

PDSN 영역내의 여러 RN간 멀티캐스트 그룹 메커니즘 기반의 Seamless 핸드오프 알고리즘

신 동 진[†] · 김 수 창^{††} · 임 선 배^{†††} · 오 재 천^{††††} · 송 병 권^{†††††} · 정 태 의^{††††††}

요 약

3GPP2가 제안하는 방식에서는 Macro Mobility 지원을 위하여 MIP를 이용하며 PDSN은 FA의 기능을 수행한다. 이때 하나의 PDSN에서 다른 PDSN으로 MS가 이동할 경우 지원되는 이동성을 Macro Mobility라 하며, PDSN 관리 영역 내의 하나의 RN에서 다른 RN으로 이동 시에 지원되는 이동성을 Micro Mobility라 한다. 3GPP2가 제안하는 방식에서는 PDSN이 FA의 역할을 수행하므로 이동성 지원이 가능하지만, 이는 Macro Mobility를 지원하기 위한 메커니즘으로, Micro Mobility를 지원하기엔 빠르고 Seamless한 핸드오프에 취약하다. 본 논문은 Micro Mobility를 지원하기 위한 멀티캐스트 그룹 메커니즘 기반의 Seamless 핸드오프 알고리즘을 제안하고 있다. 제안된 알고리즘은 MS의 이동방향과 속도를 계산하여, 예상 이동경로에 인접한 RN들을 멀티캐스트 그룹으로 구성하고, 그룹 join 시점을 최대한 늦춤으로써 망의 효율성을 높인다. 또한, 기존의 멀티캐스트 연결 방법이 가지고 있는 버퍼 오버헤드에 대한 문제점을 해결하기 위해, RN은 예상 핸드오프 시간 이후의 데이터만을 버퍼링 한다. 제안된 알고리즘의 Deadlock Freeness 및 Liveness를 검증하기 위해 State Transition Diagram을 작성하고, 페트리네트 모델을 이용 도달성 트리를 작성하였다. 또한, 시뮬레이션을 통한 성능분석을 수행하였다.

The Seamless Handoff Algorithm based on Multicast Group Mechanism among RNs in a PDSN Area

Dong-Jin Shin[†] · Su-Chang Kim^{††} · Sun-Bae Lim^{†††} · Jae-Chun Oh^{††††}
Byung-Kwon Song^{†††††} · Tae-Eui Jeong^{††††††}

ABSTRACT

In 3GPP2 standard, MIP is used and a PDSN performs the function of FA to support macro mobility. When a MS is roaming from a PDSN area to another, the mobility supported is called macro mobility, while it is called micro mobility when a MS is roaming from a RN area to another in a PDSN area. Since a PDSN performs the function of FA in 3GPP2 standard, it is possible to support mobility but its mechanism is actually for supporting macro mobility, not for micro mobility, thus it is weak in processing fast and seamless handoff to support micro mobility. In this paper, we suggest the seamless handoff algorithm based on multicast group mechanism to support micro mobility. Depending on the moving direction and velocity of a MS, the suggested algorithm constructs a multicast group of RNs on the forecasted MS's moving path, and maximally delays RNs' joining to a multicast group to increase the network efficiency. Moreover, to resolve the buffer overhead problem of the existent multicast scheme, the algorithm suggests that each RN buffers data only after the forecasted handoff time. To prove deadlock freeness and liveness of the algorithm, we use state transition diagrams, a Petri-net modeling and its reachability tree. Then, we evaluate the performance by simulation.

키워드 : 핸드오프 알고리즘(handoff algorithm), 멀티캐스트 그룹(multicast group), 마이크로 모빌리티(micro mobility), 페트리네트(Petri-net), 성능 분석(performance evaluation)

1. 서 론

이동 통신 및 무선 통신망 기술의 발달로 데이터 서비스 지원이 가능하게 되었으며, 데이터 서비스 기술도 회선교환

방식에 의하던 것이 최근에는 패킷 데이터 서비스 방식으로 진일보하고 있다. 또한 인터넷 사용의 폭발적인 증가와 함께 사용자들은 시간과 장소에 구애받지 않고 고정 망에서 제공되는 여러 응용 서비스와 같은 양질의 이동 서비스를 받기를 원하며, 이런 욕구를 이동통신망에서 수용하려는 노력으로 서비스에 있어서도 다양한 응용을 제공하려는 추세에 있다. 그러나, 이동 통신에서 인터넷 서비스의 도입에 있어서 제공되어야 하는 기능이나 인터넷에서의 이동성 관리는 아직 미약한 것으로 평가되고 있다. 따라서 서비스를 받고 있는 동

† 정 회 원 : 한국전자통신연구원 무선방송연구소 책임연구원
 †† 정 회 원 : 한국전자통신연구원 IMT-2000 개발본부 선임연구원
 ††† 정 회 원 : 한국전자통신연구원 이동관리연구실장
 †††† 준 회 원 : 서경대학교 대학원 컴퓨터학과
 ††††† 종신회원 : 서경대학교 정보통신공학과 교수
 †††††† 종신회원 : 서경대학교 컴퓨터학과 교수
 논문접수 : 2001년 4월 3일, 심사완료 : 2001년 10월 22일

안 사용자의 위치 변화에 대한 이동성 관리방법에 따라 망의 효율과 서비스의 질이 결정되기 때문에 이동성 관리에 대한 연구가 중요시되고 있으며, 이동 사용자의 이동시 발생하는 핸드오프 처리등 이동성 관리에 대한 연구가 활발히 진행되고 있다.

이동성 관리 지원을 위해 IETF(Internet Engineering Task Force)에서는 Mobile IP를 규정하고 있으며[6], 그 외 HAWAII(Handoff Aware Wireless Access Internet Infrastructure)[7]와 Cellular IP[11]등이 있다. 인터넷과의 연동 및 이동성 지원을 위한 기술들을 표준화하는 과정에 있으며, 그 대표적인 것이 북미계열의 3GPP2 그룹과 유럽계열의 3GPP 그룹이 있다. 이동성 관리는 망의 크기에 따라, 수 킬로미터 이상인 Macro 망과 수 백 미터 내외인 Micro 망에서의 이동성 관리로 나눌 수 있다. 3GPP2 방식은 PDSN(Packet Data Serving Node)이 FA(Foreign Agent)의 역할을 수행하므로 이동성 지원이 가능하지만, 이는 Macro Mobility을 지원하기 위한 메커니즘으로, Micro Mobility를 지원하기엔 빠르고 seamless한 핸드오프에 취약하다. 즉, 데이터의 손실 없이 지속적인 서비스를 제공하기에는 부족한 면이 있다. 따라서, 본 논문에서는 Micro 망에서의 사용자의 이동시 seamless한 서비스의 제공을 위하여 멀티캐스트 그룹 메커니즘 기반의 seamless 핸드오프 알고리즘을 제안한다.

본 논문은 데이터 서비스를 제공하기 위해서 Micro Mobility를 지원하는 멀티캐스트 그룹 메커니즘 기반의 seamless 핸드오프 알고리즘을 제안하는 논문으로, 2장에서는 제안하는 멀티캐스트 그룹 메커니즘 기반의 seamless 핸드오프 알고리즘에 대해서, 3장에서는 seamless 핸드오프 알고리즘의 처리 절차 및 검증에 대해서 논하며, 4장에서는 시뮬레이션 결과를 통하여 제안한 알고리즘의 성능분석을 하며, 마지막으로 결론을 맺는다.

2. 멀티캐스트 그룹 메커니즘 기반의 seamless 핸드오프 알고리즘

PDSN 영역내의 여러 RN(Radio Network)간 핸드오프를 제공하기 위하여 멀티캐스트 그룹 메커니즘을 적용할 경우, 핸드오프 처리 지연 시간이 짧아지게 되므로 이동 호스트에 seamless한 연결 서비스를 제공하기에 적합하다. 그러나, 멀티캐스트 그룹에 속한 모든 RN이 이동호스트로 전송할 데이터를 미리 저장하고 있어야 하므로 RN의 버퍼 오버헤드가 큰 것이 단점이다.

본 논문에서 제안하는 멀티캐스트 그룹 메커니즘 기반의 seamless 핸드오프 알고리즘은 MS(Mobile Station)에게 seamless한 연결 서비스를 제공할 뿐만 아니라, 기존 멀티캐스트 그룹 메커니즘[5, 7, 10]을 적용했을 경우에서 오는 버퍼 오버헤드를 줄이는 것과 버퍼 부족으로 인한 핸드오프 호의 블럭

킹 확률을 줄이는 것을 목표로 한다.

2.1 핸드오프 처리 절차

MS는 MS가 존재하는 영역을 관리하는 PDSN과 RN에 의해 서비스를 받고 있으며, MS가 RN영역을 이동 중일 경우, MS가 존재하는 영역에 인접한 RN의 신호 세기를 얻을 수 있으며, 이를 RN을 통해 PDSN에 보고한다고 가정할 때, 제시한 멀티캐스트 그룹 기반의 핸드오프 알고리즘의 핸드오프 처리 절차 과정을 살펴보면 다음과 같다.

- ① MS는 PDSN에게 자신의 위치정보를 보고하며, PDSN은 이에 대한 정보를 저장한다. 위치정보는 MS의 번호, 보고 시간, 인접 RN의 신호 세기로 구성된다.
- ② PDSN은 MS로부터 받은 위치정보와 PDSN내의 RN의 배치 정보를 가지고 있는 지역정보 및 이동환경정보를 이용하여 그룹의 크기 및 MS의 이동방향과 속력을 예측한다.
- ③ PDSN은 예측한 MS의 이동방향과 속력을 이용하여, MS의 예상 이동 경로상의 RN들을 멀티캐스트 그룹으로 구성한다. 이때 PDSN은 각 멤버 RN으로의 예상 핸드오프 시간을 계산하고 이를 각각의 RN에게 알려 준다. 예상 핸드오프 시간은 RN영역 내에서 실제 측정된 RN의 신호 세기를 유지하는 배치정보와 신호세기를 이용하여 MS의 위치를 예측하고, MS의 위치에서 그룹에 속한 각 RN들의 경계까지 거리를 구한 다음 이를 MS의 속력으로 나누어 각 RN들의 예상 핸드오프 시간을 예측한다. 이때 속력은 배치정보를 이용하여, MS가 주기적으로 보고 하는 위치정보에 의해 신호의 세기에 따른 위치를 예측하고, 보고시간에 따른 거리의 차를 보고시간의 차로 나누어 속력을 구한다.
- ④ PDSN은 MS에게 전송할 패킷을 멀티캐스트 그룹의 멤버 RN들에게 전송하며, MS가 현재 위치한 RN은 패킷을 MS에게 전송한다. 멤버 RN들은 각각의 핸드오프 예상 시간 이후의 패킷만을 저장한다.
- ⑤ MS가 인접 RN으로 이동하고, MS는 자신이 마지막으로 전송받은 패킷의 번호를 이동한 RN에게 알린다. 새로운 RN은 저장되어 있는 패킷 중에서 마지막 번호 이후의 패킷을 MS에게 전송한다. 따라서, 멀티캐스트 연결을 이용하여 핸드오프를 처리하기 때문에 MS는 데이터의 seamless한 서비스를 RN으로부터 받을 수 있으며, RN은 예상 핸드오프 이후의 패킷만을 저장하므로 RN의 버퍼 오버헤드를 줄일 수 있다.

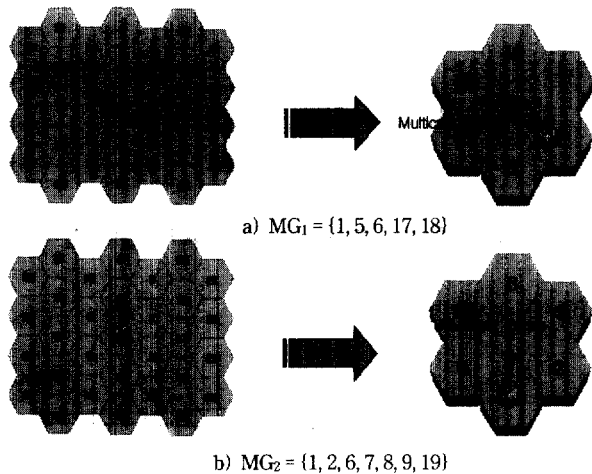
2.2 멀티캐스트 그룹 구성

각 PDSN은 여러 개의 RN들을 관리하며, RN들의 멀티캐스트 그룹 구성 및 RN과 MS의 연결 관리를 담당한다. 멀티캐스트 그룹을 구성하기 위해서는 먼저 MS의 위치와 속도에

대한 계산이 이루어져야 한다. MS의 속도를 측정하는 방법으로는 서비스 영역, 또는 핸드오프 영역에서 머무르는 시간을 이용하는 방법[12]과 위치측정 방법에는 전파의 도달 시간을 이용하는 방법으로 전파 전달 시간을 측정하여 위치를 구하는 TOA(Time Of Arrival), 그리고 전파 도달 시각의 상대적인 차를 이용하는 TDOA(Time Difference Of Arrival)방법이 있다[9]. 본 논문에서는 기존의 방법중의 하나를 이용하여 MS의 속력과 위치 측정을 하는 것으로 가정한다.

PDSN이 멀티캐스트 그룹을 구성하고, 멤버 RN들을 join 시키는 과정은 다음과 같다.

- ① MS는 수신한 인접 RN의 신호 세기, MS의 번호, 보고 시간 및 RN 번호 등으로 이루어진 위치정보를 주기적으로 RN을 통해 PDSN에게 보고한다.
- ② PDSN은 MS가 주기적으로 보고한 위치정보와 지역정보, 이동환경정보를 이용하여 그룹의 크기 및 MS의 속력과 이동방향을 예측한다.
- ③ PDSN은 예측한 MS의 이동경로상의 인접한 RN들을 선택하여 멀티캐스트 그룹을 구성한다.
- ④ PDSN은 각 멤버 RN에 대해서 MS가 이동할 예상 핸드오프 시간을 계산하여 각 멤버에게 알리고, 멀티캐스트 join을 요청한다. 만약 join 요청이 받아 들여 지지 않았을 경우 계산된 예상 핸드오프 시간 내에서 재요청을 한다. 이때 join 요청 메시지는 예상 핸드오프 시간에서 멤버 RN들을 join하는데 걸리는 전체 시간을 뺀 시점에 보낸다.



(그림 1) 멀티캐스트 그룹구성의 예

(그림 1)은 제안한 핸드오프 방법의 간단한 그룹구성의 예를 보인 것이다. (그림 1(a))는 PDSN이 MS가 RN₅에 가까이 있으며 이동방향은 RN₁에서 RN₆로 예상하여 MS의 멀티캐스트 그룹을 MG₁ = {1, 5, 6, 17, 18}로 구성한 경우이며, (그림 1(b))는 MS의 이동방향을 RN₁에서 RN₇로 예상하여 MS의 멀티캐스트 그룹을 MG₂ = {1, 2, 6, 7, 8, 9, 19}로 구성한 경우이다. 또한 그룹멤버의 RN수, 즉 그룹의 크기는 속력

에 비례하여 이동환경정보에 정의된 크기로 설정한다. 단 이동환경정보의 그룹의 크기는 합당한 값으로 이미 측정되어져 있어야 한다. 즉, MS가 현재 같은 RN이 관리하는 영역 안에 존재한다고 하더라도, MS의 위치와 이동방향, 속력에 따라 각 RN의 신호세기와 그룹의 크기가 달라지므로 PDSN은 멀티캐스트 그룹을 다르게 구성할 수 있다.

2.3 멀티캐스트 그룹 join 시점

멀티캐스트의 그룹 멤버에 속해 있는 RN은 PDSN으로부터 멀티캐스트 join을 요청받는다. 이 멀티캐스트 join 요청 메시지에는 MS의 예상 핸드오프 시간 정보가 들어 있다.

PDSN은 멤버 RN의 join 시점을, MS가 중단없는 연결 서비스를 제공받는데 지장이 없는 범위 내에서 멀티캐스트 join 요청 메시지를 보내는데, 이는 join 시점을 가능한 최대로 지연시켜 망 자원의 낭비를 줄인다. PDSN이 멀티캐스트 join 요청 메시지를 보내는 시점은 다음과 같이 결정한다.

- PDSN이 MS의 이동성에 대한 정보를 이용하여 예상 핸드오프 시간을 계산하고 그룹에 포함시킬 RN을 선택하는데 걸리는 시간을 T_{selectgroup}이라 한다.
- 각 멤버 RN를 멀티캐스트 join 시키는데 걸리는 예상시간을 T_{Join,RN(i)}}라 하고, 모든 멤버 RN중 T_{Join,RN(i)}}가 가장 큰 값을 T_{Join,max}라 한다. 따라서,

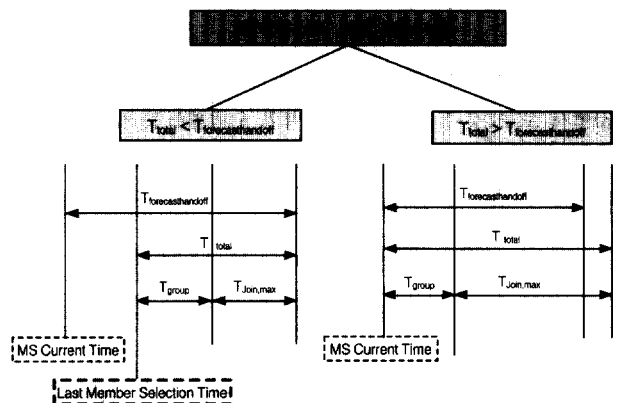
$$T_{Join,max} = \text{MAX}\{T_{Join,RN(i)} \text{ for all member RN}\}$$

- 멤버 RN들이 join하는데 걸리는 전체시간(T_{total})은 T_{Join,max}와 T_{group}의 합산한 시간이 된다.

$$T_{total} = T_{selectgroup} + T_{Join,max}$$

- PDSN이 멀티캐스트 그룹을 구성하기 위해 join 메시지를 보내는 시간(T_{sendJoin})은 MS의 핸드오프 예상시간 T_{forecasthandoff}에서 T_{total}를 뺀 시간이 된다

$$T_{sendJoin} = T_{forecasthandoff} - T_{total}$$



(그림 2) 멀티캐스트 join 시점

멤버 RN에 멀티캐스트 그룹에 대한 join 메시지를 보내는

시간($T_{sendJoin}$)은 MS의 현재 위치와 PDSN이 계산한 핸드오프 시간에 따라 양수 혹은 음수가 나올 수 있다. $T_{sendJoin}$ 의 값이 양수가 나올 경우는, MS가 핸드오프 처리를 요청하기 전에 멀티캐스트 join 작업을 완료할 수 있으므로 각 멤버 RN들은 멀티캐스트 join 작업을 수행한다. 만약 $T_{sendJoin}$ 시간이 음수 값인 경우 예상 핸드오프 시간 내에 멀티캐스트 join 작업이 완료되지 않기 때문에, 제안한 핸드오프 방법으로 연결서비스를 MS에게 제공할 수 없고, 따라서 미리 멀티캐스트 join 작업을 수행하지 않는다.

2.4 버퍼 오버헤드 계산

제안한 핸드오프 알고리즘의 버퍼오버헤드를 구하기 위하여 다음과 같은 몇 가지 변수를 정의한다.

- 정의
 - MGS : 멀티캐스트 그룹의 크기, 즉 멤버 RN의 수
 - T_0 : 그룹 조인이 이루어지고, 데이터가 멀티캐스트되는 시간
 - $T_{handoff}$: 실제 핸드오프가 발생한 시간
 - SHTG(Same Handoff Time Group) : 핸드오프 시간이 같은 RN들로 이루어진 그룹
 - $S_{SHTG(i)}$ (Size of SHTG) : 핸드오프 시간이 같은 RN 그룹의 크기, 즉 그룹에 속한 멤버 RN의 수
 - $T_{SHTG(i)}$ (Time of SHTG) : SHTG의 예상 핸드오프 시간, 멀티캐스트 그룹은 핸드오프의 시간이 같은 RN의 그룹으로 다시 분류된다.
 - BW(Bandwidth) : 연결에 할당된 대역폭

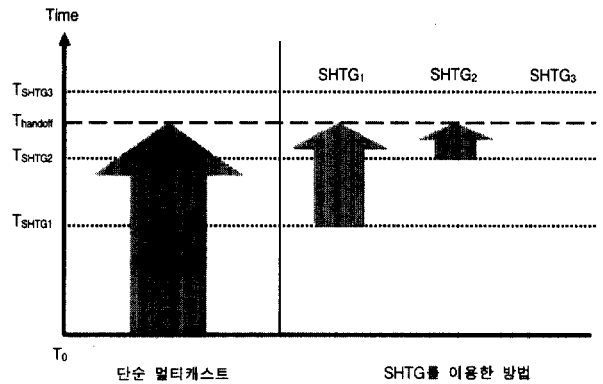
(그림 1(b))에서 멀티캐스트 그룹이 $MG_2 = \{1, 2, 6, 7, 8, 9, 19\}$ 로 구성된 경우의 버퍼 오버헤드를 알아보자. MS의 실제 핸드오프의 시간을 $T_{handoff}$, 각 멤버 RN의 예상 핸드오프 시간을 T_{SHTG1} , T_{SHTG2} , T_{SHTG3} 라고 가정한다.

단순 멀티캐스트의 방법의 경우에는 멀티캐스트그룹에 조인한 모든 멤버 RN들이 현재 MS가 존재하고 있는 RN과 같이 멀티캐스트 되어지는 데이터를 동일하게 저장하게 된다. 그러므로 멀티캐스트 그룹에 속한 모든 멤버 RN들은 (그림 3)에서 보는 것과 같이 동일한 버퍼 오버헤드를 갖는다. 따라서 단순 멀티캐스트 핸드오프 방법의 총 버퍼 오버헤드(TBO : Total Buffer Overhead)를 계산하면 다음과 같다.

$$TBO = BW * (T_{handoff} - T_0) * MGS$$

제안하는 핸드오프 알고리즘에서, 멀티그룹을 $MG_2 = \{\{1\}, \{2, 6, 7\}, \{8, 9, 19\}\}$ 와 같이, 다시 멀티캐스트 그룹의 모든 멤버 RN 중에 MS의 예상 핸드오프 시간이 같은 RN들의 집합인 SHTG로 분류할 수 있다고 하자. 즉, $SHTG_1 = \{1\}$, $SHTG_2 = \{2, 6, 7\}$, $SHTG_3 = \{8, 9, 19\}$ 이고, 각 멤버 RN들은 자신이 속한 SHTG의 예상 핸드오프시간(T_{SHTG1} , T_{SHTG2} ,

T_{SHTG3})을 고려해서 자신의 예상 핸드오프시간($T_{SHTG(i)}$) 이후의 데이터만을 저장한다. (그림 3)에서처럼 SHTG₁에 속한 멤버 RN은 T_{SHTG1} 이후의 데이터를 저장하고, SHTG₂에 속한 멤버 RN은 T_{SHTG2} 이후의 데이터를 저장하고, SHTG₃에 속한 멤버 RN은 T_{SHTG3} 이후의 데이터를 저장한다. 따라서 SHTG방법의 총 버퍼 오버헤드(TBO_{SHTG})는 다음과 같이 된다.



(그림 3) 버퍼 오버헤드 비교

$$TBO_{SHTG} = BW * ((T_{handoff} - T_{SHTG1}) * S_{SHTG1} + (T_{handoff} - T_{SHTG2}) * S_{SHTG2} + (T_{handoff} - T_{SHTG3}) * S_{SHTG3})$$

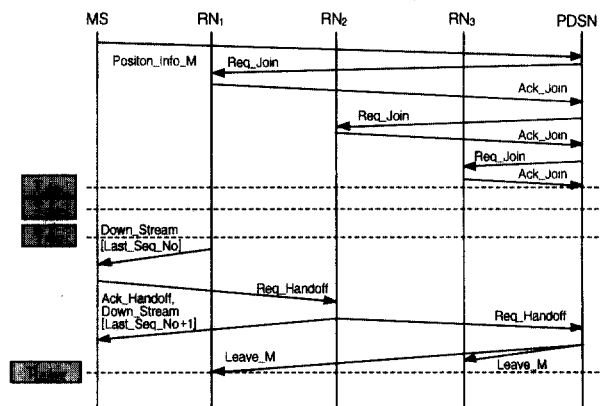
버퍼 오버헤드를 비교하면 다음과 같은 사실이 성립함을 알 수 있다.

[정리] SHTG는 기존의 단순 멀티캐스트 연결을 이용한 방법보다 항상 작은 버퍼 오버헤드를 갖는다.

$$TBO_{SHTG} \leq TBO$$

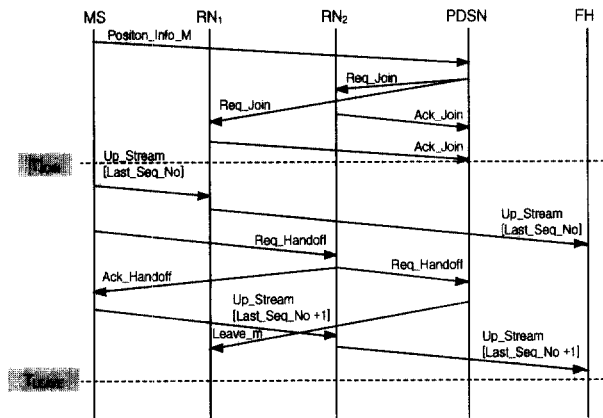
3. Seamless 핸드오프 알고리즘의 처리 절차 및 검증

3.1 핸드오프 처리 절차 흐름도



(그림 4) Down Stream의 경우 핸드오프 처리 절차 흐름도

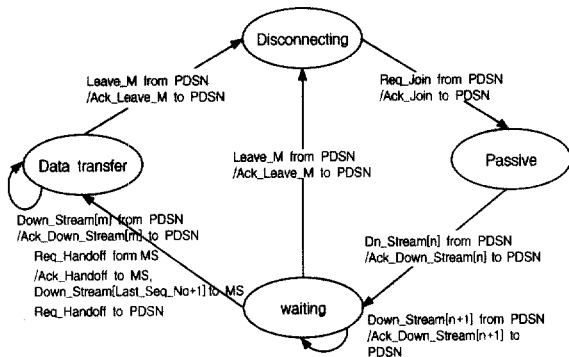
(그림 4)와 (그림 5)는 Up Stream 및 Down Stream의 경우 핸드오프 처리 절차 흐름도 이다. Up Stream은 MS가 패킷을 RN을 통하여 PDSN에 전송하는 상태를 의미하며, Down Stream은 그 반대의 경우를 의미한다. T_{JOIN} 은 멀티캐스트 그룹에 속한 모든 RN의 join이 끝난 시점을 말하며, T_{RN2} 와 T_{RN3} 은 MS가 RN_2 , RN_3 로의 핸드오프 예상시간을 나타낸다. Down_Stream[Last_Seq_No]은 MS가 RN_1 으로부터 받은 마지막 데이터를 나타내며, MS는 RN_2 로 핸드오프 요구 메시지 Req_Handoff를 보내면서 MS의 핸드오프의 처리절차는 시작된다.



(그림 5) Up Stream의 경우 핸드오프 처리 절차 흐름도

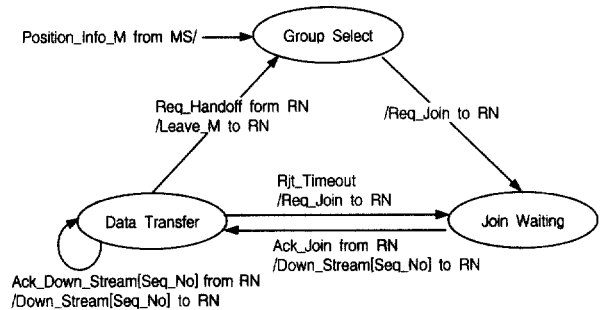
3.2 RN간 seamless 핸드오프 알고리즘의 State Transition Diagram

MS가 PDSN영역내의 여러 RN간의 이동에 있어 중단 없는 서비스를 받기 위하여 앞 절에서 제안, 설계한 멀티캐스트 그룹 메커니즘 기반의 seamless 핸드오프 알고리즘에 대하여 본 절에서는 PDSN과 RN간의 상호 동작하는 핸드오프 처리 절차를 State Transition Diagram으로 표시한다. Down Stream의 경우만 기술하고, Up Stream의 경우는 생략한다. RN은 Disconnecting 상태에 있다가 PDSN으로부터 join 요청 메시지를 받으면 Ack_Join 메시지를 되돌려 주면서 state transition이 시작된다. 자세한 state transition은 (그림 6)과 같다.



(그림 6) RN의 State Transition Diagram

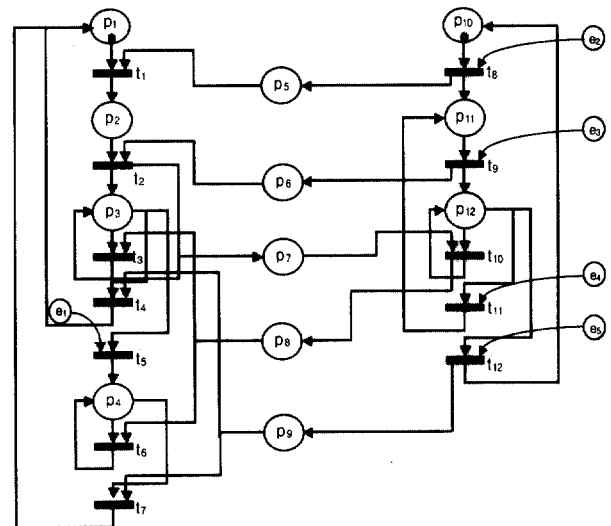
(그림 7)은 MS가 상대방 호스트로부터 데이터 서비스를 받을 때 PDSN의 state transition을 나타낸 것으로, RN통해 MS로부터 위치정보 메시지를 받으면 그룹 멤버를 결정하고 각 멤버에게 join 요청 메시지를 보내면서 Join Waiting 상태로 state transition이 시작된다.



(그림 7) PDSN의 State Transition Diagram

3.3 페트리넷(Petri-Net) 모델 및 도달성 트리

본 절에서는 3.2절의 State Transition Diagram을 토대로 RN과 PDSN 사이의 연동을 페트리넷을 이용하여 모델링하고 검증한다. (그림 8)은 Down Stream의 경우에 PDSN과 RN간의 메시지와 데이터 흐름 처리 절차에 대한 state transition을 이용하여 도시한 페트리넷 모델이다. 단, Up Stream의 경우는 생략한다. 그림에서 좌측의 토큰소는 RN의 상태, 우측의 토큰소는 PDSN의 상태, 중간의 토큰소는 메시지 전달 중인 상태를 표현한다. p_1 과 p_{10} 의 검은 점은 초기 토큰을 나타낸다. (그림 8)에서 각 토큰소의 상태는 메시지 처리 절차에 따라서 다른 상태로 이동하며, 이를 도표로 정리하면 <표 1>과 같다. 또한 외부사건(event)은 'e_i'로 표시하며, 이를 <표 2>에 정리한다.



(그림 8) Down Stream의 경우 RN과 PDSN의 페트리넷 모델링

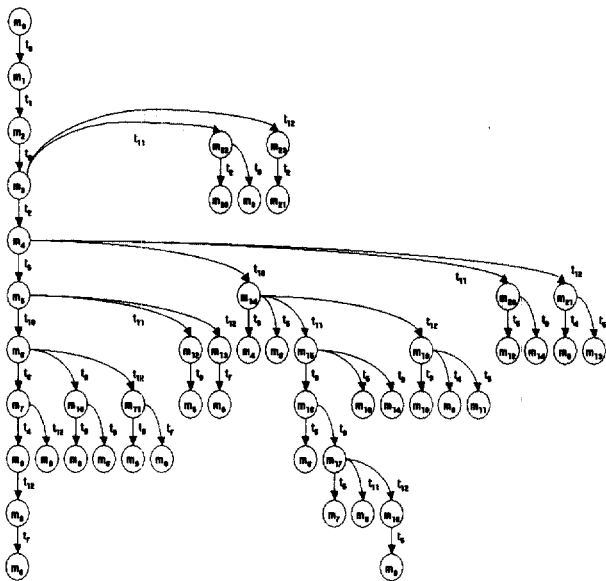
<표 1> Down Stream의 경우 RN과 PDSN간의 메시지 처리 절차에 대한 토큰소 상태

토큰 소	메시지 처리 절차에 따른 상태
p1	Disconnecting
p2	Passive
p3	Waiting
p4	Data Transfer
p5~p9	메시지, 데이터 전달 과정
p10	Group Select
p11	Join Waiting
p12	Data Transfer

<표 2> Down Stream의 경우 RN과 PDSN간의 메시지 처리 절차에 대한 외부 사건

외부사건	메시지 처리 절차에 따른 외부사건
e1	Req_Handoff from MS
e2	Position_Info_M from MS(p1을 통해서 오는 event)
e3	Ack_Join from RN
e4	Rjt_Timeout (Req_Join Send Time)
e5	Req_Handoff from MS'

알고리즘의 Deadlock Freeness 및 생동성(Liveness)에 대한 적합성을 검증하기 위한 도달성 트리(Reachability Tree)는 Down Stream의 경우 (그림 8)으로부터 구해지며, 그 결과는 (그림 9)와 같다. 그림에서 볼 수 있듯이 초기상태 m0에서 시작한 트리는 상태 m23까지 transition이 발생한다. Down Stream 경우 RN과 PDSN 페트리네트 모델의 도달성 트리 구조에서 각 상태는 서로 다른 토큰소에 존재하는 토큰으로 구분되는 상태 집합이며, 각 상태는 <표 3>과 같은 상태 집합을 갖게 된다.



(그림 9) Down Stream의 경우 RN과 PDSN 페트리네트 모델의 도달성 트리

m0는 p1과 p10에 토큰을 갖는 초기 상태이며, (그림 9)에서 볼 수 있듯이 transition t8에 의해 m1 상태로 전이된다. (그림 9)의 도달성 트리에 의해, deadlock 발생없이 항상 초기 상태로 돌아올 수 있음을 확인 할 수 있으며, 초기 상태에서 모든 상태로의 transition이 가능하며, 또한 발생하는 토큰의 수도 항상 1 이하로 제한되어 안정성이 만족됨을 확인할 수 있다. Up Stream의 경우에 대해서도 동일한 결과를 얻을 수 있음을 확인하였으며, 그에 대한 기술은 생략한다.

<표 3> Down Stream의 경우 RN과 PDSN 페트리네트 모델의 도달성 트리 상태 집합

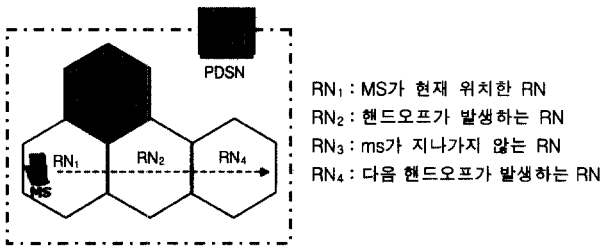
	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
m0	1									1		
m2	1				1						1	
m3		1									1	
m4		1				1						1
m5				1			1					1
m6				1				1				1
m7				1								1
m8				1								1
m9				1					1	1		
m10				1				1			1	
m11				1				1	1	1		
m12				1			1				1	
m13				1			1		1	1		
m14			1					1				1
m15			1					1			1	
m16			1								1	
m17			1									1
m18			1						1	1		
m19			1					1	1	1		
m20			1						1	1		
m21			1				1					1
m22			1				1		1	1		
m23	1					1						1
m24	1					1			1	1		

4. 제안한 알고리즘의 시뮬레이션

본 장에서는 PDSN영역내의 RN간 seamless한 핸드오프의 제공을 위해 제안한 알고리즘방법의 성능 평가를 위한 시뮬레이션을 하며, 시뮬레이션은 멀티캐스트 매커니즘을 적용하지 않았을 경우와, 단순 멀티캐스트 매커니즘을 적용하였을 경우의 핸드오프 처리 절차를 비교하여 성능을 평가한다.

4.1 시뮬레이션 환경 구성

PDSN이 관리하는 영역안에 RN1, RN2, RN3, RN4가 존재하며 각 RN이 관리하는 영역의 크기는 동일하며 반경은 단위거리 5로 설정하였다.



(그림 10) MS의 이동경로

4.1.1 이동 경로

- ① MS는 초기에 RN₁으로부터 서비스를 받으며 RN은 시간이 변함에 따라 일정한 속도로 이동한다.
- ② MS의 이동방향은 초기 RN₁에서 RN₄로 직선으로 이동한다고 가정하며, 이때 RN₃는 MS가 서비스를 받고 있는 RN₁에 인접하여 있지만 MS의 이동경로 상에 존재하지 않는다.

4.1.2 신호세기

MS의 시간에 변화에 따라 초기 RN₁에서 이동한 거리에 의해 이동을 표현하며, MS 위치에서의 각 RN 신호의 세기는 다음과 같이 정의한다.

- ③ MS가 RN 영역내의 중앙에 위치할 경우 신호의 세기값은 최대값 25를 갖는다.
- ④ RN의 신호세기는 RN의 중앙에서부터 단위거리당 1씩 감소한다.
- ⑤ RN의 배치 구조는 (그림 10)에서 보는 것과 같으며, MS가 얻는 각각의 RN의 신호 세기는 MS가 직선으로 이동하고, RN의 신호세기는 단위거리당 1씩 감소하므로 MS가 이동한 거리를 X라 하면, MS의 이동에 따른 RN의 신호세기는 다음과 같이 구할 수 있다.(MS의 위치에서 각

RN의 중앙까지의 거리를 구한다면 최고의 신호세기 25에서 뺀 값으로 구한다.)

- RN₁의 신호세기 = $25 - |X - 5|$
- RN₂의 신호세기 = $25 - |X - 15|$
- RN₃의 신호세기 = $25 - \sqrt{(x-10)^2 + 75}$
- RN₄의 신호세기 = $25 - |X - 25|$

위의 식을 이용하여 <표 4>의 값을 구할 수 있다. MS에서 각 RN의 신호의 세기값은 이동한 거리에 대한 <표 4>에서와 같은 신호의 세기를 갖는다.

4.2 시뮬레이션 방법

MS는 초기에 RN₁에서 파일을 전송받다가 이동중에 RN₂, RN₄로 핸드오프가 발생하게 되고, RN₁으로부터 받던 데이터는 핸드오프가 발생 후 RN₂, RN₄에서 받게 된다. 이때 파일의 데이터를 모두 받았을 때까지의 서비스 중단 시간과 버퍼 오버헤드를 비교 분석한다.

4.2.1 MS의 서비스 중단 시간

MS가 현재 서비스받고 있는 RN이 관리하는 영역에서 New RN이 관리하는 영역으로 이동하여 핸드오프가 발생했을 때, 핸드오프 발생시점에서 New RN으로부터 서비스를 받는 시점까지의 시간 차를 서비스 중단 시간으로 판단한다.

4.2.2 그룹에 속하는 멤버 RN들의 버퍼 오버헤드

버퍼 오버헤드는 seamless한 핸드오프를 제공하기 위하여 RN이 미리 버퍼링하는 데이터의 양으로, 각 RN이 MS가 자신의 관리 영역으로 이동할 것을 대비해서 미리 데이터를 버퍼링하는 양을 의미한다. 그러므로 버퍼 오버헤드는 각 RN에서의 PDSN이 미리 보낸 데이터의 버퍼링양으로 판단한다.

4.3 시뮬레이션의 핸드오프 처리 절차

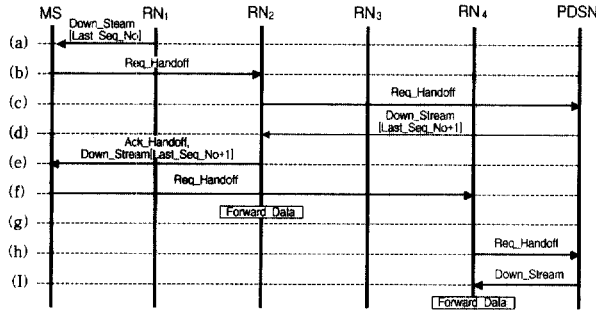
MS가 이동중 RN₁에서 서비스를 받다가 핸드오프가 발생한 경우, 시뮬레이션을 멀티캐스트 메커니즘을 적용하지 않은 경우, 단순 멀티캐스트인 경우 및 제한한 알고리즘의 경우로 분리하여 수행한다. (그림 11)은 멀티캐스트 메커니즘을 적용하지 않은 경우로 핸드오프 요구 메시지를 보내는 시점 (b)에서부터 핸드오프 요구가 받아들여진 RN으로부터 다음 데이터를 받는 (e)까지가 서비스 중단 구간이 된다. New RN은 미리 버퍼링하고 있는 데이터가 존재하지 않으므로 버퍼 오버헤드는 없다.

(그림 12)는 단순 멀티캐스트 메커니즘을 적용한 경우로, 그룹에 속한 각 RN은 MS의 이동을 대비해 미리 데이터를 버퍼에 저장하게 된다. 핸드오프가 발생하는 시점 (c)와 MS가 이동하여 핸드오프 요구가 받아들여진 RN으로부터 다음 데이터를 받는 시점 (e)사이가 서비스 중단 구간이 되며, 그룹에 속한 멤버 RN중 RN₂는 데이터를 저장하기 시작하는

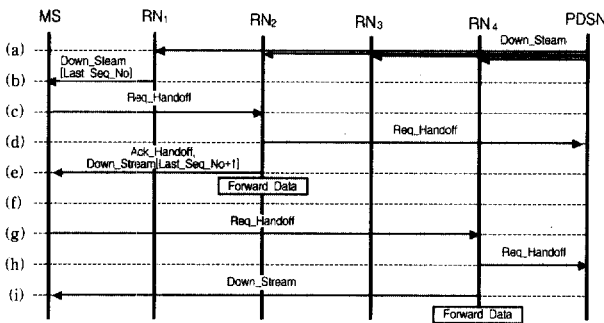
<표 4> MS의 이동거리에 따른 각 RN의 신호세기

거리	1	2	3	4	5	6	7	8	9	10
RN ₁	21	22	23	24	25	24	23	22	21	20
RN ₂	11	12	13	14	15	16	17	18	19	20
RN ₃	13	13	14	14	15	15	16	16	16	16
RN ₄	1	2	3	4	5	6	7	8	9	10
거리	11	12	13	14	15	16	17	18	19	20
RN ₁	19	18	17	16	15	14	13	12	11	10
RN ₂	21	22	23	24	25	24	23	22	21	20
RN ₃	16	16	16	15	15	14	14	13	13	12
RN ₄	11	12	13	14	15	16	17	18	19	20
거리	21	22	23	24	25	26	27	28	29	30
RN ₁	9	8	7	6	5	4	3	2	1	0
RN ₂	19	18	17	16	15	14	13	12	11	10
RN ₃	11	10	9	9	8	7	6	5	4	3
RN ₄	21	22	23	24	25	24	23	22	21	20

시점 (a)부터 (e)까지의 사이가 버퍼 오버헤드가 되고, RN₃는 (a)부터 그룹탈퇴 메시지를 받는 시점까지, RN₄는 (a)부터 서비스를 제공하는 시점(i)까지가 버퍼 오버헤드가 된다.

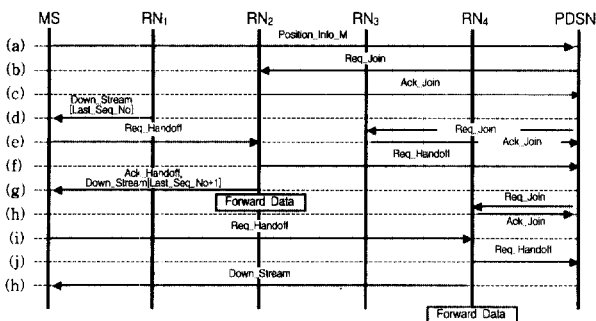


(그림 11) 멀티캐스트 메커니즘을 적용하지 않았을 경우 핸드오프 처리 흐름도



(그림 12) 단순 멀티캐스트 메커니즘의 경우 핸드오프 처리 흐름도

(그림 13)은 제안한 알고리즘의 경우 핸드오프 처리 흐름도이다. MS는 위치정보를 PDSN에게 보고하고, PDSN은 이를 이용하여 그룹을 설정, 조인을 하며, 각 그룹에 속한 각 RN에게 예상 핸드오프시간 이후의 데이터를 전송하게 된다. 이때 각 RN들은 MS의 이동을 대비하여 예상 핸드오프시간 이후의 데이터를 버퍼링한다. 서비스 중단 구간은 핸드오프가 발생하는 시점 (e)부터 다음데이터를 받는 시점(g) 사이가 되며, 그룹에 속한 각 멤버 RN의 버퍼 오버헤드는, RN₂의 경우 예상 핸드오프시간 이후의 데이터를 받는 (c)시점부터 MS의 요청에 의해 다음 데이터를 전송하는 시점인(g) 사이



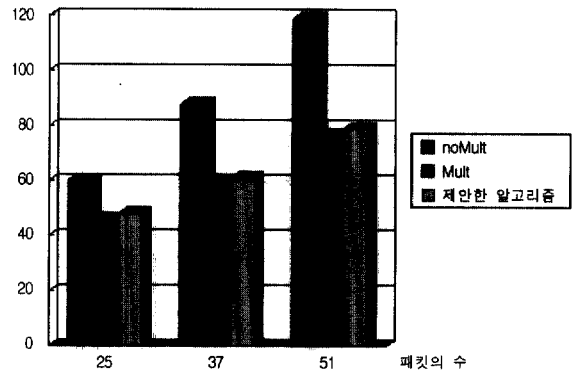
(그림 13) 제안한 알고리즘의 경우 핸드오프 처리 흐름도

이며, RN₃의 경우는 (e)부터 (h)사이, RN₄는 (h)부터 (k)까지가 된다.

4.4 시뮬레이션 결과

(그림 14)는 데이터 패킷수에 따라 서비스를 받는 시점부터 서비스가 완료될 때까지 걸리는 시간이다. (그림 14)에서 보듯이 멀티캐스트 메커니즘을 적용하지 않은 경우는 데이터 패킷의 수가 25, 37, 51일 때 시간변화는 60, 88, 119초로, 미리 데이터를 버퍼링하고 있지 않기 때문에 그만큼의 시간이 더 필요하게 되며, 단순 멀티캐스트일 경우와 제한한 알고리즘의 경우는 46, 59, 77과 48, 61, 79로 시간 변화에 있어서 거의 비슷한 결과를 얻을 수 있다.

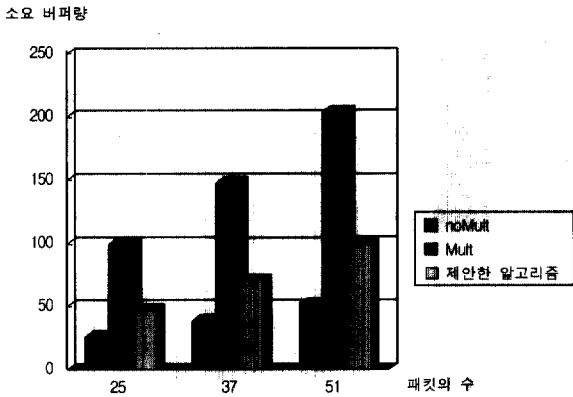
처리시간



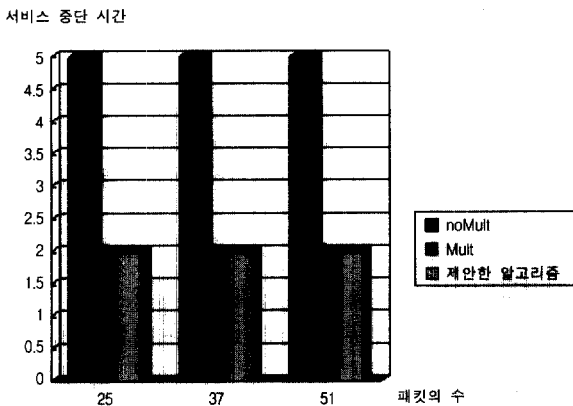
(그림 14) 패킷수와 처리시간의 관계

(그림 15)는 소요된 버퍼량의 변화이다. 멀티캐스트 메커니즘을 적용하지 않은 경우, MS가 핸드오프 전에 데이터 패킷을 요청하였으나 데이터 패킷을 받지 못한 상태에서 핸드오프가 발생하였을 때, 그것으로 인해 소요되는 버퍼량을 제외하고는 패킷의 수와 같은 버퍼량을 보이고 있다. 그러나 단순 멀티캐스트의 경우는 미리 데이터를 저장하고 있기 때문에 버퍼량이 패킷수 곱하기 멀티캐스트 그룹에 속한 RN 수 만큼 증가함을 알 수 있다. 그러나, 제안한 알고리즘은 그룹에 속한 멤버중 MS가 지나가지 않는 RN의 소요 버퍼량과 예상 핸드오프시간과 실제 핸드오프 시간차에서 오는 버퍼 소요량을 제외하고는 패킷의 수와 거의 같은 버퍼량을 보이고 있다. 즉 제안한 알고리즘에서 버퍼의 오버헤드는 MS가 지나가지 않는 멤버에서 오는 버퍼량과 예상핸드오프와 실제 핸드오프 시간차에서 오는 버퍼량의 합이다.

(그림 16)의 경우 핸드오프가 발생하였을 경우 MS에서의 서비스 중단 시간을 얻은 것으로 멀티캐스트 메커니즘을 적용하지 않았을 경우는 데이터를 미리 저장하고 있지 않기 때문에 미리 데이터를 저장하고 있는 단순 멀티캐스트와 제안한 알고리즘보다는 서비스 중단 시간이 길어짐을 알 수 있다.



(그림 15) 패킷수와 소요 버퍼량의 관계



(그림 16) 서비스 중단시간

시뮬레이션의 결과로 멀티캐스트 메커니즘을 적용하지 않았을 경우 서비스의 처리시간이 지연될 뿐 아니라 seamless한 핸드오프 처리를 하기에는 무리가 있다. 그러나 멀티캐스트 메커니즘을 적용한 단순 멀티캐스트인 경우 서비스 처리시간과 서비스 중단시간의 감소됨을 볼 수 있지만, 모든 그룹의 멤버들이 데이터를 저장하고 있어야 하는 버퍼의 오버헤드를 가지고 있는 단점을 가지고 있다. 이에 반해 제안한 알고리즘은 서비스 처리시간과 서비스 중단시간이 감소됨을 볼 수 있으며, 예상 핸드오프시간 이후의 데이터만을 저장하기 때문에 단순 멀티캐스트에서 오는 버퍼의 오버헤드를 줄일 수 있다.

5. 결 론

본 논문은 하나의 PDSN 영역 안에서 MS가 RN간을 이동할 때 seamless한 핸드오프를 처리하는데 중점을 두고 있다. 빠른 핸드오프처리를 위해 멀티캐스트 그룹 메커니즘을 적용하였는데, 이는 RN들을 멀티캐스트 그룹으로 만들고 이들 그룹에 속해 있는 멤버 RN들은 핸드오프시 필요한 패킷들을 버퍼링하고 있다. 따라서, MS가 그룹에 속한 RN으로 이동하였을 경우 seamless하게 데이터 서비스를 받을 수 있으며,

데이터의 손실 또한 없다. 그러나, 멀티캐스트 그룹에 속한 모든 멤버 RN들은 데이터를 버퍼링하고 있어야 한다는 단점이 있다. 본 논문을 통해 제안된 알고리즘은 MS의 이동방향과 속도를 계산하여 예상 이동 경로에 인접한 RN들을 선택하는 방법으로 멀티캐스트 그룹으로 구성하고, 그룹 join시점을 MS의 예상 핸드오프 시간을 예상해서 Seamless한 연결 서비스에 지장없는 범위에서, 최대한 join시점을 늦춤으로서 망의 효율성을 높인다. 또한, 기존의 멀티캐스트 연결 방법이 가지고 있는 버퍼 오버헤드에 대한 문제점을 해결하기 위해, PDSN은 예상 핸드오프 시간 이후의 데이터만을 전송하며, RN 또한 예상 핸드오프 시간 이후의 데이터만을 버퍼링한다. 제안된 멀티캐스트 그룹 메커니즘 기반의 seamless 핸드오프 알고리즘의 Deadlock Freeness 및 Liveness를 검증하기 위하여 State Transition Diagram, 페트리네트 및 도달성 트리를 이용하였으며, 시뮬레이션을 통하여 버퍼 오버헤드가 낮아지는 결과를 보였다. 추후에 속도와 이동방향에 따라 멀티캐스트 그룹의 크기나 그룹에 속할 RN을 선택하는 방법에 대해 연구가 필요하다.

참 고 문 헌

- [1] U. Dropmann, X. Lagrange, and P. Godlewski, "Architecture of a Multi-Cell Centralized Packet Access System," Proceedings of PIMRC'95, pp.279-283, 1995.
- [2] K. Kawabata, I. Nakamura, E. Fukada, "Estimating velocity using diversity reception," IEE VTC'94 pp.371-374, 1994.
- [3] K. Keeton, B. A. Mah, S. Sehan, R. H. Karz, and D. Ferrari, "Providing Connection-oriented Network Services to Mobile Hosts," Proceedings of the USENIX Symposium, August, 1993.
- [4] O. Kennenmann, "Locating Mobiles in Non-Flowing Traffic," Proceedings of PIMRC'95, pp.274-278, 1995.
- [5] Y. Kim, K. Lee, and Y. Chin, "Effect of Handoff Area Variation on PCS System Traffic," Proceedings of IEEE International Conference on Personal Wireless Communications, pp.134-139, February, 1996.
- [6] C. Perkins, "IP Mobility Support," Internet RFC 2002, Oct. 1996.
- [7] G. P. Pollini, "Trends in handover design," IEEE Communication Magazine, pp.82-90, 1996.
- [8] R. Ramjee et al., "IP Micro-Mobility Support using HAWAII," Internet Draft, Jul. 2000.
- [9] Louis A. Stulp "TDOA technology for locating narrow-band cellular signals," Mobile Radio Technology, Vol.15, No.4, 1997.
- [10] C. W. Sung, W. S. Wong, "User speed estimation and dynamic channel allocation in hierarchical cellular system," IEEE VTC'94, pp.91-95, 1994.
- [11] A. G. Valko, et al., "Mobile IP Regional Registration," Internet Draft, draft-ietf-mobileip-reg-tunnel-02.txt, March, 2000.



신 동 진

e-mail : djsin@etri.re.kr
1978년 충남대학교 전자공학과(학사)
1980년 서울대학교 대학원 전자공학과
(석사)
1990년 서울대학교 대학원 전자공학과
(박사)

1982년~현재 한국전자통신연구원 무선방송연구소 책임연구원
관심분야 : 이동통신 네트워크, 무선 멀티미디어, 음성신호처리



김 수 창

e-mail : sckim@etri.re.kr
1986년 홍익대학교 전자계산학과(학사)
1995년 충남대학교 전산학과(석사)
2000년 충북대학교 정보통신공학과
(박사수료)
1988년~현재 한국전자통신연구원 IMT-2000
개발본부 선임연구원

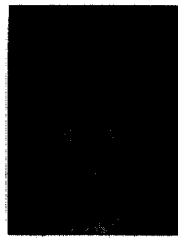
관심분야 : 이동통신네트워크, 이동통신프로토콜



임 선 배

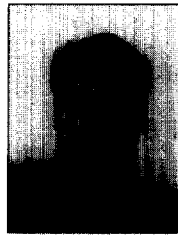
e-mail : sblim@etri.re.kr
1978년 고려대학교 전자공학과(학사)
1989년 한국과학기술원 전산학과(석사)
1993년 고려대학교 대학원 전자공학과
(박사)
1979년~1984년 금성사/금성반도체 선임연
구원

1984년~1992년 한국전자통신연구원 선임연구원
1992년~현재 한국전자통신연구원 이동관리연구실장, 핵심망기술
연구부장, 현 차세대이동통신연구부 책임연구원
관심분야 : 이동통신네트워크, 이동통신프로토콜



오 재 천

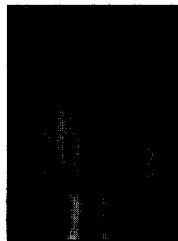
e-mail : jyoh@rta.skuniv.ac.kr
2000년 서경대학교 컴퓨터과학과(학사)
2000년~현재 서경대학교 대학원 컴퓨터과
학과 석사 과정
관심분야 : 이동통신네트워크, 이동통신프로
토콜



송 병 권

e-mail : bksong@skuniv.ac.kr
1984년 고려대학교 전자공학과(공학사)
1986년 고려대학교 대학원 전자공학과
(공학석사)
1995년 고려대학교 대학원 전자공학과
(공학박사)

1984년~1991년 삼성종합기술원 선임연구원
1995년~현재 서경대학교 정보통신공학과 조교수
관심분야 : 고속망 프로토콜, 분산처리시스템, 이동 컴퓨팅



정 태 의

e-mail : tejeong@skuniv.ac.kr
1979년 고려대학교 전자공학과 (학사)
1982년 미국 오하이오주립대 전기공학과
(석사)
1989년 미국 오클라호마대학 전산학과(석사)
1994년 미국 오클라호마대학 전산학과(박사)

1983년~1986년 금성반도체연구소 컴퓨터부문 선임연구원
1986년~1987년 United Microtek, Inc.(San Jose, California)
Engineering Manager
1995년~현재 서경대학교 컴퓨터과학과 조교수
관심분야 : Computational Complexity, Formal Languages, Graph
Languages, 알고리즘, 이동통신