

대형 이동 무선 네트워크에 적합한 3-계층 동적 분산 위치 관리 방법

이주영^o 배인한

대구효성가톨릭대학교 컴퓨터·정보통신공학부
{s6047061, ihbae}@cuth.cataegu.ac.kr

A Suitable 3-Tier Dynamic Distributed Location Management Scheme to Large Mobile Wireless Networks

Joo-Young Lee^o Ihn-Han Bac

School of Computer and Information Communication Eng., Catholic University of Taegu-Hyosung

요약

사용자들이 네트워크에 의해 커버되는 영역내의 어느 곳이던지 이동할 수 있는 이동 무선 네트워크에서 어떤 이동 단말기가 다른 이동 단말기와 통신하고자 하는 경우, 먼저 목적지 이동 단말기의 위치를 찾아야 한다. 따라서 이동 단말기의 위치 정보를 관리하는 것은 이동 무선 네트워크에서 아주 중요한 문제이다. 본 논문에서는 대형 이동 무선 네트워크에 적합한 3-계층 동적 분산 위치 관리 방법을 제안한다. 여기서는 이동 단말기가 다른 셀의 이동 단말기로부터의 위치 탐색 요청 회수를 고려하여 계층별로 협동하는 다른 위치 관리 방법을 사용하였으며, 분석적 평가를 통하여 제안하는 방법의 성능을 평가하였다.

1. 서론

기존의 무선 셀룰러 네트워크는 개인 통신 서비스(PCS: Personal Communication Services)로 발전될 것이다. PCS 환경에서 도전적인 문제 중의 하나는 이동 단말기를 갖고 자유롭게 이동하는 PCS 가입자의 위치를 효율적으로 관리하는 것이다. 이동 단말기가 셀을 횡단할 때마다 위치를 수정한다면, 신호처리(signaling) 부하는 네트워크 부하의 주된 요인이 될 것이다. 그러한 신호처리 부하는 SS7과 같은 교환기에서 주된 병목현상을 일으킬 것이다. 따라서 새로운 효율적인 위치 관리 정책들이 PCS를 위해 연구되어야 한다[1].

위치 관리는 위치 탐색과 위치 수정으로 구성된다. 위치 탐색은 어떤 이동 단말기가 위치가 알려지지 않은 다른 이동 단말기와 통신하고자 할 때 발생하고, 위치 갱신은 이동 단말기의 위치가 변경될 때 발생한다. 위치 관리에서 동적으로 변하는 내용을 갖는 위치 레지스터의 설계는 중요한 문제이다. 위치 레지스터의 설계에서 고려되어야 할 사항은 다음과 같다. (i) 위치 레지스터가 언제 수정될 것인가? 만일 이동 단말기의 위치 변경이 일어날 때마다 위치 수정을 한다면, 위치 레지스터는 최근 위치 정보를 가질 것이고, 어떤 이동 단말기를 찾는 데 시간과 노력이 줄어 들 것이다. 그러나, 그러한 정책들은 통신 네트워크와 위치 레지스터에 많은 부하를 줄 것이다. (ii) 위치 레지스터를 집중방식 또는 분산방식으로 관리할 것인가? 집중형 위치 레지스터는 견고성과 확장성 문제가 있다. 따라서 분산형 위치 레지스터가 바람직하다. (iii) 위치 정보는 위치 레지스터들 간에 어떻게 분산될 것인가? 그리고 (iv) 이동 단말기에 대한 위치 정보는 다수의 위치 레지스터들에 어떻게 복사될 것인가? 무선 이동 네트워크에서 이동 단말기의 공간 분포에서 변화와 이동 단말기 위치가 수정되어질 빈도를 사건에 결정하는 것은 거의 불가능하다. 그러므로 위치 관리 정책은 모든 위치 레지스터들간의 응답성의 공평한 분산과 확장성 보장하기 위하여 (iii)과 (iv)를 제시해야 한다.

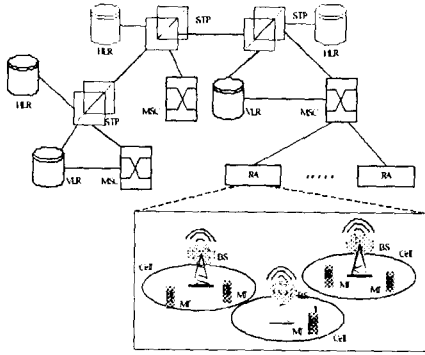
본 논문에서는 대형 이동 무선 네트워크에 적합한 동적 분산 위

치 관리 방법을 제안하고, 그것의 성능은 분석적 모델로 평가한다. 제안하는 방법에서는 위치 레지스터들을 클러스터들의 3-계층 구조로 구성하고, 이동 단말기가 다른 셀의 이동 단말기들로부터의 위치 탐색 요청 회수를 고려하여 각 계층별 다른 위치 수정 정책을 사용한다. 따라서 인기 있는 이동 단말기의 위치 정보는 다수의 위치 레지스터들에서 관리되는데 반해, 인기 없는 이동 단말기의 위치 정보는 상대적으로 적은 위치 레지스터들에서 관리된다. 따라서 인기 있는 이동 단말기의 위치 정보는 작은 비용으로 빠르게 액세스될 수 있다. 본 논문의 구성은 다음과 같다. 2장에서는 PCS의 구조에 대해 설명하고, 3장에서는 관련연구를 살펴본다. 4장에서는 본 논문에서 제안하는 대형 이동 무선 네트워크에 적합한 3-계층 동적 분산 위치 관리 방법에 대해 설명하고, 5장에서는 분석적 모델로 제안한 방법의 성능을 평가하고, 그리고 마지막으로 6장에서 결론을 맺는다.

2. PCS 구조

PCS는 공중 교환 전화망(PSTN: Public Switched Telephone Network)과 신호처리 네트워크로 구성된다. PSTN은 호에 대한 정보 내용을 실질적으로 전송하는데 사용되는 네트워크인데 반해, 신호처리 네트워크는 이동 단말기(MT: Mobile Terminal)의 위치를 추적하고 호를 설정하기 위해 사용되는 네트워크로 SS7(Signaling System No. 7) 프로토콜을 사용한다[3].

그림 1은 본 논문에서 기초한 PCS 구조를 나타낸다. 시스템의 전체 영역은 다수의 셀들로 분할되고, 몇 개의 셀들이 모여 하나의 등록 지역(RA: Registration Area)을 구성한다. 셀내의 모든 MT는 무선 링크를 통해 기지국(BS: Base Station)과 통신하고, 다시 BS는 이동 교환국(MSC: Mobile Switching Center)을 통해 고정 네트워크에 연결된다. 고정 네트워크는 MSC와 위치 데이터베이스 간에 사용자 정보와 시그널링 메시지를 전송한다. 현재 PCS 시스템에서는 PSTN이 백본 고정 네트워크로 사용된다.



(그림 1) 단순화된 PCS 구조

3. 관련연구

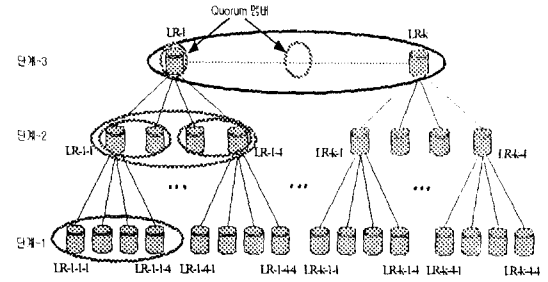
최근에 이동 무선 네트워크를 위한 다수의 위치 관리 방법들이 제안되었다. IS-41 방법에서, 위치 갱신은 MT가 다른 VLR(Visitor Location Register)에 의해 서비스되는 등록 지역으로 들어갈 때마다 HLR(Home Location Register)은 그 MT에 대한 위치 정보를 새로운 VLR의 주소로 갱신함으로써 이루어진다. 따라서, 많은 네트워크 부하와 데이터베이스 질의가 HLR에 집중된다[1]. PFDHLR(pointer forwarding with distributed HLR)에서, 위치 갱신은 HLR을 갱신하는 대신 RA를 변경한 MT의 이전 VLR에서 새로운 VLR로 포인터를 설정함으로써 이루어지고, RA는 어떤 MT가 개별적인 HLR에 의해 가장 마지막으로 탐색된 위치이므로, 분산된 HLR은 어떤 하나의 MT에 대해서도 다른 RA를 가리킬 수 있다[4]. [5]에서는 대형 PCS에 적합하도록 LR들을 클러스터들의 계층구조로 나누고, 위치 정보의 유포를 가능한 한 클러스터 내부로 제한하는 방법을 제안하였다.

4. 3-계층 동적 분산 위치 관리 방법

완전 분산 위치 관리 방법은 모든 MT들에 대한 새로운 위치 정보는 이동 무선 네트워크내의 모든 LR들에 유포되어진다. 네트워크의 크기가 커질수록 위치 정보 유포는 네트워크 대역폭의 중요한 부분을 낭비할 뿐 아니라 많은 수정 메시지를 처리하기 위하여 LR 자원들의 중요한 부분을 소비한다. 이러한 문제점을 피하기 위하여, 본 논문에서는 MT의 위치 정보 수정이 그 MT를 현재 서비스하고 있는 LR이 속한 클러스터내의 LR들로 제한되는 동적 분산 위치 관리 방법을 제안한다. 여기서는 MT와 관련된 HLR은 없고, 각 MT는 [L-T, t] 동안에 평균적으로 다른 LR들로부터 수신된 단위 시간 당 질의의 개수가 미리 정해진 임계치를 초과하면 hot MT이고, 아니면 cold MT이다. 제안하는 동적분산 위치 관리 방법에서는 그림 2와 같이 LR들을 클러스터로 구성한다 다음, 다시 3-계층구조를 형성한다.

단계-1의 각 LR은 부모 LR의 모든 자식 LR들중, 그 LR이 속한 같은 클러스터내의 모든 LR에서 현재 서비스되고 있는 MT들의 위치 정보를 관리하고, 단계-2에서는 단일 위치 변경을 일으킨 MT가 hot이면, 그 MT를 서비스하고 있는 LR을 자식으로 갖는 부모 LR의 모든 형제 LR들에서 그 MT의 위치 정보가 관리된다. 아니면, 그 MT를 서비스하고 있는 LR을 자식으로 갖는 부모 LR이 조상 LR의 left children에 속하면 조상 LR의 left subtree에 속하는 형제 LR들에서만 또는 right children에 속하면 조상의 right subtree

에 속하는 형제 LR들에서만 그 MT의 위치 정보가 관리된다. 그리고 단계-3에서는 단일 위치 변경을 일으킨 MT가 hot이면, 모든 조상 LR들에서 그 MT의 위치 정보가 관리된다. 아니면, 단계-3의 모든 LR들로 격자기반 방법으로 기초하여 quorum을 구성한 다음, 그 MT의 조상 LR의 quorum에 속하는 조상 LR들에서만 그 MT의 위치 정보를 관리한다. 단계-2와 단계-3의 LR들에서 관리되는 MT에 대한 위치 정보는 그 MT가 현재 서비스되는 형제 LR에 대한 포인터 정보이다.



(그림 2) 3-단계 계층적 배열의 개념적 모델

그림 2에서, 단일 어떤 cold MT가 LR-1-1-1의 서비스 영역 안에 있다면, 그 MT의 위치 정보는 단계-1에서는 LR-1-1이 속한 클러스터내의 LR들 즉, LR-1-1-1, ..., LR-1-1-4에서 관리되고, 단계-2의 LR-1-1에서는 그 MT에 대한 위치 정보로 LR-1-1-1을 가리키는 포인터를 관리하고, 아울러 LR-1-2에서는 그 MT에 대한 위치 정보로 LR-1-1을 가리키는 포인터를 관리한다. 그리고 단계-3의 LR-1에서는 그 MT에 대한 위치 정보로 LR-1-1을 가리키는 포인터를 관리하고, LR-1의 quorum에 속하는 다른 LR들은 그 MT의 위치 정보로 LR-1을 가리키는 포인터를 관리한다.

어떤 MT를 서비스하는 LR을 추적하는 것은 그 MT를 서비스하고 있는 LR에 도달할 때까지 홑 단위의 LR트리 순회를 포함한다. 단일 어떤 MT에 대한 위치 엔트리가 어떤 LR에 존재하지 않으면, 추적 요청이 그 LR의 부모 LR로 상향 순회된다. 목적지 MT가 cold인 경우, 그 LR의 조상 LR에서도 그 MT에 대한 위치 엔트리가 없으면, 그 조상 LR의 quorum 멤버들에게 추적 요청 메시지를 멀티캐스트 한다. 예를 들어 그림 2에서 LR-1-4-4에 있는 근원지 MT가 LR-1-1-1에 의해 서비스되고 있는 목적지 hot MT의 위치를 추적한다면, LR-1-4-4는 LR-1-4에게 추적 요청 메시지를 전송하고, LR-1-4는 LR-1-1에게 추적 요청 메시지를 전송하고, 그리고 그것을 LR-1-1에게로 하향 순회한다. 찾은 위치 정보는 LR-1-4-4에게로 반환된다.

5. 성능 평가

5장에서는 분석적 모델을 사용하여 제안하는 동적 분산 위치 관리 방법의 성능을 평가한다. 여기서 평가된 항목은 위치 탐색 비용과 위치 수정 비용을 합한 전체 비용이다.

위치 탐색 비용은 어떤 MT로부터의 호가 국부 클러스터에 있는지 없는지, hot인지 아닌지에 의존한다. 국부 클러스터로부터 호를 찾는 비용은 다음과 같다.

$$Find_{local-call-level1} = Cost(LR_{colloc} \leftrightarrow LR_{colloc}) = Cost(LR_{local} \leftrightarrow LR_{local}) \quad (1)$$

단계-2의 형제 LR로부터 호를 찾는 비용은 다음과 같다.

$$\begin{aligned}
 Find_{local-call-level2} = & Cost(LR_{caller} \leftrightarrow LR_{caller-level2}) \\
 & + Cost(LR_{caller-level2} \leftrightarrow LR_{caller-level2}) \\
 & + Cost(LR_{caller-level2} \leftrightarrow LR_{caller}) \quad (2)
 \end{aligned}$$

그리고 단계-3의 형제 LR로부터 호를 찾는 비용을 다음과 같다.

$$\begin{aligned}
 Find_{sibling-call} = & Cost(LR_{caller} \leftrightarrow LR_{caller-level2}) \\
 & + Cost(LR_{caller-level2} \leftrightarrow LR_{caller-level3}) \\
 & + Cost(LR_{caller-level3} \leftrightarrow LR_{caller-level3}) \\
 & + Cost(LR_{caller-level3} \leftrightarrow LR_{caller-level2}) \\
 & + Cost(LR_{caller-level2} \leftrightarrow LR_{caller}) \quad (3)
 \end{aligned}$$

따라서 위치 탐색 비용은 다음과 같다.

$$\begin{aligned}
 Find = & P_{local-call-level1} \times Find_{local-call-level1} \\
 & + P_{local-call-level2} \times Find_{local-call-level2} \\
 & + (1 - (P_{local-call-level1} + P_{local-call-level2})) \times Find_{sibling-call} \quad (4)
 \end{aligned}$$

여기서 $P_{local-call-level1}$ 은 도착된 호가 국부 클러스터내의 어떤 MT로부터의 호일 확률, $P_{local-call-level2}$ 는 도착된 호가 같은 조상 LR의 단계-2 형제 LR내의 어떤 MT로부터의 호일 확률을 나타낸다.

위치 수정 비용은 어떤 MT가 같은 단계-2 LR 또는 단계-3 LR에 속하는 단계-1 LR로 이동하였는지 아닌지, 그리고 그것이 hot인지 아닌지에 의존한다. MT가 같은 단계-2 RT에 속하는 RT로 이동한 경우, 그 위치 정보는 그 클러스터내의 모든 LR들에 위치 정보 유포한다. 따라서 위치 수정 비용은 다음과 같다.

$$Update_{level1} = C_{level1} \times (M_{level1} - 1) \quad (5)$$

여기서 C_{level1} 은 두 개의 인접한 단계-1 LR들을 연결하는 링크의 평균 비용이고, M_{level1} 은 단계-1 클러스터내의 LR들의 개수를 나타낸다.

만일 어떤 MT가 같은 단계-3 LR에 속하는 다른 단계-2 LR로 이동한 경우, 그 MT가 hot인지 아닌지에 따라 위치 수정 비용이 달라진다.

$$\begin{aligned}
 Update_{level2} = & 2 \times C_{level1} \times (M_{level1} - 1) + C_{level2} \\
 & \times \{P_{update-hot} \times (M_{level2} - 1) + (1 - P_{update-hot}) \times (\frac{M_{level2}}{2} - 1)\} \quad (6)
 \end{aligned}$$

여기서 C_{level2} 는 두 개의 인접한 단계-2 LR들을 연결하는 링크의 평균 비용이고, $P_{update-hot}$ 은 이동하는 MT가 hot일 확률이고, 그리고 M_{level2} 는 단계-2의 형제 LR들의 개수를 나타낸다.

어떤 MT가 다른 단계-3 LR에 속하는 LR로 이동한 경우에도 그 MT가 hot인지 아닌지에 따라 위치 수정 비용이 달라진다.

$$\begin{aligned}
 Update_{level3} = & 2 \times C_{level1} \times (M_{level1} - 1) \\
 & + 2 \times C_{level2} \times \{P_{update-hot} \times (M_{level2} - 1) + (1 - P_{update-hot}) \times (\frac{M_{level2}}{2} - 1)\} \\
 & + C_{level3} \times \{P_{update-hot} \times (M_{level3} - 1) + (1 - P_{update-hot}) \times 2(2\sqrt{M_{level3}} - 1)\} \quad (7)
 \end{aligned}$$

여기서 C_{level3} 는 두 개의 인접한 단계-3 LR들을 연결하는 링크의 평균 비용이고, M_{level3} 는 단계-3의 모든 LR의 개수를 나타낸다. 그러므로 위치 수정 비용은 다음과 같다.

$$\begin{aligned}
 Update = & P_{local-move} \times Update_{level1} \\
 & + P_{level2-sibling-move} \times Update_{level2} \\
 & + (1 - (P_{local-move} + P_{level2-sibling-move})) \times Update_{level3} \quad (8)
 \end{aligned}$$

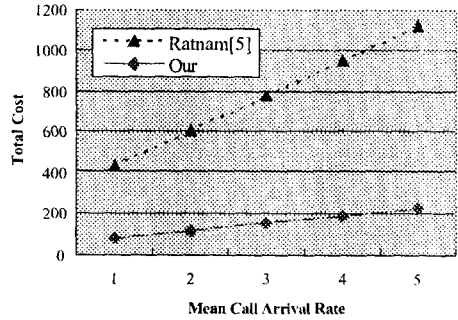
여기서 $P_{local-move}$ 는 MT가 같은 클러스터내의 다른 LR로 이동할 확률, $P_{level2-sibling-move}$ 는 MT가 같은 조상 LR의 단계-2 형제 LR로 이동할 확률을 나타낸다.

이동 당 호 도착율을 λ , 등록 영역에 MT의 평균 거주 시간을 $1/\mu$ 라 두면, 제안하는 방법의 위치 관리를 위한 전체 비용은 다음과 같다.

$$Total = Update + \frac{\lambda}{\mu} \times Find \quad (9)$$

제안하는 방법의 성능을 수치 결과로 분석하기 위하여, 우리는 논리적으로 32x32 메쉬 위상으로 구성된 1,024개의 LR들을 갖는 이동 무선 네트워크라 가정한다. 그리고 4개의 LR들을 하나의 클러스터를 구성하고, 인접한 단계-1 LR들을 연결하는 링크 비용을 1로 정규화하고, 찾는 호가 hot일 확률을 0.7, $P_{local-to-level1} = 0.5$, $P_{local-to-level2} = 0.3$, $P_{update-hot} = 0.3$, $P_{local-move} = 0.5$, 그리고 $P_{level2-sibling-move} = 0.3$ 으로 가정하였다.

성능 평가 결과, 본 논문에서 제안하는 방법이 Ratnam[5]의 방법에 비해 호 도착율에 관계없이 항상 성능이 우수함을 알 수 있다 (그림 3 참조).



(그림 3) 평균 호 도착율에 따른 전체 비용

6. 결론

본 논문에서는 대형 이동 무선 네트워크에 적합한 동적 분산 위치 관리 방법을 제안하고, 분석적 모델을 통하여 그것의 성능을 평가하였다. 그 결과, 제안하는 방법이 Ratnam의 방법보다 성능을 우수함을 알 수 있었다. 제안하는 방법에서는 이웃한 LR들로 클러스터를 구성하여 이동 국부성을 얻을 수 있었고, hot MT의 위치 정보는 단계-2와 단계-3의 다수의 LR들에 저장되어 빠른 액세스가 가능하도록 하였고, 그리고 cold MT의 위치 정보는 단계-2와 단계-3의 부분 LR들에서 관리된다. 따라서 상위 단계(단계-2와 단계-3)의 LR들의 방대한 질의 부하와 용량 부하를 크게 감소시킬 수 있었다.

참고문헌

- [1] P. Krishna, N. H. Vaidya, D. K. Pradhan, "Efficient Location Management in Mobile Wireless Networks", Technical Report #96-030, Dept. of Computer Science, Texas A&M Univ., July 1996.
- [2] R. Prakash, M. Singhal, "A Dynamic Approach to Location Management in Mobile Computing Systems," Dept. of Computer and Information Science, The Ohio State Univ., Technical Report, OSU-CISRC-4/96-TR22, p. 15, 1996.
- [3] S. Biaz, N. H. Vaidya, "Tolerating Visitor Location Register Failures in Mobile Environments", *Proceedings of the 17th IEEE SRDS'98*, Oct. 1998.
- [4] Y. B. Lin, W-N Tasi, "Location Tracking with Distributed HLRs and Pointer Forwarding", *IEEE Trans. on Veh. Technol.*, 47(1), pp.58-64, 1998.
- [5] K. Ratnam, I. Matta, S. Rangarajan, "A Fully Distributed Location Management Scheme for Large PCS," Technical Report 1999-010, Dept. of Computer Science, Boston University, p.11, 1999.