

이동 호스트의 병렬적 위치 정보 탐색을 위한 서버의 계층 구조

(A Hierarchical Server Structure for Parallel Location Information Search of Mobile Hosts)

정 광 식 [†] 유 헌 창 ^{**} 황 중 선 ^{***}

(Kwang-Sik Chung) (Heon-Chang Yu) (Jong-Sun Hwang)

요 약 이동 컴퓨팅 환경의 발전은 기존의 분산 시스템에서는 볼 수 없었던 이동 호스트의 위치 정보 관리 문제, 이동 호스트의 접속 단절 문제, 낮은 무선 통신 대역폭의 문제 등 과 같은 새로운 문제점을 발생시켰다. 특히 이동 호스트의 위치 정보 관리 기법은 이동 컴퓨팅 환경의 성능을 저하시킨다. 이동 호스트의 이동성으로 인해 이동 호스트의 주소가 이동 호스트의 위치에 종속적으로 변화하며, 이동 호스트의 주소는 고유 주소와 현재 주소의 사상을 통해 관리된다. 이동 호스트의 고유 주소와 현재 주소를 사상시켜 주기 위한 이전의 기법들은 이동 호스트의 증가 및 위치 정보의 분산관리를 고려하지 않았기 때문에 시스템의 확장성에 큰 문제를 가지고 있다.

이 논문에서는 이동 호스트의 주소를 분산 관리하기 위해 개선된 n -깊이 위치 정보 탐색 트리와 개선된 n -깊이 위치 정보 탐색 트리를 기반으로 한 병렬적 위치 정보 탐색과 갱신 기법을 제안한다. 개선된 n -깊이 위치 정보 탐색 트리는 링 구조를 이용하여 위치 정보 서버를 지역적으로 분할한 위치 정보 서버의 논리적 계층 구조이며, 병렬적 탐색 기법을 기반으로 이동 호스트의 위치 탐색 비용과 갱신 비용을 줄였다. 개선된 n -깊이 위치 정보 탐색 트리를 기반으로 한 위치 정보 탐색/갱신 기법은 위치 정보 서버의 논리적 위치가 멀리 떨어져 있는 경우에도 병렬적 탐색 기법을 통해 기존의 방법보다 좋은 결과를 보인다. 또한, 탐색 비용의 감소에도 불구하고 갱신 비용의 증가를 가져오지 않는다.

Abstract The development in the mobile computing systems have arisen new and previously unforeseen problems, such as problems in information management of mobile host, disconnection of mobile host and low bandwidths of wireless communications. Especially, location information management strategy of mobile host results in an increased overhead in mobile computing systems. Due to the mobility of the mobiles host, the changes in the mobile host's address depends on the mobile host's location, and is maintained by mapping physical address on virtual address. Since previously suggested several strategies for mapping method between physical address and virtual address did not tackle the increase of mobile host and distribution of location information, it was not able to support the scalability in mobile computing systems.

Thus, to distribute the location information, we propose an advanced n -depth LiST (Location information Search Tree) and the parallel location search and update strategy based on the advanced n -depth LiST. The advanced n -depth LiST is logically a hierarchical structure that clusters the location information server by ring structure and reduces the location information search and update cost by parallel search and updated method. The experiment shows that even though the distance of two MHs that communicate with each other is large, due to the structural distribution of location information, advanced n -depth LiST results in good performance. Moreover, despite the reduction in the location information search cost, there was no increase in the location information update cost.

[†] 학생회원 : 고려대학교 컴퓨터학과

cks@disys.korea.ac.kr

^{**} 정 회 원 : 고려대학교 컴퓨터교육과 교수

yuhc@comedu.korea.ac.kr

^{***} 중신회원 : 고려대학교 컴퓨터학과 교수

hwang@disys.korea.ac.kr

논문접수 : 2000년 2월 25일

심사완료 : 2000년 12월 6일

1. 서론

이동 컴퓨팅 시스템(mobile computing systems)은 무선 통신 기술의 발전과 수요를 바탕으로 급속히 발전하고 있다. 이동 컴퓨팅 시스템의 특징은 이동 호스트(Mobile Host: MH)의 이동중에도 무선으로 고정망(fixed network)과 연결을 유지하며 지속적인 연산 수행을 가능하게 하고, 사용자에게 투명한(transparent) 분산 시스템 환경을 제공하는 것이다[1, 2, 3, 4].

이동 컴퓨팅 시스템의 모델은 이동 호스트(mobile host)와 고정 호스트(fixed host), 두 종류의 컴퓨터 집합으로 구성된다[3]. 고정 호스트 중 일부는 기지국(base station) 또는 MSS(Mobile Support Station)로 불리는데, 이들은 이동 호스트들과의 무선 통신을 위한 무선 인터페이스 기능을 제공하며, MSS_i^k 로 표시한다. 하나의 기지국은 셀이라 불리는 하나의 통신 지역을 관리한다. 각 이동 호스트들은 하나의 기지국(MSS)과 통신하며, 기지국 MSS_i^k 가 관리하는 셀에 속한 이동 호스트는 MH_{ij}^k 로 표시한다. 전체 시스템에서 i, j, k 는 유일하다. 연산 수행 중인 호스트가 다른 셀로 전환하는 것을 핸드 오프(hand-off)라 하고, 이동중인 이동 호스트는 새로운 셀의 고정 호스트와 공조적으로 수행중인 연산을 계속할 수 있다.

기지국들은 기존의 통신망을 기반으로 다른 기지국과 연결되거나, 위치 정보 서버를 통해 다른 이동 호스트의 위치 정보를 제공받는다. 처음 이동 호스트가 이동 통신망에 등록될 때, 고유 주소(virtual address)를 부여받는다. 고유 주소는 이동 호스트의 위치에 관계없이 일정한 값을 가지며, 물리적으로 고정되어 있다. 이동 호스트가 새로운 셀에 진입하면서, 기지국으로부터 새로운 주소를 부여받는다. 이 주소를 현재 주소(physical address)라 하며, 새로운 셀로 이동할 때마다 기지국으로부터 새로운 주소값을 부여받는다. 위치 정보 서버는 기지국에 대해 이동 호스트의 위치 정보를 제공한다. 위치 정보 서버는 이동 호스트의 현재 주소 추적과 관리를 책임진다. 즉, 이동 호스트가 가지고 있는 고유 주소와 현재 주소 사이의 사상 정보(mapping information)를 관리한다[6, 7, 8].

이동 컴퓨팅 환경에서 이동 호스트의 위치를 찾아내기 위한 비용은 이동 컴퓨팅 성능을 저하시킨다[9]. 또한 이동 호스트의 증가에 대비한 위치 정보 관리의 확장성(scalability) 지원도 이동 컴퓨팅 환경의 성능 향상을 위한 해결되어야 할 문제이다. 기존의 이동 호스트 위치 관리 기법들은 HLS(home location server)와

VLS(visit location server)사이의 위치 정보 관리에 집중되어 있었다. 즉, 중앙 집중 관리를 기반으로 캐싱 기법, 위치 정보의 선점 기법에 대한 연구로 인해 이동 컴퓨팅 시스템의 확장성에 큰 문제를 가진다. 고정된 수와 고정된 위치의 HLS가 이동성을 갖는 다수의 이동 호스트의 위치 정보 관리를 중앙 집중 관리하는 구조이기 때문이다. 따라서 이 논문에서는 이동 컴퓨팅 환경의 위치 정보 관리가 확장성을 갖기 위해, 이동 호스트의 위치 정보 서버를 계층화하고, 위치 정보를 분산 관리하기 위한 n -깊이 위치 정보 탐색 트리를 제안하고, 이동 호스트의 위치를 적은 부담으로 빠르게 찾기 위해 논리적으로 분할된 위치 정보 서버에서의 병렬적 탐색 기법을 제안한다.

이 논문은 2장에서 기존 연구에 대한 고찰과 연구 동기에 대해 논의한다. 3장에서는 위치 정보 관리: 위해 개선된 n -깊이 위치 정보 탐색 트리라는 자료 구조를 제안한다. 4장에서는 제안된 개선된 n -깊이 위치 정보 탐색 트리를 기반으로 한 위치 정보의 탐색, 갱신, 탐색-갱신 기법에 대해 설명한다. 5장에서는 개선된 n -깊이 위치 정보 탐색 트리를 기반으로 위치 정보의 탐색, 갱신, 탐색-갱신 기법에 대한 성능 평가를 하고, 실험 결과를 기존의 방법과 비교한다. 6장에서 결론을 내린다.

2. 관련 연구

급속하게 발전되고 있는 이동 컴퓨팅 환경에서는 기존의 분산 컴퓨팅 환경과 달리 많은 문제점들이 발생되고 있다. 이동 호스트의 결합 포용, 이동 호스트의 위치 정보 관리 등과 같은 문제들은 이동 컴퓨팅 환경에서 이동 호스트의 이동성으로 인해 발생되며, 전체 시스템의 성능에 큰 영향을 미치고 있다. 특히 위치 정보 관리 문제는 이동 호스트의 이동 빈도에 비례하여 전체 시스템에 성능을 저하시킨다. 위치 정보 관리의 위치 정보 갱신(update), 위치 정보 탐색(search), 위치 정보 탐색-갱신(search-update)으로 구성된다. 위치 정보 갱신은 이동 호스트가 위치를 변경하였을 때 발생된다[11]. 위치 정보 탐색은 어떤 이동 호스트가 통신을 요구할 경우, 호출된 이동 호스트의 현재 주소와 고유 주소의 사상을 위해 발생된다. 위치 정보 탐색-갱신은 위치 정보 탐색이 성공하고 난 후에 탐색된 호스트에 대한 위치 정보가 갱신되었을 때 발생된다. 다양한 방법이 위치 정보 탐색, 위치 정보 갱신, 위치 정보 탐색-갱신에 대해 만들어진다.

그림 1의 이동 컴퓨팅 시스템 모델을 바탕으로 이동 호스트의 위치 정보 관리를 위해 다음과 같은 관련 연

구들이 진행되어 왔다.

논문 [5]는 이동 컴퓨팅 환경에서 위치 디렉토리 관리에 대한 결합 포용 기법을 제안하였다. 이 기법은 기지국의 실패와 이동 호스트의 접속 단절을 포용할 수 있다. 논리적 타임스탬프는 이전의 위치 정보와 새로운 위치 정보를 구분해 내는데 사용된다. 이 기법은 논리적 타임스탬프의 결합까지 포용해 준다.

논문 [10]에서는 직관적으로 위치 정보 서버를 분류하여 집단화시켰다. 즉, 이동 호스트가 자주 접속하는 셀은 함께 묶고, 자주 접속하지 않는 셀은 서로 다른 위치 정보 관리자가 관리하는 분리된 분할(partition)로 묶는다. 이러한 정보는 다양한 사용자의 이동 경향에 대한 관찰을 기반으로 하여 얻어지며, 사용자 분할(user partition)의 개념으로 관리된다. 이 기법에서는 이동 호스트의 위치 정보 관리를 위해 사용자 정보(user profile)를 사용하여, 위치 정보 서버의 분할을 동적으로 관리하였다. 하지만 이 논문에서는 이동 호스트가 사용자 정보를 항상 가지고 있어야 하며, 이는 셀 이동시에 부가적으로 이동되어야 하는 정보량을 증가시킨다. 또한 사용자 정보가 변함에 따라 디렉토리 구조를 변경시켜야 하고, 이는 셀 이동 시간을 증가시킨다.

논문 [11]에서는 트리를 이용한 위치 정보 서버의 논리적 계층 구조를 기반으로 한 위치 정보의 효율적인 적용적 관리 기법을 제안하였다. 위치 정보 서버들의 논리적 계층 구조는 위치 정보의 분산 관리와 이동 컴퓨팅 환경의 확장성 문제를 해결하기 위해 제안되었다. 하지만 실제 이동 컴퓨팅 환경에서의 셀 구성을 고려하지 않는 논리적 계층 구조였기 때문에, 인접한 셀과의 통신에서도 위치 정보 서버의 논리적 계층 구조에 따라 불필요한 부가적 탐색 시간이 필요한 경우가 발생하는 단점을 가지고 있다. 이것은 셀의 위치가 인접해 있음에도 불구하고, 위치 정보 서버의 논리적 계층 구조 때문에 위치 정보 탐색이 최상위 위치 정보 서버에서까지 수행되는 경우에 발생된다.

논문 [11]에서 제안된 기법에서 서로 인접한 셀들이 트리 구조를 이용한 논리적 계층 구조로 인해 탐색 비용이나 탐색-갱신 비용이 커지는 경우가 발생된다. 이러한 문제를 해결하기 위해 이 논문에서는 같은 깊이에 위치하는 위치 정보 서버들 사이에 논리적인 링을 가지는 트리 형태의 논리적 계층 구조를 제안한다. 위치 정보 서버에 링 구조를 적용함으로써 지역적 분할과 병렬적 탐색을 가능하게 되었다. 위치 정보 서버의 논리적 계층 구조에 의해 발생하는 불필요한 탐색 비용을 줄이고 이동성을 가지는 이동 호스트의 위치 정보를 위치

정보 서버에 분산관리하기 위한 자료 구조로 개선된 n-깊이 위치 정보 탐색 트리를 제안하고, 개선된 n-깊이 위치 정보 탐색 트리를 기반으로 위치 정보의 탐색, 갱신, 탐색-갱신 기법을 제안하고, 이를 실험한다. 실험은 셀간 호출 거리와 셀간 이동 거리를 기준으로 서로 다른 위치 정보 관리 기법의 성능을 평가하기 위해 수행되었다. 위치 정보 탐색-갱신 기법은 위치 정보 갱신의 비용을 크게 증가시키지 않고 위치 정보 탐색 비용을 감소시켰다.

3. 링 구조를 가지는 위치 정보 서버의 계층 구조

이 장에서는 트리를 기반으로 한 위치 정보 서버의 논리적 계층 구조와 트리에 링 구조를 결합한 위치 정보 서버의 논리적 계층 구조를 기반으로 한 위치 정보 탐색 기법의 성능을 비교한다. 위치 정보 서버의 논리적 계층 구조의 성능 분석을 위해 이 논문은 이동 호스트와 위치 정보 서버에 관련된 다음과 같은 가정을 한다.

[가정 1]

- i) 이동 호스트는 다른 이동 호스트의 위치 정보를 캐쉬하지 않는다.
- ii) 각 위치 정보 서버는 이동 호스트의 위치 정보를 캐쉬하지 않는다.
- iii) 기지국(단말 위치 정보 서버)에는 하나의 이동 호스트가 존재한다. 따라서, 각 셀에는 단 하나의 이동 호스트가 존재한다.
- iv) 최상위 위치 정보 서버의 깊이는 0이다.
- v) 탐색 비용은 위치 정보 서버에 존재하는 위치 정보 데이터베이스에서의 탐색 시간으로, 단위 시간 1로 가정한다.

3.1 이진 트리를 기반으로 한 위치 정보 서버의 계층 구조

이진 트리에 링 구조를 적용한 위치 정보 서버의 논리적 계층 구조를 알아본다. 논문 [11]에서 제안된 트리를 기반으로 한 위치 정보 서버의 계층 구조는 그림 1과 같다. 그림 1에서 위치 정보 서버의 계층구조는 이진 트리를 기반으로 한다. 그림 2는 이진 트리에 깊이 3의 노드들에 대해 링 구조를 적용시킨 것으로 3-깊이 위치 정보 탐색 트리라 하며, n-깊이 위치 정보 탐색 트리의 정의는 다음과 같다.

[정의 1] n-깊이 위치 정보 탐색 트리(n-depth Location information Search Tree: n-depth LiST)

(1) 공집합이거나 하나의 최상위 노드와 왼쪽과 오른쪽에 서로 다른 서브트리를 구성하는 노드들의 유한집

합이다.

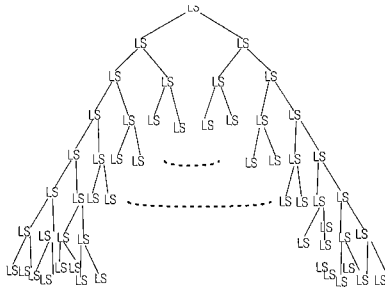


그림 1 트리를 기반으로 한 위치정보서버의 계층구조

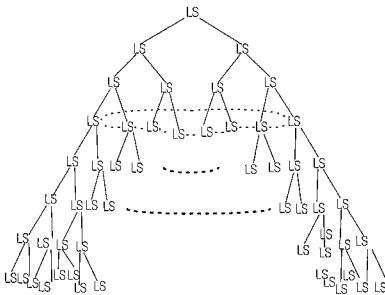


그림 2 링 구조가 결합된 위치정보서버의 계층구조

(2) 트리를 구성하는 깊이 n 인 모든 노드는 4개의 링크를 갖는다. 2개의 링크는 두 개의 자식 노드(child node)에 대한 링크로 실선으로 표시하고, 나머지 2개의 링크는 위치 정보 탐색 링크라 하고 같은 깊이에서 인접한 다른 노드에 대한 링크로 점선으로 표시한다. (최상위 노드의 깊이는 0으로 한다.)

(3) 깊이 n 인 노드들의 위치 정보 탐색 링크에 의한 연결은 링 구조를 형성한다.

1-깊이 위치 정보 탐색 트리의 논리적 계층 구조는 이진 트리의 위치 정보 서버의 계층 구조와 같은 탐색 비용을 보인다. 이것은 깊이 1의 링 구조에서 위치 정보 탐색 연산은 이진 트리 구조에서 최상위 위치 정보 서버에서의 탐색과 같기 때문이다. 2-깊이 위치 정보 탐색 트리는 깊이 2에 있는 위치 정보 서버 4개에 링 구조를 적용한 구조이다. 3-깊이 위치 정보 탐색 트리에서 링 구조에 적용되는 위치 정보 서버는 8개이고, 4-깊이 위치 정보 탐색 트리에서는 깊이가 4인 16개의 위치 정보 서버가 링 구조를 가지게 된다. 1-깊이 위치 정보 탐색 트리에서의 탐색 비용(SCNRSB; Search Cost of No Ring Structure in Binary tree), 2-깊이

위치 정보 탐색 트리에서의 탐색 비용(SCRSB-2; Search Cost of Ring Structure in 2-depth Binary tree), 3-깊이 위치 정보 탐색 비용(SCRSB-3; Search Cost of Ring Structure in 3-depth Binary tree)과 4-깊이 위치 정보 탐색 트리에서 위치 정보 탐색 비용(SCRSB-4; Search Cost of Ring Structure in 4-depth Binary tree)이 그림 3의 그래프에서 비교된다.

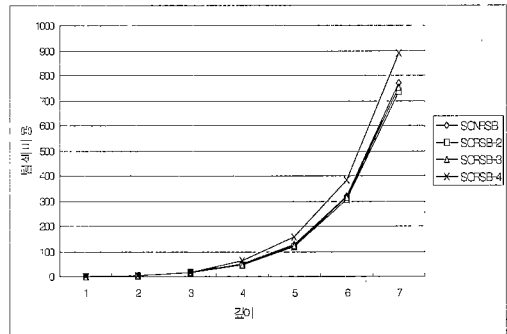


그림 3 탐색 비용의 비교

각 셀에 존재하는 하나의 이동 호스트가 자신이 속한 셀을 제외한 다른 셀에 존재하는 모든 이동 호스트와 통신하기 위한 위치 정보 탐색 비용의 평균값을 구한 결과이다. 즉, 셀 사이의 거리와는 관계없이 각 셀에 존재하는 하나의 이동 호스트가 하나의 셀에서 이동없이 전체 셀에 분포된 모든 이동 호스트를 호출하였을 경우, 발생하는 전체 위치 정보 탐색 비용의 셀 수에 대한 상대적 비용을 다음의 식을 통해 얻었다. 이진 트리를 기반으로 한 이동 호스트의 위치 정보 탐색 비용(SCNRSB)은 식 (1)이다. n 은 트리 구조에서, 위치 정보 탐색을 시작한 단말 위치 정보 서버가 존재하는 깊이이다. [가정 1]에 따라 이동 호스트의 기지국은 다른 이동 호스트의 위치 정보를 캐쉬하지 않으므로 통신하기 위해서는 위치 정보 탐색 연산이 필요하다. 즉, 이진 트리에서 하위 위치 정보 서버로부터 상위 위치 정보 서버로 탐색해나가며 필요한 탐색 비용의 총합이다.

$$SCNRSB = \sum_{k=1}^n k * 2^{k-1} \tag{1}$$

1-깊이 위치 정보 탐색 트리의 논리적 계층 구조는 이진 트리의 위치 정보 서버의 계층 구조와 같은 탐색 비용을 보인다. 이것은 깊이 1의 링 구조에서 위치 정보 탐색 연산은 이진 트리구조에서 최상위 위치 정보 서버에서의 탐색과 같기 때문이다. 이제 2-깊이 위치 정보 탐색 트리에서 위치 정보 서버의 논리적 계층 구조의

탐색 비용에 대해 알아본다. 식 (2)는 깊이 2에서 링 구조를 결합한 트리 구조가 가지는 위치 정보 탐색 비용(SCRSB-2)이다. 깊이 2에 있는 위치 정보 서버 4개에 링 구조를 적용한 구조이다. 식 (1)과 마찬가지로 n은 위치 정보 탐색이 시작되는 단말 위치 정보 서버가 존재하는 깊이이다.

$$SCRSB-2 = \sum_{k=1}^k (k-1) * 2^{k-2} + (n-1) * 2^{n-2} + n * 2^{n-2} \quad (n \geq 3) \quad (2)$$

3-깊이 위치 정보 탐색 트리에서 위치 정보 탐색 비용(SCRSB-3)과 4-깊이 위치 정보 탐색 트리에서 위치 정보 탐색 비용(SCRSB-4)은 각각 식 (3)과 식(4)이다.

$$SCRSB-3 = \sum_{k=1}^2 k * 2^{k-1+3+n} * 2^{n-1} + (3 * n - 3) * 2^{n-3} \quad (n \geq 4) \quad (3)$$

$$SCRSB-4 = \sum_{k=1}^4 k * 2^{k-1} + 2^{n-4} * \sum_{k=n-3}^{n-4} k + 2^{n-4} * \sum_{k=n-3}^{n-3} k \quad (n \geq 5) \quad (4)$$

그림 3의 그래프를 보면 4-깊이 위치 정보 탐색 트리가 가장 높은 비용을 갖는다. 이것은 4 이상의 깊이에서 링 구조를 갖는 경우, 링 구조에서의 병렬적 탐색 비용이 너무 커지기 때문이다. 2-깊이 위치 정보 탐색 트리를 기반으로 한 위치 정보 탐색 기법의 평균 탐색 비용이 가장 좋은 성능을 갖는다. 따라서 이진 트리를 기반으로 한 위치 정보 서버의 논리적 계층 구조의 경우, 깊이 2에서 링 구조를 결합하는 것이 가장 좋은 성능을 보인다.

3.2 개선된 n-깊이 위치 정보 탐색 트리(Advanced n-level Location information Search Tree)

이 절에서는 이진 트리를 기반으로 한 2-깊이 위치 정보 탐색 트리와 이를 개선된 2-깊이 위치 정보 탐색 트리에 대해 알아본다. 개선된 2-깊이 위치 정보 탐색 트리는 2-깊이 위치 정보 탐색 트리에 비해 더욱 좋은 성능을 갖는다. [정의 2]는 n-깊이 위치 정보 탐색 트리 구조의 성능을 향상시킨 개선된 n-깊이 위치 정보 탐색 트리이다.

[정의 2] **개선된 n-깊이 위치 정보 탐색 트리**

1) n-깊이 위치 정보 탐색 트리이다.

2) 깊이가 nN인 노드들 중 하나의 노드를 루트로 하는 하위 노드들중에서 깊이가 n(N+1)인 노드들은 링 구조를 이룬다.(n, N은 1보다 크거나 같은 자연수이다.)

개선된 n-깊이 위치 정보 탐색 트리의 단말 노드는 LiST에서와 같이 위치 정보 서버로서 각 셀에서 이동 호스트들간의 통신을 담당하는 기지국에게 직접적으로 위치 정보를 제공한다. 하나의 셀에 속해 있는 이동 호스트들은 하나의 기지국을 통해 통신하며, 상위 혹은 동일한 링에서의 이웃하는 위치 정보 서버로부터 다른 이

동 호스트의 위치 정보를 얻는다. 하나의 위치 정보 서버에 속하는 이동 호스트들의 정보는 동일한 위치 정보 서버들에 의해 관리된다. 위치 정보 서버와 기지국은 LiST를 기반으로 하는 논리적 계층 구조를 이룬다. 그림 3은 개선된 n-깊이 위치 정보 탐색 트리 구조를 보여 준다. LS₀을 최상위로 하는 전체 LiST₀에서 모두 21개의 하위 LiST를 가지게 된다. LiST₁에서 위치 정보 서버 LS₁을 최상위로 하는 하위 LiST는 LiST₂를 포함하여 모두 4개이다. 계층적인 링 구조는 위치 정보 서버를 지역적으로 분할함으로써 한 지역내에서의 위치 정보 탐색의 탐색 비용을 줄인다.

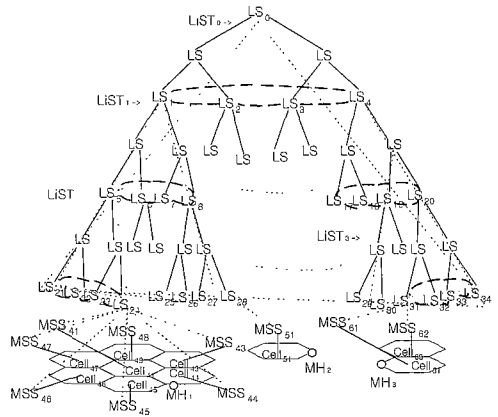


그림 4 2-깊이 위치 정보 탐색 트리의 예

개선된 n-깊이 위치 정보 탐색 트리에서의 위치 정보 탐색 연산의 비용 계산은 링 구조에 속하는 위치 정보 서버에서의 탐색 비용 계산과 단말 위치 정보서버로부터 링 구조의 위치 정보 서버에 이르는 경로에 존재하는 위치 정보 서버에서의 탐색 비용 계산으로 구성된다.

식 (5)는 그림 4와 같은 개선된 2-깊이 위치 정보 탐색 트리에서의 탐색 비용(ASCRSB-2; Advanced Search Cost of Ring Structure in 2-depth Binary tree)을 계산한 식이다.

$$ASCRSB-2 = \sum_{k=1}^{k_1} (2 * 3 * k + 4) * 2^{*(k-1)+k_1} + \sum_{k=1}^{k_2} (k-1) * 2^{k-2} + (k_1-1) * 2^{k_1-2} + k_1 * 2^{k_1-2} \quad (5)$$

(단, k₁ = n을 2로 나눈 나머지 +2, k₂ = n을 2로 나눈 몫-1, n>=4)

식 (6)은 개선된 3-깊이 위치 정보 탐색트리에서의 탐색 비용(ASCRSB-3; Advanced Search Cost of Ring Structure in 3-depth Binary tree)을 계산한 식

이다.

$$ASCRSB-3 = \sum_{k=1}^{k_1} (3*7*k+16)*2^{3*(k-1)+k_1} + \sum_{k=1}^{k_2} k*2^{k-1} + 3*k_1*2^{k_1-3} + (3*k_1-3)*2^{k_1-3} \quad (6)$$

(단, $k_1=n$ 을 3로 나눈 나머지 +3, $k_2=n$ 을 3로 나눈 몫-1, $n \geq 5$)

개선된 n-깊이 위치 정보 탐색 트리에서의 위치 정보 탐색 연산 비용의 계산은 링 구조에 속하는 위치 정보 서버에서의 탐색 비용 계산과 단말 위치 정보서버로부터 링 구조의 위치 정보 서버에 이르는 경로에 존재하는 위치 정보 서버에서의 탐색 비용 계산으로 구성된다. 그림 5의 그래프는 이진 트리로 구조화된 위치 정보 서버의 논리적 구조(SCNSRB), 2-깊이 위치 정보 탐색 트리(SCRSB-2), 개선된 2-깊이 위치 정보 탐색 트리(ASCRSB-2), 3-깊이 위치 정보 탐색 트리(SCRSB-3), 개선된 3-깊이 위치 정보 탐색 트리(ASCRSB-3)에서의 위치 정보 탐색 비용의 비교를 나타낸다.

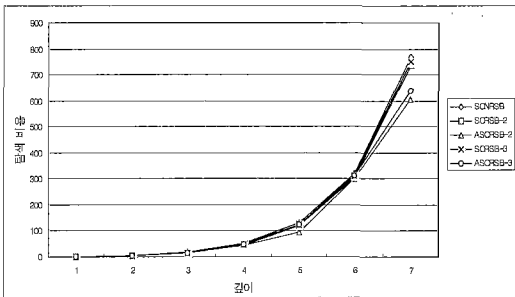


그림 5 개선된 N-깊이 위치 정보 탐색 트리에서의 위치 정보 탐색 비용

그림 5의 그래프에서 보듯이, 개선된 2-깊이 위치 정보 탐색 트리는 2-깊이, 3-깊이 위치 정보 탐색 트리에 비해 좋은 성능을 갖는다. 개선된 2-깊이 위치 정보 탐색 트리는 그림 3에서 2-깊이 위치 정보 탐색 트리와 같이 가장 좋은 성능을 갖는다. 따라서 깊이 2에서의 링 구조 적용이 병렬적 위치 정보 탐색 연산에 가장 적합함을 알 수 있다. 개선된 2-깊이 위치 정보 탐색 트리를 기반으로 한 위치 정보 탐색 연산과 위치 정보 갱신 연산은 4장에서 자세히 설명된다.

4. 위치 정보 탐색/갱신 트리

이동 호스트는 통신을 하기 위해 기지국을 거쳐야 하며, 기지국은 이동 호스트가 통신을 요청한 상대 이동 호스트의 현재 위치를 찾아야 한다. 이때 기지국은 위치

정보 서버가 관리하는 위치 정보 데이터베이스에 위치 정보를 질의하는 위치 정보 탐색 연산을 수행한다. 이동 호스트가 셀 이동을 하면, 이동 호스트의 현재 주소는 위치 정보 데이터베이스에서 갱신되어야 하며, 이것을 위치 정보 갱신 연산이라 한다.

4.1 위치 정보 탐색 기법

이동 호스트가 속해 있는 기지국은 위치 정보 서버에게 통신하려는 상대 이동 호스트의 현재 위치를 위치 정보 서버에게 질의한다. 위치 정보 서버들은 개선된 n-깊이 위치 정보 탐색 트리를 기반으로 논리적으로 구성되며, 개선된 n-깊이 위치 정보 탐색 트리를 이용하여 탐색 연산이 수행된다.

위치 정보 탐색 연산은 동일 깊이에 위치하는 위치 정보 서버에서 수행하는 병렬적 탐색 연산과 상위 깊이에 위치하는 위치 정보 서버에서 수행되는 탐색 연산이 있다.

상위의 위치 정보 탐색 연산은 호출하는 이동 호스트가 속한 기지국의 위치 정보를 관리하는 바로 상위 위치 정보 서버에서 시작된다. 위치 정보를 원하는 기지국은 부모 위치 정보 서버에 이동 호스트의 고유 주소를 보낸다. 만일 위치 정보(현재 주소)가 존재한다면, 기지국에 상대 이동 호스트의 위치 정보를 보내 준다. 위치 정보를 가지고 있지 못하다면 상위 위치 정보 서버에게 현재 주소 질의를 보낸다.

동일 깊이의 위치 정보 서버에서 위치 정보 탐색 연산은 링 구조를 통해 병렬적으로 수행된다. 병렬적 탐색 연산은 링 구조에서 양방향으로 수행된다. 병렬적 탐색 연산은 링 구조의 대칭 위치에 존재하는 위치 정보 서버에서 무조건 끝나게 된다.

그림 4에서 MH_1 이 MH_2 , MH_3 과 통신한다고 가정한다. 먼저 MH_1 은 자신의 기지국이 MH_2 와 MH_3 에 대한 위치 정보를 MSS_{44} 가 가지고 있는지 확인한다. 만일 MSS_{44} 가 위치 정보를 가지고 있지 못하다면, MSS_{44} 는 LS_{30} 에게 위치 정보 질의를 한다. LS_{30} 은 MH_2 와 MH_3 에 대한 위치 정보를 캐시해 놓지 않으므로 위치 정보를 가지고 있지 않고, 다시 상위 위치 정보 서버에게 위치 정보 질의를 보낸다. 하지만 링 구조에 속하지 않은 위치 정보 서버이기 때문에 자신의 상위 위치 정보 서버인 LS_5 에게 위치 정보 질의를 보낸다. LS_5 에는 MH_2 와 MH_3 에 대한 위치 정보가 없으므로 링 구조에 있는 위치 정보 서버에게 위치 정보 질의를 한다. 우선 LS_8 과 LS_6 에게 위치 정보 질의를 한다. 이때 MH_2 의 현재 주소가 LS_8 에서 찾아지고 질의에 대한 결과를 MSS_{44} 에게 보낸다. 하지만 MH_3 의 위치 정보 질의에 대한 연산

이 남아 있으며, 위치 정보 질의는 LS₇에 보내진다. LS₇에도 MH₃에 대한 위치 정보가 없으므로, MH₃에 대한 위치 정보 질의는 LS₁에게 보내진다. LS₁에서 위치 정보 질의는 LS₄와 LS₂에 보내지며, LS₄에는 MH₃에 대한 위치 정보가 있고, MH₃의 현재 주소는 MSS₄₄에게 보내진다.

위치 정보 서버에서의 위치 정보 탐색은 그림 6의 위치 정보 탐색 알고리즘인 함수 Location_Information_Search()을 수행한다. 이 함수는 기지국의 상위 위치 정보 서버에게 기지국의 이름과 Location_Information을 보내고, 개선된 n-깊이 위치 정보 탐색 트리를 기반으로 위치 정보 탐색을 하게 된다. 위치 정보의 탐색이 성공하면 수행을 멈추고, 기지국에게 현재 위치 정보를 알려준다.

```

Location_Information {
    MH_logical_address[];
    MH_physical_address[];
}

Location_Information_Search( MSS, Location_Information )
{
    if(have_the_location_information())
    then {
        send Location_Information to MSS;
        return;
    }
    else {
        if(have_location_information_search_link())
        then {
            query Location_Information to next LSs;
            if(success_search_link())
            then {
                receive Location_Information;
                send Location_Information to MSS;
                return;
            }
            else
                Location_Information_Search(upper_LS,
                Location_Information);
        }
        else {
            query Location_Information to upper LS;
        }
    }
}
    
```

그림 6 위치 정보 탐색 알고리즘

소는 위치 정보 서버에게 반영되어야 한다. 따라서 이동 호스트의 이동후에 빠른 컴퓨팅 속도를 유지하기 위해서는 위치 정보 갱신 연산이 효율적으로 수행되어야 한다. 즉, 탐색 연산의 빠른 수행을 보장해 주면서, 현재 위치 정보의 갱신 연산에 대한 비용이 작아야 한다.

개선된 n-깊이 위치 정보 탐색 트리를 기반으로 한 갱신 연산은 새로 진입한 셀그룹을 관리하는 위치 정보 서버에서 위치 정보 삽입 연산과 이전의 위치 정보 서버에서는 위치 정보 삭제 연산으로 구성된다. 그림 4에서 만일 MH₁이 현재 자신이 속한 셀이 아닌 다른 셀로 이동한다면, MH₁의 위치 정보는 개선된 n-깊이 위치 정보 탐색 트리의 최상 위치 정보 서버 LS₀과 MH₁이 이동전에 속한 셀을 관리하는 기지국사이의 경로에 속한 위치 정보 서버중 링 구조에 속한 위치 정보 서버와 이동한 셀을 관리하는 기지국사이의 경로에 속한 위치 정보 서버중 링 구조에 속한 위치 정보 서버에서 갱신 연산이 발생한다. MH₁이 Cell₄₄에서 Cell₅₁로 셀 이동을 한다고 가정한다. 위치 정보 삭제가 발생하는 위치 정보 서버는 LS₃₀, LS₅, LS₁이고, 위치 정보 삽입 연산이 발생하는 위치 정보 서버는 LS₂₈, LS₈, LS₁이다. 하지만 LS₁의 경우, MH₁에 대한 위치 정보 갱신 연산이 발생된다. 만일 MH₁이 Cell₄₄에서 Cell₆₁로 셀 이동을 한다고 가정한다. 이 경우는 MH₁의 위치 정보 삭제가 발생하는 위치 정보 서버는 LS₃₀, LS₅, LS₁이고, 위치 정보 삽입 연산이 발생하는 LS₃₁, LS₂₀, LS₄에서 발생된다.

위치 정보 갱신 알고리즘에서 위치 정보의 갱신 연산은 이동한 이동 호스트를 관리하였던 기지국이 속한 단말 위치 정보 서버에서 시작된다.

```

Location_Information_Update_upper_LS(MSS, Location_Information )
{
    if(is_Root_LS( ))
    then {
        update Location_Information;
        return();
    }
    else {
        update Location_Information;
        update Location_Information on upper LS;
    }
}
    
```

그림 7 위치 정보 갱신 알고리즘

4.2 위치 정보 갱신 기법

이동 호스트가 이동하면 이동 호스트가 속한 셀 그룹의 변경이 발생된다. 즉, 셀 내에서의 물리적 위치가 변경된다. 셀내에서의 물리적 위치 변경은 이동 호스트의 현재 주소를 변경시키고, 이동 호스트의 변경된 현재 주

4.3 위치 정보 탐색-갱신 기법

이동 컴퓨팅 환경에서의 작업은 계속적으로 위치 변경을 하는 여러 이동 호스트가 서로 메시지를 주고 받으면서 이루어진다. 따라서 이동 호스트의 위치 정보 갱신과 탐색은 단순히 효율한 이동 호스트의 위치 정보

서버뿐만 아니라 호출된 이동 호스트의 이전 위치 정보 서버에서도 이루어진다면, 전체 시스템의 통신 성능은 좋아질 것이다. 호출된 이동 호스트에 대한 탐색 연산의 성공 후, 각 위치 정보 서버에서의 상대 이동 호스트의 위치 정보를 어떻게 갱신해 줄 것인가 하는 문제는 전체 시스템 성능에 큰 영향을 미친다. 호출한 이동 호스트의 위치 정보 서버에서는 호출된 이동 호스트의 위치 정보가 삭제되고, 호출된 이동 호스트가 속한 기지국이 의존하는 위치 정보 서버에서 자신의 위치 정보가 갱신되어야 한다. 이 논문에서 탐색후의 갱신에 대한 특별한 기법이 탐색후에 이동한 이동 호스트의 위치가 변경되면 바로 위치 정보 갱신 연산이 수행된다.

5. 실험

3장에서 [가정 1]을 기반으로 하나의 이동 호스트가 일대일로 통신하는 상황에 대한 탐색 비용과 탐색-갱신 비용에 대한 실험값을 비교 분석한다. 실험을 위한 위치 정보 서버의 논리적 계층 구조로서 개선된 n -깊이 위치 정보 탐색 트리와 이전 트리를 기반으로 한다. 개선된 n -깊이 위치 정보 탐색 트리와 이전 트리의 깊이 H 는 7이며, 위치 정보 서버의 논리적 계층 구조에서 단말 위치 정보 서버들의 수는 2^H 이므로, 128개의 단말 위치 정보 서버가 존재한다. 하나의 단말 위치 정보 서버는 7개의 기지국을 관리하고, 896개의 기지국이 존재하며, 하나의 셀을 하나의 기지국이 관리하므로 896개의 셀이 존재한다고 가정한다. 또한 8개의 단말 위치 정보 서버가 최하위의 링 구조에 속하므로 56개의 셀이 논리적으로 하나의 분할에 속한다고 가정한다.

셀들의 물리적 거리는 셀과 셀사이에 위치하는 중간 셀의 최소 갯수를 셀간 거리라 하고, 논리적 셀의 위치와는 무관하다. 셀간 거리는 이동 호스트의 호출 거리와 이동 거리가 되며, 탐색 비용과 탐색-갱신 비용의 비교 기준이 된다.

통신하는 두 이동호스트가 속한 셀은 무작위로 선출되며, 각 이동호스트가 속한 셀간 거리가 계산된다. 셀간 거리를 기준으로 이동호스트의 위치 정보 탐색 비용, 혹은 갱신 비용이 비교 분석된다.

위치 정보 탐색 비용은 이동 호스트가 호출하는 이동 호스트의 위치 정보를 찾아내기 위해 필요한 비용으로, 호출된 이동호스트의 위치 정보에 대해 캐시기법이나 전향 포인터가 사용되지 않는다고 가정한다. 탐색-갱신 비용은 위치 정보 탐색 비용과 위치 정보 탐색의 성공 후, 호출된 이동호스트가 셀 이동을 통해 발생한 위치 정보의 갱신 비용의 합이다.

5.1 위치 정보 탐색 비용

위치 정보 탐색 비용 측정 실험에서는 두 개의 이동 호스트가 서로에 대한 위치 정보를 캐시하지 않는 상황에서 발생하는 호출에 대해 이루어진다. 호출하거나, 호출된 이동 호스트의 이동은 없다고 가정하므로, 위치 정보 갱신 비용은 0으로 가정한다. 실험은 [11]에서 언급된 트리를 기반으로 한 위치 정보 서버의 계층구조에서의 탐색 기법과 이 논문에서 제안된 개선된 n -깊이 위치 정보 탐색 트리를 이용한 탐색 기법에서의 탐색 비용을 측정한다. Protocol B는 이 논문에서 제안된 개선된 n -깊이 위치 정보 탐색 트리를 기반으로 한 탐색 기법이고, Protocol A는 [11]에서 제안된 트리를 기반으로 한 탐색 기법이다.

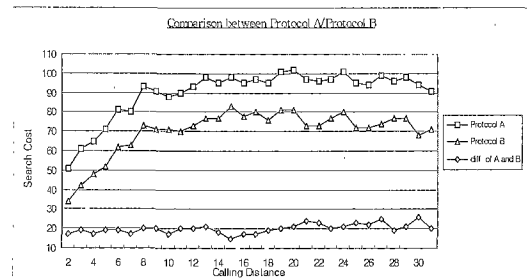


그림 8 위치 정보 탐색 실험

Protocol B에서 각 이동 호스트의 현재 위치(위치 정보)는 각 이동 호스트가 속한 셀을 관리하는 위치 정보 서버와 개선된 n -깊이 위치 정보 탐색 트리의 최상위 위치 정보 서버사이의 최단 경로에 존재하는 위치 정보 서버중 링 구조에 속한 위치 정보 서버가 관리한다. Protocol A에서는 각 이동 호스트의 현재 위치는 각 이동 호스트가 속한 위치 정보 서버와 최상 위치 정보 서버사이의 최단 경로에 존재하는 모든 위치 정보 서버가 알고 있다고 가정한다. 각 이동 호스트의 셀 거리는 인접 셀에 속한 이동 호스트사이의 통신인, 셀 거리 2인 호출부터 이동 호스트간 거리가 셀 거리 31인 호출에 대한 탐색 비용 실험 결과이다. 무작위로 이동 호스트가 선택되고, 이들에 속한 이동 호스트의 통신을 위한 탐색 비용이 계산된다. 호출한 이동 호스트는 호출된 이동 호스트의 위치 정보를 가지고 있는 위치 정보 서버로부터 호출된 이동 호스트의 현재 주소(위치 정보)를 받아야 한다.

그림 8에서의 실험 결과를 보면, 셀 거리가 20인 경우부터, protocol A와 protocol B에서 모두 탐색 비용

의 증가가 없어진다. 이것은 위치 정보 서버의 논리적 계층구조에서 최상위 위치 정보 서버까지의 탐색이 이루어진다는 것을 의미한다. 즉, 셀 거리가 20을 넘을 경우, 기지국으로부터 최상위 위치 정보 서버까지의 위치 정보 탐색 연산이 필요함을 의미한다. 기지국의 수가 증가하면, 위치 정보 서버의 계층 깊이가 깊어진다. 이 경우, 최상위 위치 정보 서버에서 만족되는 위치 정보 탐색 연산이 필요한 셀간 호출 거리도 20을 넘어 증가할 것이다. 그림 8에서 셀간 거리가 증가할수록 protocol A와 protocol B사이의 탐색 비용 차이는 커진다. 즉, 논리적 계층 구조로 인해 발생하던 최상위 위치 정보까지의 위치 정보 탐색이 감소했다는 것을 알 수 있다. 이것은 이동 호스트의 수가 증가할수록, 위치 정보 탐색 트리의 깊이가 깊어질수록 protocol A와 B의 성능 격차가 커진다는 것을 의미한다. 전체적인 위치 정보 탐색 비용은 protocol B가 좋은 효율을 가지는 것을 알 수 있다. Protocol B의 전체 위치 정보 탐색 비용이 감소했으며, 이것은 개선된 n-깊이 위치 정보 탐색 트리가 가지는 논리적 계층 구조의 특징 때문이다.

5.2 위치 정보 탐색-갱신 비용

탐색-갱신 비용은 탐색 비용과 탐색 연산이 성공한 후에 호출된 이동 호스트의 위치 정보를 갱신해야 할 위치 정보 서버의 수, 즉 위치 정보 서버들에게 보내져야 하는 메시지의 수가 된다. 이 논문에서는 임의의 이동 호스트가 다른 임의의 이동 호스트에 대해 호출할 경우 셀 이동한 이동 호스트에 대한 탐색 비용과 호출된 이동 호스트가 셀 이동을 하고 셀 이동한 이동 호스트의 위치 정보에 대한 갱신 비용의 합이 탐색-갱신 비용이 된다. 갱신 비용은 통신할 이동 호스트의 이동 셀 거리를 기준으로 하고, 기지국이나 위치 정보 서버의 갱신 시간을 갱신 비용으로 한다. 탐색-갱신 기법은 [11]에서 제안된 탐색-갱신 기법과 이 논문에서 제안된 개선된 n-깊이 위치 정보 탐색 트리를 기반으로 하는 탐색-갱신 기법을 사용한다. [11]에서 제안된 기법은 완전 위치 정보 갱신(full update)기법으로 그림 9에서 protocol A이다. 완전 위치 정보 갱신 기법은 호출된 이동 호스트가 속한 기지국으로부터 최상위 위치 정보 서버에 이르는 모든 위치 정보 서버에서 발생하는 갱신과 이동 호스트간 통신을 위해 발생하는 위치 정보 탐색으로 이루어진다. Protocol B는 이 논문에서 제안된 개선된 n-깊이 위치 정보 탐색 트리를 기반으로 한 탐색 기법과 갱신 기법을 사용한다. 호출된 이동 호스트가 속한 기지국으로부터 최상위 위치 정보 서버에 이르는 위치 정보 서버의 경로사이에서 링 구조에 속한 위치 정보 서

버에서의 위치 정보 갱신과 위치 정보 탐색으로 이루어진다. 실험은 각각의 셀 거리에 대해 두 가지 탐색-갱신 기법이 30회씩 적용된 비용의 합이다.

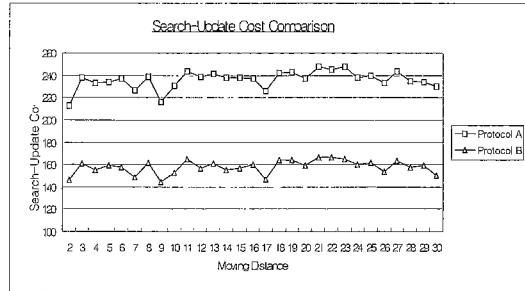


그림 9 위치 정보 탐색/갱신 실험

실험 결과에서 x축은 호출된 이동 호스트의 위치 정보 탐색이 성공적으로 이루어진 후, 이동한 셀간 이동 거리를 의미한다. Protocol A와 B에서 탐색-갱신 비용은 거의 일정하게 나타난다. 이것은 위치 정보 탐색 비용은 이동 호스트의 셀간 호출 거리와 비례했다, 하지만, 위의 실험 결과는 이동 호스트의 셀간 이동 거리를 기준으로 하였기 때문에, 셀 이동 거리에 비례하여 미세하게 증가한다. 프로토콜 A와 B의 실험 결과를 보면, protocol B의 성능이 전체적으로 좋은 것을 알 수 있다. 이것은 개선된 n-깊이 위치 정보 탐색 트리가 가지는 논리적 구조 때문이다.

6. 결 론

이동 컴퓨팅 환경에서 이동 호스트의 현재 주소는 셀 이동으로 인해 가변적이므로, 고유 주소와 현재 주소를 사상시켜 유지되어야 한다. 이동 호스트의 고유 주소와 현재 주소를 사상시켜 주기 위해 많은 기법들이 제안되었지만, 이동 컴퓨팅 환경의 확장성(scalability)과 위치 정보의 분산을 고려한 기법은 논문 [11]에서 제안되었다. 논문[11]은 실제 셀의 물리적 위치에 관계없이 위치 정보 서버의 논리적 계층구조에 따라 탐색 비용과 갱신 비용이 결정되었다. 우리는 논문[11]에서 제안된 위치 정보 서버의 논리적 계층 구조에 의해 발생하는 부가적 탐색 비용을 줄이며, 위치 정보의 분산 관리를 위해 개선된 논리적 계층 구조인 개선된 n-깊이 위치 정보 탐색 트리를 제안하였다. 개선된 n-깊이 위치 정보 탐색 트리는 이동 호스트의 주소 관리에 이동 호스트가 속한 셀의 물리적 위치에 무관한 부가적 탐색 비용을 줄이고,

병렬적 탐색 연산을 가능하게 하며, 위치 정보 서버를 지역적으로 분할하기 위해 계층 구조에 링 구조를 결합시켰다. 또한 개선된 n-깊이 위치 정보 탐색 트리를 기반으로 한 이동 호스트의 위치 탐색 기법과 탐색-갱신 기법을 제안했다.

실험 결과를 보면 개선된 n-깊이 위치 정보 탐색 트리를 기반으로 한 탐색 연산은 위치 정보 서버의 논리적 계층 구조에 의해 발생하는 부가적인 위치 정보 탐색 비용을 줄였을 뿐만 아니라 전체적인 탐색 비용의 감소시켰다. 또한 탐색-갱신 비용에서도 탐색 비용의 증가없이 갱신 비용의 감소를 갖는 탐색-갱신 기법이 실험을 통해 증명되었다.

현재 이 논문에서는 이동 호스트의 셀 이동에 대해 수반되는 핸드 오프 연산이 언급되지 않았다. 또한 개선된 n-깊이 위치 정보 탐색 트리를 기반으로 한 위치 정보 서버의 계층 구조에서 핸드 오프 연산에 대한 연구가 수행될 예정이며, 특히 이동 호스트의 실패 발생에 대비한 결합 포용 기법을 고려한 핸드 오프 연산이 연구될 것이다. 또한 셀의 구성 특성을 고려하여 개선된 n-깊이 위치 정보 탐색 트리에 대해 위치 정보 서버를 분배하는 셀그룹 관리 기법과 이동 호스트의 호출-이동 특성에 따른 동적 탐색 기법이 연구중이다.

참 고 문 헌

[1] D. Dutchman, "Issues in Wireless Mobile Computing," In *Proceedings of 3rd IEEE Workshop on Workstation Operation Systems*, pp. 1-7, Apr. 1992.

[2] G. H. Forman and J. Zahorjan, "The Challenges of Mobile Computing," *IEEE Computers*, 27(6), pp. 38-47, Apr. 1994.

[3] T. Imielinski and B. R. Badrinath, "Wireless Mobile Computing: Challenges in Data Management," *CACM*, pp. 18-28, 1994.

[4] E. Pitoura and B. Bhargava, "Dealing with Mobility: Issues and Research Challenges," *Technical Report CSD-TR-93-070*, Dept. of Computer Sciences, Purdue Univ., 1993.

[5] S. Rangarajan, K. Ratnam, and A. T. Dahbura, "A Fault-Tolerant Protocol for Location Directory Maintenance in Mobile Networks," *FTCS*, pp. 222-238, Aug. 1983.

[6] Tomasz Imielinski, Henry F. Korth, *Mobile Computing*, Kluwer Academic Publishers, 1996

[7] F. Teraoka, Y. Yokote and M. Tokoro, "A Network Architecture Providing Host Migration Transparency," *Proc. ACM SIGCOMM Sympo-*

sium on Communication, Architectures and Protocols, 1991.

[8] H. wada et. al., "Mobile Computing Environment Based on Internet Packet Forwarding," *Proc. of Winter USENIX*, Jan. 1993.

[9] Dhiraj K. Pradhan, P. Krishna, Nitin H. Vaidya, "Recoverable Mobile Environments: Design and Trade-off Analysis," In *26th International Symposium on Fault-Tolerant Computing (FTCS-26)*, June 1996.

[10] B. R. Badrinath, T. Imielinski, and Aashu Virmani, "Locating Strategies for Personal Communication Networks," In *IEEE Globecom '92 Workshop*, Dec. 1992.

[11] P. Krishna, Nitin H. Vaidya, Dhiraj K. Pradhan "Static and Adaptive Location Management in Mobile Wireless Networks," In *Proceedings of Third IEEE Conference on Parallel and Distributed Information Systems*, Sept. 1994.



정 광 식

1993년 고려대학교 컴퓨터학과 학사.
1995년 고려대학교 컴퓨터학과 석사.
1995년 ~ 현재 고려대학교 컴퓨터학과 박사과정. 관심분야는 분산시스템, 이동 컴퓨팅 시스템, 결합포용시스템



유 현 창

1989년 고려대학교 이과대학 컴퓨터학과 졸업. 1991년 고려대학교 대학원 컴퓨터학과 졸업(이학석사). 1994년 고려대학교 대학원 컴퓨터학과 졸업(이학박사). 1995년 ~ 1997년 서경대학교 이공대학 컴퓨터공학과 조교수. 1998년 ~ 현재 고려대학교 사범대학 컴퓨터교육과 조교수. 관심분야는 분산 시스템, 이동 컴퓨팅 시스템, 결합 포용 시스템, 웹기반교육



황 중 선

1978년 Univ. of Georgia, Statistics and Computer Science 박사. 1978년 South Carolina Lander 주립대학교 조교수. 1981년 한국표준연구소 전자계산실 실장. 1995년 한국정보과학회 회장. 1982년 ~ 현재 고려대학교 컴퓨터학과 교수. 1996년 ~ 현재 고려대학교 컴퓨터과학기술대학원 원장. 관심분야는 알고리즘, 분산시스템, 데이터베이스 등임.