 수학적 모델을 근거로 포인터 포워딩 기법 및 분기포인터 기법에 대한 성능을 분석하였다. 성능 분석을 위하여 사용된 파라미터들은 다음과 같다. 기존의 포인터 포워딩 기법과의 비교를 위하여 각 파라미터의 값들은 [6]에서 제시한 값들을 채택하여 성능 평가를 하였다.

- θ : call-to-mobility ratio (CMR) $0.1 \leq \theta \leq 6$
- N : 원격 PSTN 영역의 개수 ($N=4$)
- S : 등록 영역의 크기 ($S=16$)
- V : 사용자가 등록영역에 머무르는 시간 분포에 대한 분산 ($V=1$)
- n : 분기포인터의 개수 ($n=1$)

그림 5는 δ 및 δ' 이 0.5의 값을 가질 때 두 모델의 "find"비용, "update"비용을 나타내고 있다. 이때 "find"비용 관점에서 보면, CMR이 0.1일 때 분기포인터 기법은 포인터 포워딩 기법 대비 17%의 성능 향상을 보

며, CMR이 2일 때는 13%의 성능 향상을 보인다. 그러나 착신호의 발생이 점차 증가함에 따라 포인터 포워딩 기법과 유사한 성능을 보인다. 이는 착신호의 발생 빈도와 등록 영역의 이동 횟수를 비교 시 상대적으로 착신호의 발생 빈도수가 많아지면 HLR의 데이터베이스가 자주 갱신되므로 최신의 등록 정보를 유지할 수 있기 때문이다.

반면, "Update" 비용 관점에서 보면 분기포인터 기법의 경우, 두 VLR 간 분기포인터를 설정하고 기존의 VLR이 가리키고 있는 포워딩 포인터를 삭제하기 위한 추가적인 비용을 고려해야 한다. 그래서 분기 포인터를 관리하는 추가적인 비용을 포인터 포워딩 기법과 비교하여 보면 포인터 포워딩 기법의 "update"비용 대비 약 18% 정도의 추가 비용이 부담된다.

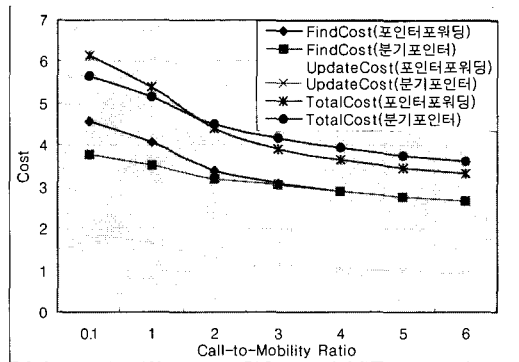


그림 6 비용분석 ($\delta, \delta' = 0.5$)

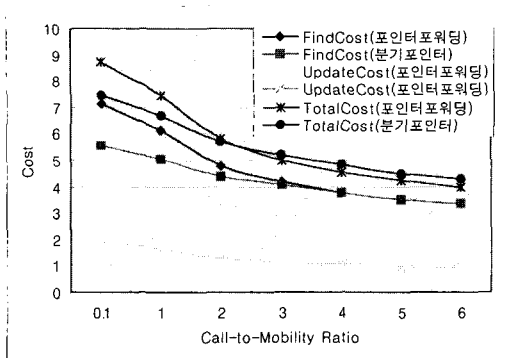


그림 7 비용분석 ($\delta, \delta' = 1$)

그림 6과 7은 두 기법에 대한 "find" 비용, "update" 비용 및 총 비용을 나타내는 그림으로 각각 δ 및 δ' 이 0.5와 1일 때의 값들을 나타낸다. "find" 비용과 "update" 비용을 모두 고려할 경우, CMR이 적을수록 분기포인트 기법에서 적은 비용을 부담하게 된다. 반면, CMR이 커지면 기존 포인터 포워딩 기법에서의 비용 부담이 적어진다.

6. 결론

본 논문에서는 분산 HLR 환경에서 이동 단말의 위치 추적을 위한 비용을 줄이기 위한 분기포인트 기법 (Jumping Pointer Scheme)을 제시하였다. 이 기법은 분산 HLR 환경에서 사용되는 포인터 포워딩 기법 (Pointer Forwarding Scheme)의 변형으로, 최근에 갱신된 HLR이 아닌 다른 HLR로 부터 착신호가 유발되어 질 경우 포인터 포워딩 기법에서는 착신 이동 단말을 찾기 위해서 경유해야 하는 VLR의 개수가 많아질 수 있는데 이러한 현상을 막고자 최근에 갱신된 VLR과 현재 이동 단말이 위치하고 있는 등록 영역의 VLR간 분기포인트(Jumping Pointer)로 연결하여 보다 빠르고 신속하게 위치 추적을 할 수 있도록 한다. 분기포인트를 사용할 경우 위치 추적을 위한 "find" 비용과 HLR 및 VLR 데이터베이스의 자료 갱신을 위한 "update" 비용에 대한 수학적 모델을 설정하고 이에 대한 성능을 평가한 결과, "find" 비용에서는 상당한 성능 향상을 보인 반면, "update" 비용에서는 분기포인트 관리를 위한 추가 비용을 부담해야 한다. 결국 총 비용 관점에서 보면 상대적으로 착신호의 발생은 적고 등록 영역간 이동이 적은 경우, 포워딩 포인터 기법과 비교해 상대적으로 향상된 성능을 보인다.

일반적으로 이동성 관리를 위한 기법을 비교할 때 "find" 비용과 "update" 비용 관점에서 비교된다. 그러나 "find" 비용과 "update" 비용간에는 트레이드오프 (trade-off)가 존재한다. 즉, "find" 비용이 줄어들어 성능이 향상되면 "update" 비용에 대한 부담이 많아진다. 예를 들어, 호 전달 시 착신 이동 단말의 위치를 보다 빠르게 찾기 위하여 HLR 및 VLR 데이터베이스는 보다 정확하고 최신의 정보로 유지되어야 할 필요가 있다. 최신의 정보를 유지하기 위하여 "update"를 위한 정보를 자주 갱신하여야하며, 즉 그에 대한 비용 부담이 많아 지게된다.

그러므로 이동성 관리를 위한 기법을 채택할 경우 "find" 비용 또는 "update"비용 어느 관점에 적합한 관리 기법을 선택할 것이냐를 판단해야한다. 본 논문에서 제시한 분기포인트 기법은 "update" 비용은 조금 더 들더라도 이동 단말의 위치를 보다 빠르게 찾기 위한 관리 기법을 원할 경우 이용이 가능하다.

또한 포인터 포워딩 기법 및 분기포인트 기법 두가지를 동시에 수용하는 방안이 고려되어 질 수 있다. 즉, 착신호 발생 빈도수 대비 등록 영역 간 이동의 비율을 매개 변수로 하여 적응적(Adaptive)으로 포인터 포워딩 기법 또는 분기포인트 기법을 선택하는 방안에 대한 연구가 향후 과제로 남아 있다.

참고 문헌

- [1] R.Jain, Y.-B. Lin, C.N.Lo, and S.Mohan, "A caching strategy to reduce network impacts of PCS," *IEEE J. Selected Areas Commun.*, vol. 12, no. 8, pp.1434-1445, 1994.
- [2] Y.-B. Lin, "Reducing location update cost in a PCS network," *IEEE/ACM Trans. Network.*, vol. 5, no. 1, pp.25-33, 1997.
- [3] R.Jain, Yi-Bing Lin, "Performance modeling of an auxiliary user location strategy in a PCS network," *ACM-Baltzer wireless networks*, Vol. 1, No. 2, pp. 197-210, 1995.
- [4] G.P.Pollini and S.Tabbane, "The intelligent network signaling and switching costs of an alternate location strategy using memory," *IEEEVTC*, 1993.
- [5] H.Xie, S.Tabbane and J.D.Goodman, "Dynamic location area management and performance analysis," *IEEEVTC*, 1993.
- [6] Yi-Bing Lin, Wen-Nung Tsai, "Location Tracking with Distributed HLRs and Pointer Forwarding," *IEEE Trans. on Vehicular Technology*, Vol.47, No. 1, pp.58-64, February 1998



이 보 경

1987년 고려대 수학과(학사). 1987년 ~ 1998년 (주)데이콤 근무. 1995년 Univ. of Birmingham Dept. of Computer Science(석사). 1996년 ~ 현재 고려대 컴퓨터학과 박사과정. 관심분야는 무선데이터통신, 분산처리.



황 중 선

1978년 Univ. of Georgia, Statistics and Computer Science 박사. 1978년 South Carolina Lcander 주립대학 조교수. 1981년 한국표준연구소 전자계산실 실장. 1995년 한국정보과학회 회장. 1982년 ~ 현재 고려대학교 컴퓨터학과 교수. 관심분야는 알고리즘, 데이터베이스, 분산시스템.