

무선 ATM 망에서 개선된 COS 탐색 알고리즘을 이용한 최소 자원예약 핸드오버 기법

정회원 최승권*, 안상규**, 이병수***, 조용환*

Handover Mechanism on Minimal Resource Reservation Using the Advanced COS Discovery Algorithm in Wireless ATM Networks

Seung-Kwon Choi*, Sang-Kyu An**, Byeong-Su Lee***, Yong-Hwan Cho* *Regular Members*

요 약

본 논문에서는 무선 ATM 망을 위한 DH-BT COS 탐색 알고리즘과 최소자원예약 핸드오버 기법을 제안하였다. 현재 연결지향적인 무선 ATM에서 빠르고 효율적이며 연속적인 핸드오버 기법은 매우 중요한 기술이다.

시뮬레이션 결과 제안한 DH-BT COS 탐색 기법은 현존하는 COS 탐색 기법에 비해 새로운 부분 경로 설립시 회선 재사용율에서 우수한 성능을 보였으며 적절한 최소 홉 수를 나타내었다. 또한 핸드오버 처리시간과 대역폭 사용은 제안한 핸드오버 기법이 HOS나 VCT 등과 비교하여 볼 때 가장 우수한 것으로 확인되었다.

ABSTRACT

In this paper, an advanced DH-BT COS discovery algorithm and a handover mechanism over WATM networks are proposed. The proposed handover mechanism has the ability to provide adaptive handover for WATM services. The ability to provide a fast, efficient and continuous handover mechanism is a key technology of a connection-oriented WATM.

Simulation results show that the proposed DH-BT COS discovery scheme outperforms existing COS discovery algorithms in providing good circuit reuse efficiency and reasonable minimum hop counts to establish a new partial path.

Also, in the view of handover processing time and bandwidth use, the proposed handover mechanism is compared to the HOS, and the VCT handover mechanism. The proposed handover mechanism shows better performance than the HOS mechanism, and the VCT mechanism.

I. 서론

현재의 무선통신망은 음성과 저속 데이터 서비스를 위주로 한 셀룰러 이동전화 및 무선 패킷 데이터 망, 그리고 무선 LAN(Local Area Network) 등을 주축으로 성장하고 있으며, 이와 같은 무선통신망의 발전에 따라 무선 멀티미디어 서비스에 대한

요구도 증대할 것으로 예상된다. 이에 따라 5GHz, 17GHz, 60GHz 등의 고주파 대역을 이용하여 궁극적으로는 155Mbps의 데이터 전송률을 제공하면서 ATM(Asynchronous Transfer Mode) 통신망 서비스를 제공하기 위한 무선 ATM(Wireless ATM)이 제안되었다.

무선 ATM 망에서의 중요한 문제중 한 가지는 현재의 기지국과 통신을 하던 사용자가 통신 중 다

* 충북대학교 컴퓨터공학과(yhcho@cbuucc.chungbuk.ac.kr)

** 한국 에너지 기술연구소

*** (주)중의메디칼

논문번호 : 99063-0218, 접수일자 2월 18일

른 셀로 이동하는 경우 핸드오버 처리를 통해 이전의 기지국과 새로운 기지국에 대해 적절한 호 처리를 하고 채널을 할당해 주어야 한다는 것이다. 핸드오버 처리가 적절치 못하면 망에 많은 부담을 주게 되어 전송중인 정보들 세시간에 전송하지 못하거나, 또는 정보를 올바르게 전달할 수 없게 되므로 무선 ATM에서 핸드오버는 매우 중요하다.

기존에 제안된 핸드오버에 대한 연구로는 R. Yuan 등이 제안한 HOS(HandOver Switch) based scheme^[1], S. Acampora 등이 제안한 VCT(Virtual Connection Tree)^[2], M. Veeraraghavan 등이 제안한 BAHAMA^[3], Bora A. Akyol 등이 제안한 NCNR(Nearest Common Node Rerouting) 기법^[4] 등이 있다.

그러나, 기존의 핸드오버 기법들을 살펴보면 HOS 기법의 경우는 HOS의 선택에 따라 최적화된 경로에 접근하게 되지만 시그널링 부하가 많아지고, 셀 손실 방지를 위한 버퍼링이 많아 핸드오버 시간이 길어지므로 호의 블로킹 확률이 높아지는 단점이 있고, VCT 기법의 경우는 연결 설정시 망을 가상트리 형태로 구성하여 VCT 내에서는 HAC(Handover Admission Control)가 필요 없이 신속한 핸드오버가 가능하지만 사용되지 않는 브랜치에도 자원을 할당하여 많은 자원의 낭비를 가져오게 되는 단점이 있다.

본 논문에서는 이러한 단점을 보완하고, 무선 ATM 망에서 신속하고 신뢰성 있는 핸드오버를 위해 요구되는 최적의 COS(CrossOver Switch) 탐색 기법과 효율적인 핸드오버 기법을 제안하고자 하였다. 이를 위해 C. K. Toh가 제안한 COS 탐색 알고리즘들^[5]을 중심으로 분석하여 LS(Location Server), CS(Connection Server)가 없이 분산처리 방식으로 이루어지는 개선된 DH-BT COS 탐색 알고리즘을 제안하고 이를 Random, Star, Tree 망을 가정하여 최선 재사용률과 COS까지의 최소 홉 수를 시뮬레이션을 통해 기존 기법에 비해 성능이 개선되었음을 입증하였다.

또한, 개선된 COS 탐색 알고리즘을 사용하면서 최소자원만을 예약하여 신속한 핸드오버를 가능하게 하는 최소 자원예약 핸드오버 기법을 제안하였다. 이는 핸드오버를 클러스터내, 클러스터간 핸드오버로 구분하여 핸드오버를 수행하는 방법이며, 클러스터간의 핸드오버 경우에 한해 2-3개의 인접 클러스터 경계 셀의 경로만을 예약함으로써 최소의 자원으로 핸드오버 처리시간을 감소시킬 수 있는 효

율적인 방법이 될 것이다. 이를 위해 기존에 제안한 대표적인 핸드오버 기법인 HOS 기법과 VCT 기법 및 그 밖에 제안된 기법들에 대해 살펴보고, 문제점을 분석하였으며, 최소 자원예약 기법에 대한 모델링과 시뮬레이션을 통해 핸드오버 처리시간 분석과 대역폭 점유율 등을 분석하여 그 성능을 입증하도록 하였다.

II. 개선된 핸드오버 기법 제안

1. 기존 핸드오버 기법의 문제점

무선 ATM에서는 주파수 자원을 효율적으로 사용하기 위해 마이크로/피코 셀의 개념을 도입하여 셀의 반경을 작게 하였으며, 이는 주파수 재사용 거리를 줄임으로써 무선 자원의 용량을 증대시키기 위함이다^[2]. 여러 개의 마이크로/피코 셀은 하나의 ATM 교환기에 연결되어 하나의 클러스터를 구성한다. 셀의 반경이 감소함에 따라 동일한 이동에 대하여 기존의 셀룰러 망에 비해 핸드오버 요구가 상대적으로 빈번하게 발생하게 된다. 하나의 ATM 교환기는 마이크로/피코셀을 담당하는 여러 개의 기지국과 연결되며 경로설정을 위한 정보 등을 포함한 데이터 베이스에 접속한다.

클러스터의 개념을 기반으로 핸드오버를 고려할 때 핸드오버는 두 가지의 형태를 고려할 수 있다^[4]. 하나는 클러스터 내의 핸드오버이고 다른 하나는 클러스터 간의 핸드오버로 구분할 수 있다. 클러스터 내의 핸드오버는 같은 클러스터 내의 셀 사이에 단말이 이동할 경우에 발생한다. 이 경우 해당 클러스터의 ATM 교환기는 이전 기지국으로의 연결을 해제하고, 사용자가 새로 이동한 셀의 기지국으로 새로운 연결을 설정한다. 클러스터 내에서 발생하는 핸드오버는 무선 접속점을 전환만 해주고 경로 설정은 요구하지 않으므로 제안하는 기법에서는 더 이상 이를 고려하지 않는다. 그러나 클러스터 사이에서 단말이 이동할 경우의 핸드오버 절차는 교환기 사이의 연결 경로를 재설정하는 절차를 필요로 한다.

이미 존재하는 기존 연결을 부분적으로 재사용하는 방법은 완전한 경로를 다시 설정할 필요가 없으며, 따라서 이는 완전한 경로를 다시 설정하는데 걸리는 시간을 단축한다. 또한 교환기 사이의 새로운 링크에 경로를 설정하기 위한 신호 메시지의 발생이 줄어들므로 수행 절차를 단순화하고 셀 손실을 감소시킬 수 있다^[6-8]. 이러한 방법에서는 COS를 결

정하여 주어야 한다.

일부 기존의 COS 선택 방법들은 경로 재설정 절차가 빠르고 간단하게 수행하기 위하여 ATM 교환기 사이의 연결을 트리 형태로 제한하였다. 이때 이전 경로상의 교환기 중에서 핸드오버 수행과 관련된 두 기지국과 가장 가까운 공통 노드가 COS로 결정한다. 트리 구조의 백본 망에서는 이러한 COS는 효율적인 경로를 설정하게 된다. 그러나, 트리 형태의 망은 여러 가지 문제점을 갖는다. 트리 형태로 구성된 망은 상위의 교환기 노드에 고장이 발생할 때 하위의 교환기 노드 사이에서의 연결이 크게 영향을 받기 때문에 망의 신뢰성이 떨어지며 상위의 교환기에 트래픽이 집중되어 병목현상이 일어나기 쉽다. 또한, 망의 구성 형태를 일괄적인 트리 형태로 구성하는 것도 어려운 일이다.

HOS와 같은 핸드오버 기법들은 망의 구조에 상관없이 모든 망 구조에서 구현이 가능하고 필요한 대역폭만을 사용하므로 자원의 낭비를 줄일 수 있으며 다양하게 적용할 수 있는 장점이 있지만, 핸드오버시 처리시간이 길어지므로 셀 손실이나 호의 블러킹 확률이 높아지게 된다. 또한, 핸드오버가 발생하는 시점에서 HAC를 처리해야 하므로 이동 단말에서 요구하는 QoS를 만족하지 못할 수도 있다.

이에 따라 본 논문에서는 망의 특정한 형태에 제약 없이 받고 효율적인 경로를 설정해 줄 수 있는 개선된 COS 탐색 알고리즘을 도출하고, 이를 이용하여 이동이 예상되는 지역에만 자원을 미리 할당하여 핸드오버 처리시간을 줄이고 대역폭 낭비를 최소화하는 최소 자원예약 핸드오버 기법을 제안하고자 한다.

2. DH-BT COS 탐색 알고리즘

기존의 Loose Select COS 탐색 기법과 Prior Path COS 탐색 기법은 모두 중앙의 연결 서버(CS)에서 전체 노드의 토폴로지 정보를 얻어 COS 선택에 이용하는 방법이다. 반면에, 본 논문에서는 CS나 LS를 두지 않고 각 노드에서 네트워크의 연결 정보를 수집하여, COS를 선택하는 분산형 처리 방식을 채택하고, threshold R의 개념을 도입하여 Distributed hunt COS 탐색 기법과 Backward tracking COS 탐색 기법중 한가지를 선택하여 사용하도록 하는 DH-BT COS 탐색 기법을 제안한다. 여기서 R 값은 0.0~1.0의 범위를 가진다. 먼저 단말이 BS_{OLD}에서 BS_{NEW}로 이동할 때, BS_{NEW}의 라우팅 테이블을 검색하여 BS_{OLD}까지의 최소 홉 수를 구한다. 여기서 구한 BS_{NEW}에서 BS_{OLD}까지의 최소 홉 수를

N이라 정의한다.

다음으로 기존 경로(BS_{OLD}에서 BS_{DEST}까지의 경로)의 홉 수를 M이라 가정하고 N/M 값을 구하여, 이 값을 threshold R 값과 비교한다. 여기까지의 과정은 BS_{NEW} 측에서 수행한다. 만일 N/M이 R보다 작다면 기존의 Distributed hunt COS 탐색 기법을 적용하고, N/M이 R보다 크다면 Backward tracking COS 탐색 기법을 적용하여 COS를 탐색한다. 또한 Distributed hunt COS 탐색 기법에서 BS_{NEW}에서 기존 경로상의 노드들에 이르는 경로중 최소 홉 수를 가지는 노드가 여러개 존재할 경우 임의로 COS를 선택하였으나 제안한 DH-BT COS 탐색 기법에서는 BS_{OLD}에 가장 가까운 노드를 COS로 선택한다. 이것은 같은 기존 경로상의 노드까지 같은 홉 수를 가지는 노드가 다수일 경우, 이중 BS_{OLD}에 가까운 노드를 선택하는 편이 최선 재사용율을 높일 수 있기 때문이다.

본 논문에서 제안하는 DH-BT COS 탐색 기법은 최소 홉 수 성능을 높이면서 동시에 최선 재사용율도 높이기 위하여 threshold R 값을 선택 기준으로, Distributed hunt 탐색 기법과 Backward tracking 탐색 기법 중 유리한 것을 선택하여 사용한다.

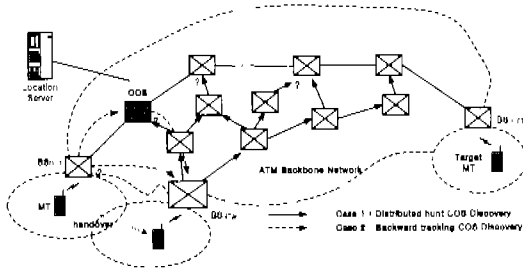
먼저 Distributed hunt COS 탐색 기법의 경우 COS 탐색을 위해 BS_{NEW}로부터 COS 탐색을 위한 멀티캐스팅이 필요하다. 이는 COS가 기존 경로에서 적은 홉 수 내에 존재하면 효율적인 방법이 될 수 있으나 COS가 BS_{NEW}로부터 멀리 존재한다면 멀티캐스팅으로 인한 트래픽 부하가 커지고 처리시간 또한 증가하게 된다.

그러나, DH-BT COS 탐색 기법은 먼저 BS_{NEW}부터 BS_{OLD}까지의 홉 수(N)를 BS_{OLD}에서 BS_{DEST}까지의 홉 수(M)로 나눈 값이 R보다 작으면 Distributed hunt COS 탐색 기법을 적용한다. N/M이 작다는 것은 새로 이동한 BS_{NEW}의 위치가 M을 고려해 볼 때 상대적으로 기존의 경로(BS_{OLD}에서 BS_{DEST})에 가까이 위치함을 의미한다. 따라서, Distributed hunt COS 탐색 기법을 사용하는 것이 효율적임을 알 수 있다.

반대로 N/M이 R보다 크다면 BS_{NEW}의 위치가 기존 경로에서 멀리 떨어져 있음을 의미한다. 이러한 경우 Distributed hunt COS 탐색 기법을 사용하면 브로드캐스팅으로 인한 트래픽 부하와 처리시간이 증가하게 된다. 따라서, BS_{OLD}부터 기존 경로물 따라 COS를 탐색해 나가는 Backward tracking COS 탐색 기법이 유리하다.

DH-BT COS 탐색 기법은 Backward tracking 기법의 높은 회선 재사용율을 유지하면서, COS까지의 최소 홉 수를 보완하기 위해 Distributed hunt 기법을 사용하도록 하여 COS까지의 최소 홉 수를 유지할 수 있도록 보완하고자 하는 것이다. 즉, Distributed hunt 기법과 Backward tracking 기법의 장점은 유지하면서 단점을 보완한 기법이라 할 수 있다.

극단적인 경우 N/M 값이 크면서 새로운 BS_{NEW}가 기존 경로의 BS_{DEST}에 가까이 위치할 수도 있고, 또 그 반대의 경우로 N/M이 작으면서 새로운 BS_{NEW}가 기존 경로의 BS_{DEST}에서 멀리 위치할 수도 있다.



Routing Table Structure for DH-BT COS Discovery

Node i			
Destination Node	Minimum Hop	Next Hop Nodes (including node i)	Connection Supported

그림 1. DH-BT COS 탐색 기법

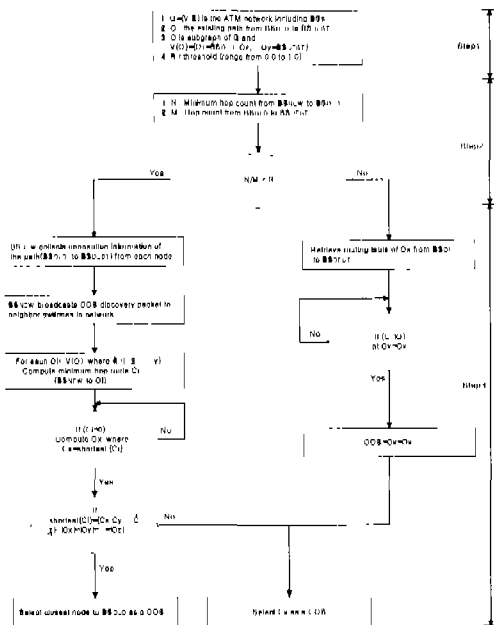


그림 2. DH-BT COS 탐색 알고리즘의 흐름도

그러나 이러한 경우는 특수한 지형 등의 경우에 존재할 수 있겠으나, 일반적인 형태는 아니므로 본 논문에서는 고려하지 않는다.

본 논문에서 제안하는 DH-BT COS 탐색 기법의 개념과 흐름도는 그림 1, 그림 2에 나타나 있으며 알고리즘은 다음과 같다.

Step 1 : Define notations for DH-BT COS discovery

1. Let $G=(V,E)$ represent the backbone ATM network (including BSs)
2. O : the existing route from BS_{OLD} to BS_{DEST}
3. O is subgraph of G and $V(O)=\{O_1=BS_{OLD}, O_2, \dots, O_y=BS_{DEST}\}$
4. R : threshold (range from 0.0 to 1.0)

Step 2 : Select COS discovery algorithm

1. N : Minimum hop count from BS_{NEW} to BS_{OLD}
2. M : Hop count from BS_{OLD} to BS_{DEST}(existing path)
3. Compare N/M and R
4. If $N/M < R$ goto case 1
else goto case 2

Step 3 : COS discovery

Case 1. Distributed hunt COS discovery

Begin

Having discovered BS_{OLD} to BS_{DEST} route nodes information

For each $O_i \in V(O)$, where $\{i=1, 2, \dots, y\}$,

Compute the minimum hop routes from BS_{NEW} to O_i

Let these route be C_i

If $(C_i \neq 0)$

Begin

Compute O_x , where $C_x = \text{shortest } \{C_i\}$

Hence $COS = C_x$

or

/* Case when multiple minimum hop BS_{NEW} to COS route exist */

Begin

If $\text{shortest}\{C_i\} = \{C_x, C_y, \dots, C_z\}$,

i.e. $|C_x| = |C_y| = \dots = |C_z|$ (if BS_{NEW} have multiple route to each node which have same hop count)

Begin

Select closest node to BS_{OLD} as a COS

End

End

End

Else

Begin

G is partitioned

i.e. COS unreachable from BS_{NEW}
 End
 End

Case 2. Backward tracking COS discovery

```

Begin
I→0
Again :
Starting from Oi+1,
Compute the minimum hop route from Oi to BSNEW
Let this route be denoted by C,
where V(C)={C1=Oi, C2, C3, ..., Cy=BSNEW}
If (C≠0),
    Begin
        If (C∩O) at Cx=Ox=BSOLD,
            Then COS=BSOLD, goto done
        or
/* Case when COS Convergence does not occur at the
BSOLD */
        Begin
            If (C∩O) at (C1=O1, C2=O2, ...)
                Then COS=Ok, where Ok is furthest from
BSOLD on O, goto done
            or
                Begin
                    If (C∩O) at C1=O1,
                        Then COS=O1, goto done
                End
            End
        End
    else
        Begin
            If (Oi+BSDEST) goto Again /* i.e., Backtrack */
        else
            Begin
                G is partitioned
                i.e., BSNEW is unreachable from Oi
            End
        End
    done:
End
    
```

3. 최소 자원예약 핸드오버 기법의 제안

본 절에서는 현재의 이동 단말이 핸드오버 지역(handover area)내의 셀에 위치할 때, 인접 클러스터 내 2~3개의 이웃한 셀들에 대해 핸드오버가 발생하기 전, 미리 경로를 설정하여 예약하는 최소 자원예약 핸드오버 기법을 제안한다. COS 탐색은 이미 앞에서 언급한 바와 같이 핸드오버 처리과정의

일부분이므로 최소 자원예약 핸드오버 기법에서는 COS 탐색 기법으로 앞서 제안한 DH-BT COS 탐색 기법을 사용한다. DH-BT COS 탐색 기법은 COS 탐색을 필요로 하는 핸드오버 기법이라면 모두 적용이 가능하며, 본 절에서 제안하는 최소 자원예약 핸드오버 기법도 이를 사용할 수 있다.

최소 자원예약 핸드오버 기법에서는 클러스터 경계에 위치하는 마이크로/피코 셀을 핸드오버 지역으로 지정하고 이 셀의 BS는 인접한 모든 클러스터의 BS들에 대한 정보를 가지고 있다고 가정한다. 그리고 이동 단말이 클러스터 간의 핸드오버 지역에 진입할 경우 인접 클러스터의 이웃한 셀에 미리 핸드오버 절차를 수행하도록 하여 단말이 클러스터 경계를 넘어갈 경우 바로 핸드오버를 마칠 수 있도록 한다.

그림 3은 제안한 최소 자원예약 핸드오버 기법의 개념을 나타낸 것이다. 그림 3과 같이 무선 ATM 망은 마이크로/피코 셀과 이들의 집합인 클러스터로 구성된다. BS는 이동단말과의 무선연결에 관여하며 클러스터 내의 핸드오버는 ATM 스위치의 무선 접속 포트번호만 변경하면 되므로 큰 오버헤드가 되지 않는다.

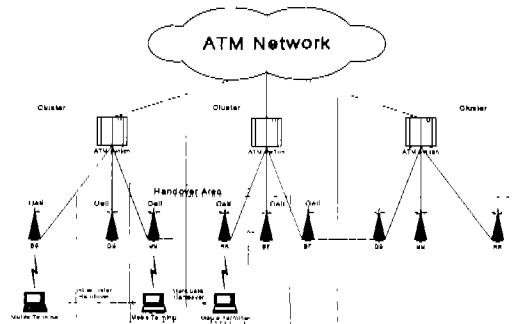


그림 3. 최소 자원예약 핸드오버 기법

그러나, 클러스터간 핸드오버는 경로의 재라우팅과 핸드오버 수락제어 등이 필요하므로 핸드오버의 처리시간이 문제가 되며, 연결이 이루어지지 않을 경우 호의 블러킹이 발생할 수도 있다.

따라서, 이와 같은 문제점을 해결하기 위해 최소 자원예약 기법에서는 클러스터 경계에 있는 BS들이 관장하는 셀들을 핸드오버 지역으로 지정하고, 이 BS들은 인접 클러스터의 BS들에 대한 정보를 가지며 이동 단말이 핸드오버 지역에 진입할 경우 인접 클러스터의 BS들에 대한 핸드오버 절차를 수행하고 경로를 배정받게 된다.

기존에도 VCT 기법과 같이 이동 가능한 모든 BS에 경로를 배정하여 핸드오버 처리를 빠르게 수행할 수 있는 방법은 있으나, 이는 이동 단말이 설정된 모든 연결중 하나만을 사용하므로 자원의 낭비가 심하고 충분한 대역폭이 있다는 가정하에서만 사용이 가능하다.

반면에 HOS 기법과 같이 단지 필요한 대역폭만 핸드오버시 할당하여 사용할 경우, 자원의 낭비는 없지만 핸드오버시 처리시간이 길어짐에 따라 셀 손실과 호의 블리킹 분계가 대두된다.

이에 비해 최소 자원예약 핸드오버 기법은 2개 또는 3개 경로만이 필요하므로 오버헤드가 상당히 줄어들며, 실제 핸드오버가 일어날 때는 이미 호의 핸드오버 수락세어가 끝나게 되어 있으므로 호의 클러킹 확률을 감소시킬 수 있다. 또한 핸드오버 후에도 이전 셀의 경로를 해제하지 않고 다시 이동 단말이 핸드오버 할 경우를 대비하여 예약상태로 남겨둠으로써, 장시간 지그재그로 이동하는 사용자들을 위해서도 핸드오버 처리의 부하를 줄일 수 있다.

만일 기지국들을 서로 물리적으로 연결한다면 이웃한 셀에 대한 정보를 쉽게 알 수 있으며 핸드오버 처리중 버퍼링된 셀을 바로 이동할 새로운 기지국으로 전송하여 처리할 수 있으므로 가장 이상적인 방법일 수도 있다. 그러나, 이러한 방법은 모든 이웃한 셀들간에 물리적 경로가 존재하여야 하므로 망의 구조가 복잡해지고 상당한 비용의 증가를 피할 수 없게 된다. 이에 따라 최소 자원예약 핸드오버 기법에서는 인접 셀들과의 물리적인 경로는 없으며 다만 인접 셀에 대한 정보만을 가지고 있다고 가정한다.

핸드오버 지역의 셀들은 다음과 같은 방법으로 구분할 수 있다. 하나의 클러스터는 기지국들의 집합과 이들 기지국들과 연결되어 고정방 접속을 제공하는 하나의 ATM 교환기로 구성된다. 하나의 클러스터 C는 각 셀 c_i 와 관련된 N개의 셀들과 기지국들로 구성되어 있다고 할 때 다음과 같이 나타낼 수 있다.

$$C = \{c_1, c_2, \dots, c_N\}$$

또한, 그 클러스터가 M개의 클러스터와 인접해 있다고 한다. 우리는 D_i 를 인접 클러스터 i의 근처에 있는 셀들의 집합으로 정의한다.

$$D_i = \{d_1, d_2, \dots, d_k\} \subset C, \quad d_i \subset C$$

어떠한 D_i 에도 속해 있지 않은 셀들의 집합을 R이라고 하면 다음과 같다.

$$R = \{r_1, r_2, \dots, r_l\} \subset C, \quad r_l \subset C, \\ R \cap D_i = \emptyset \text{ for all } i$$

예를 들어, 그림 4와 같이 20개의 셀을 갖는 하나의 클러스터를 고려해 보면 이 클러스터는 4개의 인접 클러스터를 가진다. 그림 3으로부터 다음과 같이 나타낼 수 있다.

$$C = \{c_1, c_2, c_3, \dots, c_{19}, c_{20}\}$$

$$D_1 = \{c_9, c_{18}, c_{19}, c_{20}\}$$

$$D_2 = \{c_9, c_{10}, c_{11}, c_{12}\}$$

$$D_3 = \{c_{12}, c_{13}, c_{14}, c_{15}\}$$

$$D_4 = \{c_6, c_{16}, c_{17}, c_{18}\}$$

$$R = \{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$$

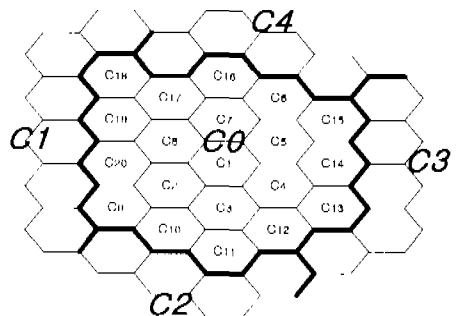


그림 4. 4개의 인접 클러스터를 가지는 클러스터의 예

D의 각 집합은 다른 클러스터와 인접하며 셀 하나당 인접 클러스터의 2~3개의 셀과 인접하게 된다.

위와 같은 방법으로 핸드오버 지역의 셀과 그렇지 않은 셀들을 구분할 수 있으며 이는 각 클러스터 내의 교환기에서 정보를 관리하도록 한다. 또한, D의 각 집합은 자신과 인접한 클러스터의 셀에 대한 정보도 가지고 있어야 한다.

최소 자원예약 핸드오버 기법의 수행절차는 다음의 그림 5와 같으며 시그널링 흐름은 그림 6과 같다.

최소 자원예약 핸드오버 기법은 핸드오버 준비를 위한 step 1과 실제 핸드오버를 수행하는 step 2로 구분되며 알고리즘은 다음과 같다.

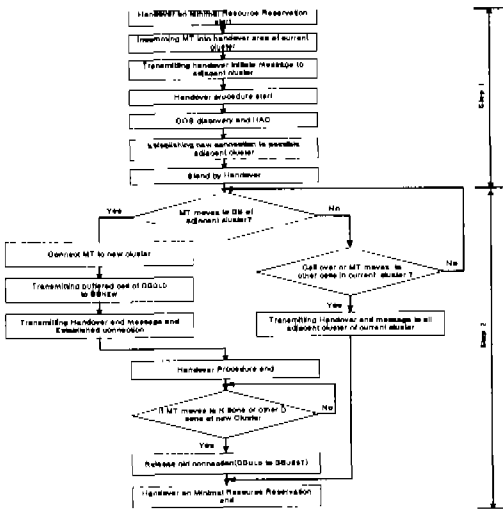
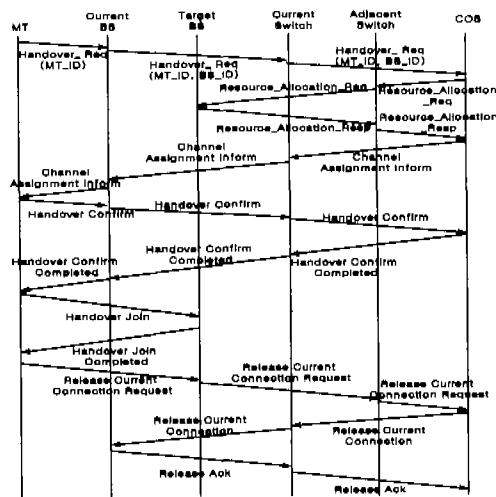


그림 5. 최소 자원예약 핸드오버 기법의 수행 흐름도



Handover_Req : 핸드오버 요청, 새로운 경로 설정 요청
 Resource_Allocation_Req : 자원 할당 요청
 Resource_Allocation_Resp : 자원 할당 응답
 Channel_Assignment_Inform : 채널 할당 정보
 Handover_Confirm : COS에 새로운 경로를 활성화 시켜줄 것을 요청
 Handover_Confirm_Completed : Handover_Confirm 응답
 Handover_Join : MT와 Target BS의 무선 연결 설정에 의한 BS2를 통한 연결 완료
 Handover_Join_Completed : 무선 경로 연결
 Release_Current_Connection_Request : 기존 경로 해제 요청
 Release_Current_Connection : 기존 경로 해제
 Release_Ack : 경로 해제 응답

그림 6. 최소 자원예약 핸드오버 기법의 시그널링 흐름도

Step 1 : Handover reservation

Begin

Incomming MT into handover area(D zone) of current cluster

Current BS Transmits handover initiate message to adjacent cluster

Handover procedure start

Begin

COS discovery and HAC

Establishing new connections to possible adjacent clusters (Maximum 3)

Standby Handover

Step 2 : Handover processing

Begin

If MT moves to cell of adjacent cluster then **Begin**

connect MT to BS_{NEW}

Transmitting buffered cells

of BS_{OLD} to BS_{NEW} via COS

Transmitting handover end

message to all adjacent cluster of new cluster

If MT Moves to R zone or

other D zone of new cluster

then release old

connection(BS_{OLD} to BS_{DEST})

endif

end

else

If Call over or MT moves to other cells in current cluster(R zone)

then Transmitting handover end message to all adjacent cluster of new cluster

endif

end

end

end

III. 시뮬레이션 및 결과 고찰

본 장에서는 제 II 장의 DH-BT COS 알고리즘과 제안한 최소 자원예약 핸드오버 기법에 대한 시뮬레이션을 수행하고 이를 분석하였다. 이를 위해 먼저 기존의 COS 탐색 알고리즘과 DH-BT COS 탐색 알고리즘을 시뮬레이션을 통하여 최선 재사용율과 최소 홉 수 비교 분석을 하여 성능이 개선되었음을 확인하고, 최소 자원예약 핸드오버 기법은 대역폭 점유율, 핸드오버 처리시간 등을 비교 분석하였다.

1. 시뮬레이션 모델링

COS 탐색 알고리즘의 시뮬레이션을 위해서 Random 망, Star 망, Tree 망을 가정하여, 이를 기존의

Loose select, Prior path knowledge, Distributed hunt, Backward tracking 탐색 기법과 비교하였다. 실험을 위해 구성된 망에서는 이동 단말은 스위치와 연결된 BS를 랜덤하게 이동한다. 여기에 사용된 망의 노드 수는 10, 20, 30, 40, 50개를 가정하였다.

먼저 Random 망의 경우는 완전한 Mesh 망이 아닌 부분적인 Mesh 망으로 구성하였으며 모든 노드로 가는 경로가 존재하도록 하였다. 또한, 연결의 수는 노드 수의 두배로 하였다.

Star 망은 노드 수와 동일한 연결의 수를 입력하여 구성하였다. 실험에서 구성한 star 망은 5개의 노드를 중심으로 10, 20, 30, 40, 50 노드에 따라 각각 1, 3, 5, 7, 9개의 노드를 균등하게 배분하여 망을 구성하였다.

Tree 망은 연결의 수는 노드 수보다 연결의 수가 하나 작게 하여 구성하였으며, 시뮬레이터는 IBM PC 상에서 Visual C++ 언어를 사용하였다.

COS 탐색 알고리즘은 각 망 모델에 대하여 1000 번씩 시뮬레이션을 수행하여 회선 재사용율과 COS까지의 최소 홉 수를 분석하였다. 회선 재사용율은 핸드오버 후의 경로, 즉, BS_{NEW}에서 BS_{DEST}까지의 경로와 재라우팅에 따른 핸드오버 이전 경로중에서 재사용하는 부분의 비율을 비교하기 위한 것으로 다음과 같이 설정하였다.

$$\text{회선 재사용율} = \left(\frac{\text{COS에서 BS}_{DEST}\text{까지의 홉수}}{\text{BS}_{NEW}\text{에서 BS}_{DEST}\text{까지의 홉수}} \right)$$

COS까지의 최소 홉 수는 각각의 망 모델에 대하여 50개의 노드를 가지는 경우를 시뮬레이션을 수행하여 발생빈도를 구하고 성능을 비교하였다.

회선 재사용율이 높고 COS까지의 홉 수가 작다는 것은 셀 손실이나 핸드오버 처리 지연이 작아짐으로서 원활한 핸드오버가 가능하다는 의미로 지니게 되므로 COS 탐색 기법의 성능 분석에서 매우 중요한 요소이다.

최소 자원예약 핸드오버 기법의 성능분석을 위해서는 핸드오버 처리시간과 대역폭 점유율을 시뮬레이션하여 이를 기존의 HOS 기법과 VCT 기법을 함께 비교하였다. 각 핸드오버 기법에 대한 시간 복잡도는 다음과 같이 구할 수 있다.

HOS 기법의 경우 한 점에서 다른 한 점으로의 경로의 수를 e 라 하면 최단 거리 경로를 위하여 $O(e)$ 의 시간이 필요하다.

VCT 기법은 VCT가 생성되어 있는 상태에서 이

동 단말이 이동하는 경우와, VCT의 범위를 벗어나 새로 VCT를 생성해야 하는 경우를 생각해 볼 수 있으며 다음과 같은 가정이 필요하다.

VCT가 이진 트리로 구성되어 있다고 가정하면 $O(I)$ 의 시간이 필요하며, VCT를 생성할때는 한 노드를 이진트리에 삽입하는 시간 $O(\log_2 n)$ 에 의해 $O(\sum_{i=1}^I \log_2 n) = O(\log_2 n!)$ 의 시간이 필요하다.

제안한 최소 자원예약 핸드오버 기법은 최대 3개의 연결이 필요하므로 $O(3)$ 의 처리시간이 필요하다. 공간 복잡도의 경우 HOS는 자신이 필요한 경로 하나만을 사용하므로 $O(I)$ 로 나타낼 수 있고, VCT는 논리적 트리를 구성하는 노드를 모두 포함하므로 $O(n)$, 제안한 최소 자원예약 핸드오버 기법은 최대 3개의 연결과 기존의 연결을 포함하여 $O(4)$ 의 값을 가진다.

시뮬레이션에 사용한 모델은 $I=50, J=10$ 을 가정하였으며, 5개의 VCT로 구성될 수 있다. 먼저 I 는 ATM 스위치의 총 개수로 가정하며, J 는 하나의 VCT 범위내에 존재하며 COS로 동작할 수 있는 ATM 스위치의 총 개수라 가정하며, I 는 J 에 비해 훨씬 크다고 할 수 있다. 또한, 이동 단말은 콜러스터간을 랜덤하게 이동한다고 가정하였다.

대역폭 점유율은 망 전체의 노드 수를 변화시켜 가면서 시뮬레이션을 수행하였다.

2. 결과 고찰

본 절에서는 DH-BT COS 알고리즘과 최소 자원예약 핸드오버 기법의 시뮬레이션을 수행한 결과를 설명한다.

먼저 그림 7~그림 9는 DH-BT COS 탐색 알고리즘을 기존의 COS 탐색 알고리즘과 회선 재사용율면에서 비교하여 본 것이다. 여기서 threshold R 값은 여러 가지 가정이 있을 수 있으나 본 논문에서는 임의로 0.2, 0.5, 0.8을 R 값으로 사용하였다.

그림 7에서는 Random 망에 대한 회선 재사용율을 비교한 것으로 DH-BT COS 탐색 알고리즘이 기존의 네 가지 COS 탐색 알고리즘 중 Backward tracking을 제외한 세가지 기법에 비해 회선 재사용율이 훨씬 높게 나타남을 볼 수 있다. 특히 R 값이 0.2일 때 좋은 성능을 보였는데, 이는 DH-BT COS 탐색 기법에 적용시킨 Backward tracking 기법이 다른 기법들과 달리 BS_{OLD}에서부터 COS를 탐색하여 회선 재사용율을 높일 수 있기 때문이다. 그러나 Backward tracking의 경우 최적의 COS를 탐색하기

는 어렵기 때문에 다른 세 가지 기법들에 비해 최소 홉 수의 COS 선택 성능은 약간 낮다.

다음의 그림 8, 그림 9는 Star 망과 Tree 망에 대하여 최선 재사용율을 비교한 것이다. Random 망과 마찬가지로 본 논문의 시뮬레이션에 사용한 망 모델에 대해서는 Star 망과 Tree 망 모두 R=0.2에서 최선 재사용율이 가장 높게 나타났다.

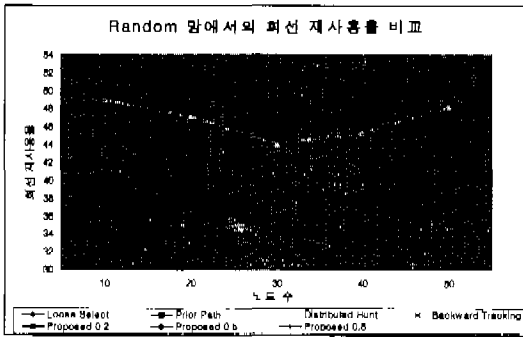


그림 7. Random 망에서 COS 탐색 알고리즘의 최선 재사용율 비교

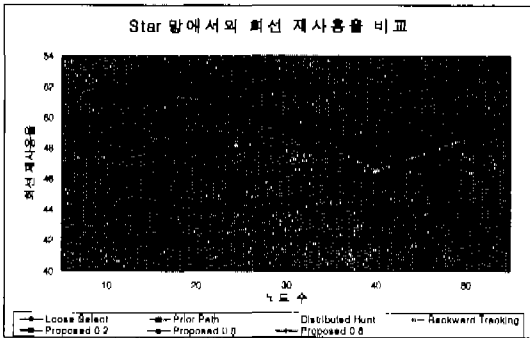


그림 8. Star 망에서 COS 탐색 알고리즘의 최선 재사용율 비교

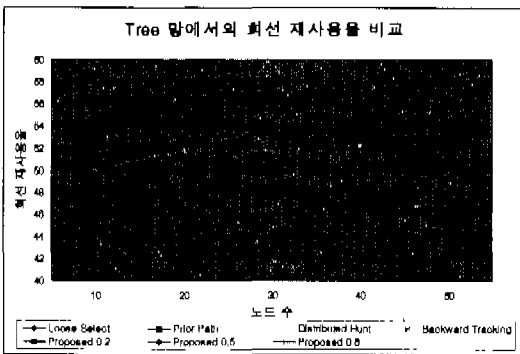


그림 9. Tree 망에서 COS 탐색 알고리즘의 최선 재사용율 비교

다음의 그림 10~그림 12는 Random, Star, Tree 망에서 각각 50노드씩 가지는 것으로 가정하고, 1000번씩 시뮬레이션을 수행하여 BS_{NEW}부터 COS까지 홉 수 빈도를 구한 것이다. 여기서 홉 수가 0인 것은 BS_{NEW}와 연결된 중단 스위치가 COS로 선택된 경우로, 이 때는 BS와 스위치간의 홉을 계산하지 않았다.

먼저 Random, Star, Tree 망에서 기존의 네가지 기법을 최소홉 수의 측면에서 비교해보면, Backward tracking을 제외한 나머지 세가지 기법은 거의 비슷한 결과값 보이거나 Backward tracking은 핸드오버 시 새로 설정된 BS_{NEW}부터 COS까지 경로의 홉 수가 더 많은 것을 알 수 있다. 이는 Backward tracking이 BS_{OLD}부터 탐색을 시작하여 기존 경로에서 BS_{NEW}까지의 경로가 가장 먼저 나타나는 스위치를 COS로 선택하기 때문에 최선 재사용율은 높지만 최적경로가 아닐 수 있기 때문이다.

반면 제안한 DH-BT COS 탐색 기법은 Distributed hunt COS 탐색 기법과 Backward tracking COS 탐색 기법을 N/M에 따라 선택적으로 사용하도록 하였기 때문에 Backward tracking COS 탐색 기법의 단점을 보완 할 수 있다. 즉, 최선 재사용율은 높이면서 COS까지의 최소 홉 수도 만족할 만한 성능을 보인다.

COS까지의 최소 홉 수는 앞에서도 언급한 바와 같이 COS의 거리가 멀리 떨어져 있으면 버퍼링된 셀 전송시 셀 손실율이 높아지며 처리시간이 지연되는 문제점이 있기 때문에 매우 중요한 요소이다.

시뮬레이션 결과, COS까지의 최소 홉 수는 기존의 기법중 Backward tracking을 제외한 나머지 기법들과 제안한 DH-BT 기법이 거의 비슷한 결과값 보였다. DH-BT 기법은 특히 R=0.2에서 가장 좋은 결과값 보였으며 0.8에서는 Backward tracking 보

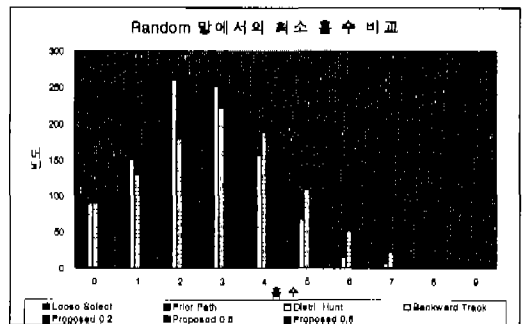


그림 10. Random 망에서 최소 홉 수와 빈도 비교

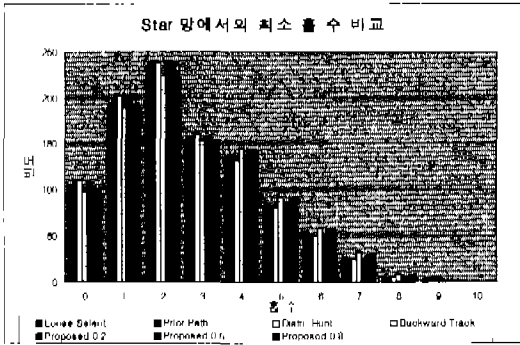


그림 11. Star 망에서 최소 홉 수와 빈도 비교

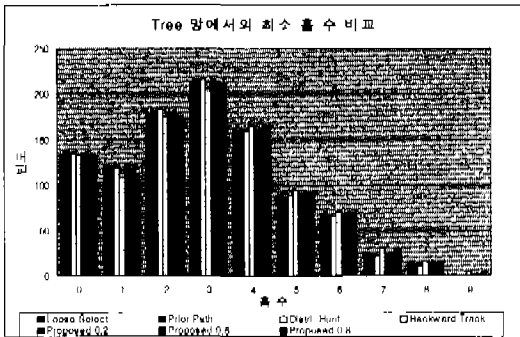


그림 12. Tree 망에서 최소 홉 수와 빈도 비교

다는 나은 결과를 보였지만 나머지 기존의 세가지 기법에 비해서는 성능이 낮았다.

시뮬레이션 결과 본 논문에서 제시한 DH-BT COS 탐색 기법의 최선 재사용율은 Backward tracking 정도로 높은 성능을 보이며, Backward tracking의 단점인 최소 홉 수를 가지는 COS 선택은 Distributed hunt 기법을 선택적으로 사용하게 하여 성능이 개선되었음을 알 수 있다. 즉, Backward tracking의 장점인 최선 재사용율은 그대로 유지하면서 COS까지의 최소 홉 수 성능을 Distributed hunt 기법에 의해 보완한 것이다.

이 기법은 구원시 N/M값이 큰 경우 Backward tracking을 사용함으로써 방의 부하를 감소시키고 최선 재사용율을 높일 수 있으며, N/M이 작은 경우 Distributed hunt 기법을 사용함으로써 방에 큰 부하 없이 COS 탐색을 신속히 처리하고 최소 홉 수로 COS를 선택할 수 있도록 한다.

본 논문에서 가정한 세가지 망 모델에 대해서는 R=0.2에서 모두 가장 좋은 결과를 보였다. 실제 망에 적용시 망 구성 단계에서 최적의 R 값을 구하여 COS 탐색시 사용할 수 있는 것이며, 이는 망 구조 변경시 다시 계산되어 사용할 수 있다.

다음으로 제안한 최소 자원예약 핸드오버 기법의 시뮬레이션을 수행하였다. 먼저 최소 자원예약 핸드오버 기법과 기존의 HOS, VCT 기법의 처리시간을 I=50, J=10인 경우에 대하여 비교하였다. 그림 13에 나타난 바와 같이 핸드오버 처리시간은 VCT 보다는 크고 HOS 보다는 작다.

VCT와 제안한 최소 자원예약 핸드오버 기법 모두 핸드오버가 가능한 BS에 미리 연결을 설정하므로 HOS 기법에 비해 적은 시간이 필요하다. 그러나 VCT에서는 이동단말의 핸드오버시 VCT를 생성해야 할 경우 처리시간이 많이 걸린다.

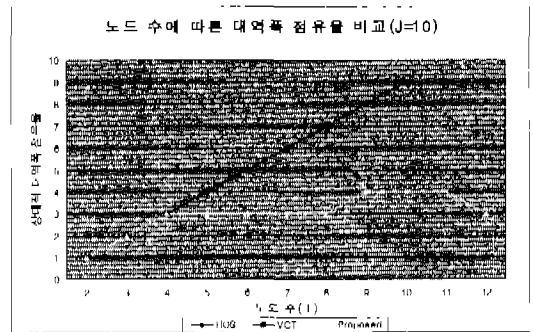


그림 13. 핸드오버 처리시간 (I=50, J=10)

HOS와 최소 자원예약 핸드오버 기법은 단말의 이동횟수에 영향을 받지 않고 처리시간이 거의 일정하다. 그러나 VCT의 경우 10개의 노드를 가지는 트리를 가지고 있으므로 생성된 트리를 벗어나는 경우 다시 VCT를 생성해야 한다. 따라서 이러한 경우는 최소 자원예약 핸드오버 기법에 비해 많은 시간을 소모한다.

다음의 그림 14는 J는 10이하로 제한하고 I값을 변화시켰을 때 즉, 전체 노드 수의 변화에 따른 핸드오버 처리시간을 나타낸 것이다. HOS의 경우는 노드 수 보다 경로 수에 직접적인 영향을 받게 되므로 여기서는 경로의 수를 노드 수의 두 배수로 가정한다. 단, 노드 수가 5개 이하인 경우는 경로의 수가 노드 수의 두 배수에 이르지 못하므로 완전 매쉬망으로 가정한다.

그림에서 보는 바와 같이 VCT가 생성되어 있는 경우에는 VCT가 가장 좋은 결과를 보인다. 그러나 앞에서 언급한 바와 같이 VCT를 생성해야 하는 경우에는 VCT 노드 수가 증가할수록 제안 핸드오버 기법에 비해 많은 처리시간을 요구한다. 즉, 트리를 구성하는 노드의 수가 증가함에 따라 VCT의 성능은 저하될 수밖에 없다. 그러나 이는 트리를 구성하

는 노드 수인 10을 초과하면 전체 망의 노드 수와 관계없이 일정한 값을 보인다.

HOS의 경우에도 노드수의 증가에 따라 탐색해야 할 노드가 많아지므로 처리시간은 증가하게 되며 핸드오버 기법 중 가장 처리시간이 길게 나타났다.

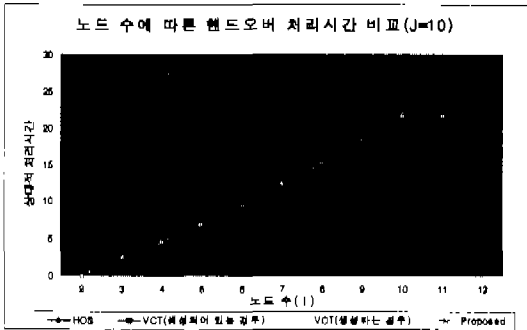


그림 14. 노드수의 변화에 따른 핸드오버 처리시간

대역폭 점유율은 HOS를 기준으로 하여 HOS, VCT와 제안 핸드오버 기법을 J는 10이하로 제한하고 I값을 변화시키면서 비교하였다. 그림 15에서 보는 바와 같이 대역폭 점유율은 VCT에 비해 최소 자원예약 핸드오버 기법이 훨씬 작다. 이는 VCT는 가상 트리의 루트 이하의 모든 스위치들에 연결을 설정해 주어야 하기 때문에 실제 필요한 것보다 훨씬 많은 대역폭을 필요로 한다.

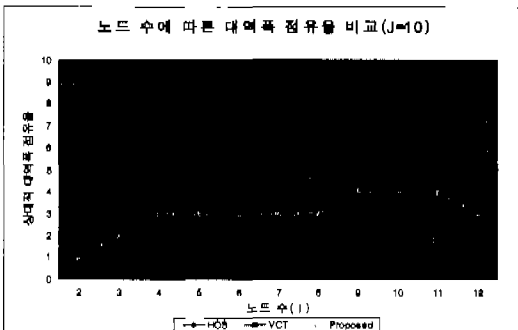


그림 15. 핸드오버시 대역폭 점유율 비교

그러나 최소 자원예약 핸드오버 기법에서는 2개 또는 3개의 BS만을 미리 예약시켜 놓기 때문에 VCT에 비해 대역폭 사용이 매우 작다. 그러나 HOS에 비해서는 3~4배의 대역폭을 필요로 한다. 이는 VCT의 경우 트리 내의 모든 경로에 연결을 설정해야 하지만, HOS는 필요시 이전의 경로와 새로운 클러스터의 경로가 각각 하나씩만 필요하며,

제안 핸드오버 기법에서는 미리 새로운 경로를 2~3개 준비해야 하기 때문이다. 그러나 HOS 기법의 경우 대역폭은 실제 이동단말이 필요한 만큼만 사용하므로 가장 효율적으로 보일 수도 있으나 핸드오버시 호의 블러킹이 발생할 확률이 높아지는 등 서비스 품질의 저하가 있을 수 밖에 없다.

IV. 결론

제안한 DH-BT COS 탐색 기법은 Backward tracking 기법의 높은 회전 재사용율을 유지하면서, COS까지의 최소 홉 수를 보완하기 위해 Distributed hunt 알고리즘을 선택적으로 사용하도록 하여 COS까지의 최소 홉 수를 유지할 수 있도록 보완하고자 하는 것이다. 이는 이동단말과 목적지까지 홉 수에 threshold 값 R을 주어 이 값에 따라 두 가지 알고리즘 중 처리시간이 빠르고 망에 부하를 감소시킬 수 있는 기법을 선택하여 사용한다.

최소 자원예약 핸드오버 기법은 클러스터 경계에 위치하는 BS들이 인접 클러스터의 BS들에 대한 정보를 가지고 있으며, 이 지역을 핸드오버 지역으로 지정하여 이동 단말이 핸드오버 지역으로 진입하면 핸드오버 절차를 미리 수행하도록 하였다. 또한 BS들이 인접 셀에 대한 정보를 가지고 있으므로 이동이 가능한 인접 클러스터에 빠른 핸드오버 처리 요청이 가능하고 핸드오버 처리시 셀 버퍼링에 따른 트래픽을 감소시킬 수 있다.

시뮬레이션 결과는 DH-BT COS 탐색 알고리즘의 경우 부분 경로 재설정시 중요한 파라미터인 회전 재사용율이 기존 COS 탐색 알고리즘에 비해 높게 나타났다. 이는 최소한의 경로배정으로 핸드오버를 가능하게 할 것이며, 경로의 최소 홉 수에서도 Backward tracking 기법의 단점을 보완하여 DH-BT COS 탐색 알고리즘이 개선된 성능을 보이는 것으로 나타났다.

본 논문에서 가정한 Random, Star, Tree 망에 대해서는 R=0.2일 때 가장 좋은 결과를 보였다. 그러나 이는 망 구조에 따라 가변적인 것으로 실제 구현시에는 망 구성단계에서 최적의 R 값을 구하여 사용할 수 있으며, 또한 망 구조의 변경시에는 다시 최적의 R 값을 구하여 사용할 수 있을 것이다.

최소 자원예약 핸드오버 기법은 비슷한 핸드오버 처리시간을 보이는 VCT 기법의 가장 큰 단점인 대역폭 낭비를 감소시키면서 이동이 가능한 인접 클

러스터에 대한 핸드오버 절차를 미리 수행하여 연결의 신뢰도를 높이고 핸드오버 처리시간을 감소시킬 수 있음을 보였다. 또한 HOS 기법에 비해서는 처리 시간이 대폭 감소된 결과를 나타내었다.

본 논문에서 제안한 DH-BT COS 탐색 알고리즘과 최소 자원예약 핸드오버 기법은 망 토폴로지에 관계없이 실제 망 구성과 운영기법으로 사용될 수 있을 것이다.

참 고 문 헌

[1] R.Yuan, S.K.Biswas, and D.Raychaudhuri, "A Signaling and Control Architecture for Mobility Support in Wireless ATM Networks," Proc. of IEEE ICC '96. Dalas, Texas, pp.478-484, Jun. 1996

[2] S.Acampora and M.Naghshineh, "An Architecture and Methology for Mobile-Executed Handover in Cellular ATM Networks," IEEE JSAC, Vol. 12, pp.1365-1375, Oct. 1994

[3] M.Veeraraghavan, M.J.Karol, and K.Y.Eng, "Mobility and Connection Management in a Wireless ATM LAN," IEEE JSAC, Vol. 15, No. 1, pp.50-68, Jan. 1997

[4] Bora A.Akyol and D.C.Cox, "Rerouting for Handover in a Wireless ATM Network," IEEE Personal Communications Magazine, pp.26-33, Oct. 1996

[5] C.K.Toh, "Wireless ATM and Ad-Hoc Networks," Kluwer Academic Publishers, 1997

[6] C.K.Toh, "The Design and Implementation of a Hybrid Handover Protocol for Multimedia Wireless LANs," Mobicom '95, pp. 13-15, Nov. 1995

[7] D.C.Cox, "Rerouting for Handoff in a Wireless ATM Network," 5th IEEE ICUPC, vol. 1, pp.374-379, 1996

[8] Bora A.Akyol and D.C.Cox, "Handling Mobility in a Wireless ATM Network," INFO-COM '96, pp.24-28, Mar. 1996

최 승 권(Seung-Kwon Choi) 정회원
한국통신학회논문지 제23권 9호 참조
현재: 충북대학교 컴퓨터공학과 박사수료

안 상 규(Sang-Kyu An) 정회원
한국통신학회논문지 제24권 3호 참조
현재: 한국 에너지 기술연구소 정보전산팀
선임기술원

이 병 수(Byeong-Su Lee) 정회원
한국통신학회논문지 제22권 4호 참조
현재: (주) 중위메디칼 상무

조 응 환(Yong-Hwan Cho) 정회원
한국통신학회논문지 제23권 9호 참조
현재: 충북대학교 컴퓨터공학과 교수
<주관심 분야> 멀티미디어 통신, 정보통신
정책(IMT-2000, 번호 정책, 번호 이동성,
요금 정책)
e-mail : yhcho@cbucc.chungbuk.ac.kr