

# 이동 컴퓨팅 환경에서 통신 비용을 최소화하는 중복서버 위치 할당 기법의 성능 분석\*

임성화<sup>0</sup>, 김재훈  
아주대학교 정보통신전문대학원  
(hollyfire, jaikim)@adang.ajou.ac.kr

## Performance Analysis of Replicated Servers Allocation Strategies in Mobile Computing

Sung-Hwa Lim, Jai-Hoon Kim

Graduate School of Information and Communication, Ajou University

### 요 약

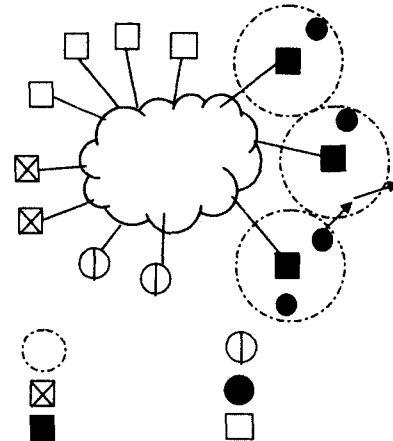
셀룰러 기반의 이동 컴퓨팅환경에서는 이동 호스트(MH)의 위치와 상태관리를 위하여 디폴트 서버기법이 널리 사용되는데, 이동 호스트로 전송되는 데이터는 먼저 디폴트 서버에 문의하여 해당 MH가 위치하는 로컬 서버로 전송된다. 디폴트 서버기법에 기반을 둔 SDN(Single Default Notification)기법에서는 클라이언트가 MH와의 연결요청을 하면 쿼리 서버를 통해 해당 디폴트 서버에 MH의 위치 및 상태를 문의한 후 통신이 이루어진다. 그러나 쿼리횟수가 많고 디폴트 서버와의 거리가 멀거나 기지국의 수가 많은 경우 디폴트 서버와의 통신 오버헤드가 커지며, 디폴트 서버에 결함이 발생할 경우 해당 MH와의 연결이 불가능해진다. 디폴트 서버를 여러 개로 중복 시킬 경우 위와 같은 문제점을 경감시키고 디폴트 서버와의 통신 비용을 줄일 수 있다. 본 논문에서는 통신 환경에 따른 디폴트 서버의 최적의 중복 개수와 중복 위치를 구하는 방법을 제시하였고, 시뮬레이션을 통하여 수학적 방법이 유용함을 보였다.

### I. 서론

최근 이동 통신과 컴퓨터 하드웨어 기술의 발전으로 HPC나 팜탑과 같은 휴대용 컴퓨터를 이용한 이동 컴퓨팅이 발달하게 되었으며, 이동 컴퓨팅의 사용자들의 수가 많아짐에 따라 관리해야 하는 사용자 데이터의 양이 많아짐에 따라, 원하는 순간에 적은 시간 지연으로 필요한 정보를 얻을 수 있는 능력이 필요하게 되었다. 셀룰러 망에서의 이동 호스트의 통신을 위해서는 MH들의 위치 정보를 관리해야 하는데, 하나의 MH의 위치 정보를 하나의 전담 서버(디폴트 서버)에서 관리하는 디폴트 서버 기법이 가장 널리 사용된다[3]. MH가 다른 기지국으로 이동할 경우 디폴트 서버의 위치 정보가 갱신 되는 방법으로 MH의 정보가 관리 되는데, 다른 클라이언트로부터 MH로 통신 요구가 발생할 경우 디폴트 서버를 통해 MH가 위치하는 기지국과 통신을 할 수 있다. MH당 하나의 디폴트 서버를 사용하게 되면 MH에 대한 통신 요구 및 MH의 이동 시 모든 정보가 하나의 디폴트 서버를 거쳐야 하므로, 특정 MH에 대한 통신 요구가 빈번한 환경에서는 해당 디폴트 서버와의 통신이 증가하고, 디폴트 서버에 대한 과부하가 예상된다. 특히 디폴트 서버가 결함에 의해 정지했을 경우 통신이 불가능한 상황도 가능한데, 이 경우 디폴트 서버를 여러 개로 중복 시킴으로써 위와 같은 문제점들을 경감 시킬 수 있다[9]. 그러나 디폴트 서버를 중복 시킬 경우 MH의 이동 성향과 쿼리 요청의 빈도, 중복 디폴트 서버의 할당 위치와 중복 개수에 따라 통신 비용이 증가할 수 있다. 본 논문에서는 대표적인 디폴트 서버 기법 중 하나인 SDN(Single Default Notification) 기법에서 셀의 개수와 통신 성향에 따라 디폴트 서버의 최적 중복 개수와 중복 위치를 구하는 방법을 제시하고 시뮬레이션을 통하여 수학적 방법의 유용성을 검증하였다.

### II. 단일 디폴트 서버 기법[5]

모바일 컴퓨팅에서는 이동성을 지원하기 위해 사용자의 위치 및 상태에 관한 정보를 서버가 실시간으로 제공 할 수 있어야 한다. 사용자의 위치 정보를 관리하는 방식에는 MH의 위치 이동 시 전체 서버에 브로드캐스트 해주는 BN(Broadcast Notification)기법과 디폴트 서버를 두어 이동호스트의 정보를 지속적으로 관리하게 하는 DF(Default Forwarding) 기법이 있다[4,5]. DF 기법(디폴트 서버 기법)에 기반을 둔 통신 프로토콜로는 SDN(Single Default Notification) 또는 WDN(double Default Notification)기법이 있는데, [그림 1]에서와 같은 네트워크 구조를 갖는다[5]. MH가 e 지역에서 f 지역으로 이동할 때 기지국 f는 디폴트 서버(m(i))에 알려주며, 클라이언트 c가 i와 통



신을 시도할 경우 쿼리 서버 a나 b를 통해 디폴트 서버에 문의를 하게 되는데, 디폴트 서버는 이 경우 MH가 위치하는 기지국으로 연락을 해서 클라이언트 c와 MH i 간에 통신이 이루어지도록 한다.

[그림 1] SDN 기법의 네트워크 구조[5]

기지국은 셀 내에 위치하는 MH들에게 여러 지 정보를 얻기 위해 쿼리 패킷을 보낼 수 있는데, MH의 정보에 관한 쿼리의 종류는 다음과 같다.

L 쿼리 : MH의 위치에 대한 정보를 갖는다.

E 쿼리 : MH의 상태를 나타내며 즉 MH가 켜졌는지에 대한 정보를 포함한다.

D 쿼리 : MH에 대한 여러 가지 정보를 갖는다.

SDN 기법은 MH가 이동할 경우(migration)와 L 쿼리, E 쿼리, 그리고 D 쿼리시에 다음과 같은 통신 부하를 갖는다.  $C_{Migration}$ 은 MH가 다른 셀로 이동했을 경우의 통신 비용이며,  $C_L$ ,  $C_E$ ,  $C_D$ 는 각각의 쿼리에 따른 통신 비용이다. 또  $D_{f,i}^{(M)}$ 는 기지국 f와 디폴트 서버 m(i) 간의 통신 비용이고,  $D_{a,i}^{(Q)}$ 는 쿼리서버 a와 디폴트서버 m(i)간의 통신 비용이며,  $H_{a,i}$ 는 쿼리서버 a와 기지국 f 간의 통신 비용이다(클라이언트와 디폴트서버, 기지국과 MH 간의 통신비용은 생략되었다). 실제로  $n \times n$ 의 n-grid 구조에 적용하였을 때의 통신비용 D와 H의 평균값은 다음과 같이 계산할 수 있다[5].

\* 이 논문은 2000년도 한국학술진흥재단의 지원에 의하여 연구되었음.  
(KRF-2000-041-E00293)

$$C_{Migration} = D_{f,i}^{(M)}$$

$$C_L = 2D_{a,i}^{(Q)}$$

$$C_E = D_{a,i}^{(Q)} + D_{f,i}^{(M)} + H_{o,f}$$

$$C_D = D_{a,i}^{(Q)} + D_{f,i}^{(M)} + H_{a,f}$$

$$\bar{H} = \frac{2(n-1)(n+1)}{3n}$$

$$D^{(Q)} = D^{(M)} = \begin{cases} \frac{(n+1)(n-1)}{2n} : n \text{은 홀수} \\ \frac{n}{2} : n \text{은 짝수} \end{cases}$$

또한 [이진 트리 구조에서의 통신비용 D]와 H의 평균값은 다음과 같이 구할 수 있다(n은 트리의 높이) [5].

$$\bar{H} = 2(n-2) + 2^{2-n}$$

$$D^{(Q)} = D^{(M)} = n - 1$$

### III. 중복 디폴트 서버 기법

SDN 기법에서는 MH에 call을 할 경우 해당 MH의 디폴트 서버에 먼저 문의를 하여 MH가 위치하는 셀(cell)의 위치 정보를 받은 후, 기지국을 통하여 통신이 이루어진다. 이 경우 디폴트 서버가 하나이므로 클라이언트 또는 쿼리 서버가 디폴트 서버와 멀리 떨어져 있을 경우, 통신 비용이 무시할 수 없을 정도로 증가할 수 있으며, 최악의 경우 가까이 위치한 MH로의 통신을 위해서도 멀리 떨어진 디폴트 서버와의 통신이 선행되어야 한다. 디폴트 서버와 기지국, 그리고 쿼리 서버간의 통신비용을 줄이기 위해서는 디폴트 서버를 중복시키는 방법이 효과적이다[9]. 반면에 서버를 중복시킬 경우 중복 투명성을 제공해야 하며, 중복에 따른 오버헤드를 부담해야 하는 단점이 있다. 서버 중복을 위해서 본 논문에서는 하나의 데이터가 저장 또는 갱신되었을 경우 시스템에서 자동적으로 중복된 서버들의 데이터를 저장 및 갱신 시키도록 가정하였다. 기존의 SDN 기법을 개선하여 디폴트 서버를 중복시킬 경우의 통신비용( $C_{Migration}$ ,  $C_L$ ,  $C_E$ ,  $C_D$ )은 다음과 같다[9].

$$C_{Migration} = D_{f,i}^{(M)} + U$$

$$C_L = 2D_{a,i}^{(Q)}$$

$$C_E = D_{a,i}^{(Q)} + D_{f,i}^{(M)} + H_{o,f}$$

$$C_D = D_{a,i}^{(Q)} + D_{f,i}^{(M)} + H_{a,f}$$

디폴트 서버를 중복시킬 경우 네트워크에서 중복 서버의 위치를 효과적으로 할당하는 것은 매우 중요하다. 디폴트 서버의 중복 개수를 m이라고 할 때, 전체 네트워크를 m개로 균등히 나누고, 각 나눠진 지역에서의 중심점에 중복서버를 할당함으로써 중복 디폴트 서버의 위치를 최적으로 할당할 수 있다. 각각의 나눠진 지역 내의 임의의 점에서 그 지역의 중심점까지의 평균 거리는 다른 점까지의 평균 거리보다 짧으므로, 각 나눠진 지역의 중심점에 중복서버를 할당하는 것이 통신비용을 최소화시킬 수 있다. 만약 중복 디폴트 서버를 지역의 중심점이 아닌 부분에 할당할 경우, 중심점에 할당한 경우에 비하여 디폴트 서버와 높은 비용으로 통신해야 하는 기지국들(더 멀리 떨어진 기지국)의 수가 많아진다. 즉 디폴트 서버의 위치가 중심에서 벗어날수록 디폴트 서버와의 통신 비용에 대하여 손해를 보는 기지국의 수가 이득을 보는 기지국의 수보다 많아지므로, 중심에 디폴트 서버를 할당하는 것이 효과적이다. [Algorithm 1]에서 중복 서버 위치 할당 알고리즘을 나타내었다.

[Algorithm 1] 최적 중복서버 위치 할당 알고리즘

- Step 1. Divide the network into the same number of sub-networks as the default servers with the same size.

Step 2. Find the centroid in each sub-network.

Step 3. Allocate a default server on each centroid.

서버의 개수를 n, 디폴트 서버의 중복 개수를 m이라 할 때, n x n의 캠퍼스 네트워크에서의 통신 비용 H,  $D^{(Q)}$ , 그리고  $D^{(M)}$ 의 평균값과 디폴트 서버 갱신 비용 U는 다음과 같이 구할 수 있다. (단 m은  $k^2$ , k는 양의 정수).

$$H = \frac{2(n-1)(n+1)}{3n}$$

$$D^{(Q)} = D^{(M)} = \frac{n}{2} \sqrt{\frac{m}{m+1}}$$

$$U = n(1 - \sqrt{\frac{m}{m+1}})$$

[그림 2]는 n=7인 n-grid 네트워크에서 [Algorithm1]을 적용하여 디폴트 서버를 중복시킬 예를 나타내고 있다.

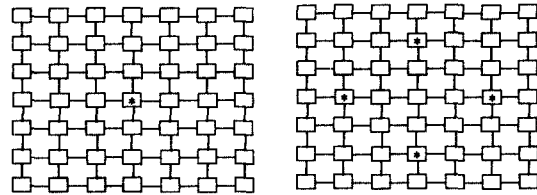
이진 트리 구조 네트워크에서의 중복 서버의 위치도 [Algorithm1]에 기반을 두고 할당될 경우 비용을 최소화시킬 수 있다. 결과적으로 중복 서버의 개수가 n일 경우, 중복 서버의 위치는 이진 트리에서 루트에서  $(\log_2 N)$ -번째 깊이의 자녀 노드들에 할당된다. [그림 3]은 트리의 깊이가 n인 이진 트리 네트워크에서 중복 디폴트 서버의 수가 2, 4인

경우의 할당 예를 나타내고 있다. 서버의 개수를 n, 디폴트 서버의 중복 개수를 m이라 할 때, 깊이가 n인 이진 트리 네트워크에서의 통신 비용 H,  $D^{(Q)}$ ,  $D^{(M)}$ 의 평균값과 디폴트 서버 갱신 비용 U는 다음과 같다 (단 m은  $2^k$ , k는 양의 정수).

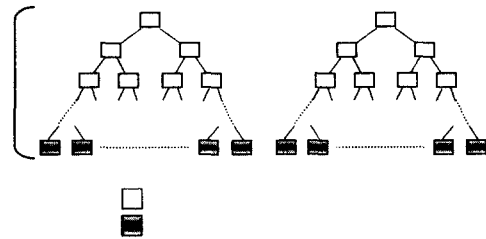
$$\bar{H} = 2(n-2) + 2^{2-n}$$

$$D^{(Q)} = D^{(M)} = n - \log_2 m - 1$$

$$U = 2m - 2$$



[그림 2] n=7인 경우, n x n 네트워크에서 중복 디폴트 서버의 수(m)가 2,4일 때의 최적의 위치 할당 예



[그림 3] 이진 트리 네트워크에서 중복 디폴트 서버의 수(m)가 2,4일 때의 최적의 위치 할당 예

### IV. 성능 평가

본 논문에서는 단일 디폴트 서버를 사용했을 경우와, 디폴트 서버를 m개 중복시켰을 경우의 통신비용을 분석함으로써 성능 평가를 하였다. n-grid 네트워크 및 이진 트리 구조에서 중복 디폴트 서버의 성능 평가를 위하여 디폴트 서버를 중복하지 않은 경우와 m개 중복했을 경우 발생하는 통신비용에 대하여 비교하였다. 전체 통신비용은 다음과 같이 구할 수 있으며,  $C_{LoadD}$ 는 쿼리에 대한 비용이고,  $C_{Migration}$ 은 쿼리 횟수와 MH의 위치 이동 횟수의 비율이고,  $C_{Migration}$ 은 MH의 위치 이동에 따른 통신비용을 의미한다.

$$Cost = C_{L \text{ or } E \text{ or } D} \times (Query / Migration) + C_{Migration}$$

n-grid 네트워크에서 디폴트 서버를 m개 중복시킬 경우 L 쿼리, E 쿼리 D 쿼리에서의 MH의 위치이동 당 통신비용  $Cost_L$ 과  $Cost_{E \text{ or } D}$ 는 다음과 같다.

$$Cost_L = \frac{n}{\sqrt{m}} (Query / Migration) + \frac{1}{2\sqrt{m}} + n(\sqrt{m} - \frac{1}{\sqrt{m}})$$

$$Cost_{E \text{ or } D} = \left\{ \frac{n}{\sqrt{m}} + \frac{2(n-1)(n+1)}{3n} \right\} (Query / Migration) + \frac{1}{2\sqrt{m}} + n(\sqrt{m} - \frac{1}{\sqrt{m}})$$

$Cost_L$ 과  $Cost_{E \text{ or } D}$ 를 m에 관한 함수 C(m)으로 나타내고, m에 대하여 미분을 하여

$$\frac{dC(m)}{dm} = 0, \quad \frac{d^2C(m)}{dm^2} < 0$$

을 만족하는 m을 구하면, m=(Query/Migration)-1인 순간, cost는 최소가 된다.

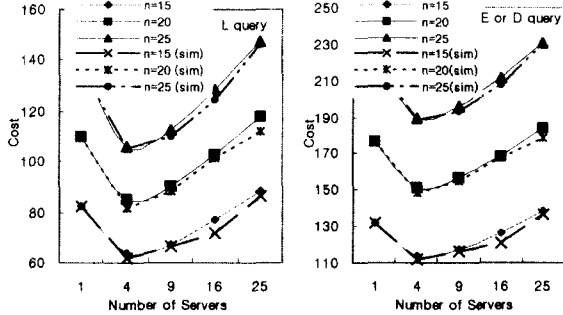
[그림 4]은 n-grid 네트워크에서 (Query/Migration) 비율을 5로 고정시키고 n의 값이 각각 15,20,25인 경우, m의 변화에 따른 L 쿼리와 E 쿼리의 통신비용을 나타내었다. 위에서 이론적으로 구한 결과의 유용성을 검증하기 위하여 시뮬레이션에 의한 결과와 비교하였다. 사용된 시뮬레이션 환경은 다음과 같다.

- n x n 네트워크에서 [algorithm1]에 의하여 m개의 디폴트 서버를 할당한다.
- 쿼리가 발생했을 경우 디폴트 서버가 할당된 네트워크에서 기지국과 디폴트서버와의 통신비용(홉수)를 측정하여 평균값(D)을 구한다.
- MH의 이동(migration)시 디폴트 서버간의 갱신 비용(홉수)이 발생하며, 한번 발생 시의 갱신비용(U)을 측정한다.
- 유선 호스트간의 통신비용만을 고려하며, MH와 기지국간의 무선

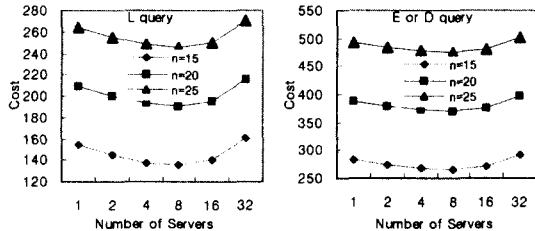
통신비용, 디플트 서버를 중복 시킬 경우의 서버의 유지 비용은 고려하지 않는다.

● 디플트 서버 및 기지국의 결합은 고려하지 않는다.  
 [그림 4]의 결과에서와 같이 이론적 최적화 중복개수  $m = (Query / Migration) - \frac{1}{2} \approx 4.5$  근처인  $m=4$  인 순간에서 통신 비용이

최소화 됨을 알 수 있다.  $m$ 이 커짐에 따라  $D$  값이 작아지므로 전체 통신 비용이 줄어드나,  $U$  값이 상승하므로  $m$ 의 최적점을 넘으면 전체 통신 비용이 오히려 늘어남을 알 수 있다. 즉 최적의 중복 개수는 쿼리와 MH의 위치 이동 횟수의 비율에 따라 결정됨을 알 수 있다.



[그림 4] n-grid 네트워크에서 L 쿼리, E 쿼리에 대한 디플트 중복 개수에 따른 통신 비용의 변화(Query/Migration = 5, n=15,20,25)



[그림 5] 이진 트리 네트워크에서 L 쿼리, E 쿼리에 대한 디플트 중복 개수에 따른 통신 비용의 변화(Query/Migration = 5, n=15,20,25)

깊이가  $n$  인 이진트리 네트워크에서 디플트 서버를  $m$  개 중복시킬 경우 L 쿼리, E 쿼리, D 쿼리에서의 MH의 위치이동 당 통신비용은 다음과 같다.

$$Cost_L = (2n - 2 \log_2 m - 2)(Query / Migration) + n - \log_2 m + 2m - 3$$

$$Cost_{E \text{ or } D} = (4n - 2 \log_2 m - 6)(Query / Migration) + n - \log_2 m + 2m - 3$$

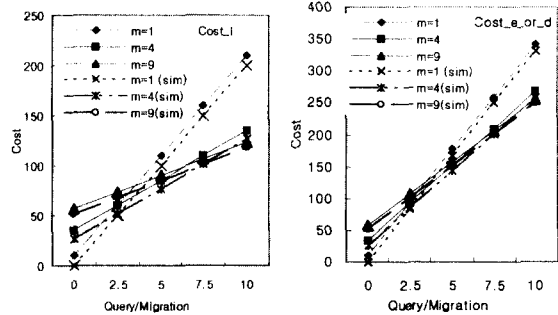
Cost<sub>L</sub>과 Cost<sub>E or D</sub>를 다음과 같이  $m$ 에 관한 함수로 나타내고  $m$ 에 대하여 미분을 하여  $\frac{dC(m)}{dm} = 0$ ,  $\frac{d^2C(m)}{dm^2} < 0$  을 만족하는  $m$  값을 구하면,

$m = \frac{2(Query / Migration) + 1}{2 \ln 2}$  인 순간 cost는 최소가 된다. [그림 5]는

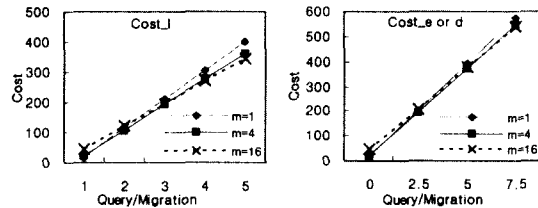
이진 트리 네트워크에서 (Query/Migration) 비율이 5로 고정되고,  $n$ 의 값이 각각 15,20,25인 경우,  $m$ 의 변화에 따른 L 쿼리와 E 쿼리에 대한 통신 비용을 나타낸다. 결과에서 볼 수 있듯이  $m = \frac{2(Query / Migration) + 1}{2 \ln 2}$  (즉,  $m \approx 8$ )에서 최적의 통신 비용을 나타내었다.

다. [그림 6]은 n-grid 네트워크에서  $n$ 이 20으로 고정되고, 디플트 서버를 각각 1, 4, 8 개로 중복 시킨 경우, (Query/Migration) 비율의 변화에 따른 L 쿼리와 E 쿼리에 대한 통신 비용을, 수식에 의한 결과와 시뮬레이션에 의한 결과로 나타내었다. 쿼리에 비해 이동(migration)이 잦은 환경에서는 디플트 서버의 갱신이 자주 일어나므로, 동기화를 위해 중복된 디플트 서버를 모두 갱신하는 횟수가 증가한다. 반면 Query/Migration 비율이 증가하면, 디플트 서버와 쿼리 서버 및 MH의 기지국과의 통신의 횟수가 증가하는데, 디플트 서버를 중복시키면 디플트 서버와의 통신 비용(D)가 감소하므로 전체 통신 비용을 감소시킬 수 있다. 그러므로 쿼리가 잦고 이동이 적을수록 디플트 서버의 중복 개수를 증가시키는 것이 효과적이다. [그림 7]은 이진 트리 네트워크에서  $n$ 이 20으로 고정되고, 디플트 서버를 각각 1, 4, 16 개로 중복 시킨 경우, (Query/Migration) 비율의 변화에 따른 L 쿼리와 E 쿼리에

대한 통신 비용을 나타낸다. n-grid 네트워크의 경우와 같이 이진 트리에서도 쿼리 횟수에 따른 이동호스트의 위치 이동 비율이 증가함에 따라 최적 중복 개수가 증가함을 알 수 있다.



[그림 6] n=20인 n-grid network에서 m의 값과 (Query/Migration) 비율의 변화에 따른 통신 비용의 변화의 수학적 분석과 시뮬레이션



[그림 7] 깊이가 20인(n=20)인 이진 트리 네트워크에서 m의 값과 (Query/Migration)비율의 변화에 따른 통신 비용의 변화

V. 결론

디플트 서버를 중복 시킬 경우 쿼리에 대한 통신 비용은 감소하였으나 MH가 다른 기지국으로 이동할 경우(migration) 중복된 디플트 서버들에 대한 갱신 작업에 필요한 오버헤드가 추가적으로 발생하므로 중복 디플트 서버의 통신 환경에 따른 최적의 중복 개수와 중복 디플트 서버의 위치를 결정하는 일이 중요하다. 본 논문에서는 최적의 중복 디플트 서버 위치 할당 알고리즘을 제안하고, Query/Migration 비율과 기지국의 수에 따른 최적의 디플트 서버 중복 개수를 구하고 시뮬레이션을 통하여 유용성을 검증하였다. 실현 결과에 따르면 디플트 서버의 중복은 MH의 위치 이동에 대한 쿼리의 비율이 큰 경우 즉, 사용자의 이동이 잦지 않은 환경에서 적합함을 알 수 있었다. 그러므로 통신환경의 특성에 따라(이동 패턴, 통신 요청 횟수)에 따라 디플트 서버의 중복 여부 또는 중복 정도가 적합하게 결정되어야 한다. 차세대 통신에서 지원될 글로벌 로밍과 같이 넓은 환경에서 특정한 상황(예:Query/Migration 비율이 높음)을 갖는 사용자들에게는 중복 디플트 서버기법이 매우 효과적일 것으로 예상된다.

참고문헌

- [1] G. H. Forman and J. Zahorjan, "The Challenges of Mobile Computing," IEEE Computer, V 27, N 4, pp. 38-47, April 1994.
- [2] E. Pitoura and G. Samaras, "Data Management for Mobile Computing," Kluwer Academic Publishers, 1997.
- [3] M. Satyanarayanan, "Fundamental challenges of mobile computing," ACM Symposium on Principles of Distributed Computing, 1995 (PODC'95 invited lecture).
- [4] F. Terakoa, Y. Yokote and M. Tokoro, "A Network Architecture Providing Host Migration Transparency," Proc. of ACM-SIGCOMM'91, pp. 45-65, 1991.
- [5] M. Tsukamoto, R. kadobayashi and S. Noshio, "Strategies for Query Processing in Mobile Computing," Mobile Computing, Kluwer Academic Publishers, pp. 595-620, 1996.
- [6] 임정식, "이동 컴퓨팅 연구동향," 정보과학회지, 제 16 권, 제 1 호, pp. 7-11, 1998년 1월.
- [7] B. Awerbuch and D. Peleg, "Concurrent online tracking of mobile users," Proc. ACM SIGCOM '91 pp. 221-233, 1991.
- [8] A. Tanenbaum, "Distributed Operating Systems," Prentice Hall, 1995.
- [9] 임성화, 김재훈, 김성수, "모바일 컴퓨팅 환경에서 중복 디플트 서버를 이용한 쿼리 프로세싱 기법의 성능 분석," 한국 통신학회 논문지, 제 25 권 8 호, 2000.
- [10] E. Pitoura and G. Samaras, "Locating Objects in Mobile Computing," IEEE Transactions on Knowledge and Data Engineering, 2000.