

# 효율적인 채널 사용을 위한 호 수락 제어 분석

이명희, 이준구, 손혁민, 이상훈  
연세대학교  
wireless@yonsei.ac.kr jglee9@nate.com  
sbpgood@hanmail.net slee@yonsei.ac.kr

## Call Admission Control Analysis for Efficient Channel Utilization

### 요 약

본 논문은 다양한 서비스 클래스 유저에게 QoS를 제공하기 위해 OFDM 4G 시스템의 downlink에서 광대역주파수를 효율적으로 사용하기 위한 호 수락 제어 방안에 대하여 살펴본다. n차원의 서비스 클래스가 존재할 때, 고정채널할당과 가변채널할당 두 가지 상황에서 stationary state 확률을 구하고, user의 차단 확률 및 채널 사용 효율을 수학적으로 분석한다. 다양한 서비스 클래스가 존재하는 경우 가변 채널을 할당해 호 수락 제어를 수행하는 것이 고정 채널을 할당하는 것 보다 user의 blocking probability가 더 낮고, 채널 사용효율이 더 좋음을 시뮬레이션 결과로 보였다.

### 1. 서론

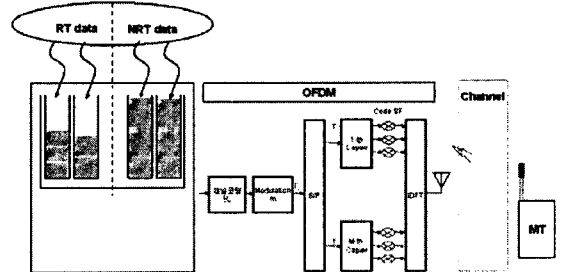
데이터 서비스가 본격적으로 시행되는 WCDMA와 cdma2000 같은 3세대 이동 통신에서 무선 자원의 효율적 사용은 중요한 이슈이다. 이런 무선 시스템을 통한 멀티미디어 서비스의 구현을 위해 시스템의 효율을 극대화해야 한다. 즉 시스템은 각기 다른 지연, 효율 등의 QoS(Quality of Service)를 제공해야 한다.[1] QoS는 RRM(Radio Resource Management)에서 수행하는 호 수락 제어 될 수 있다. 호 수락 제어의 목적은 시스템의 안정성과 높은 시스템의 효율성을 보장하는 것이다.

그 동안 호 수락 제어 방법에 대한 많은 연구가 이루어져왔다. [2]에서는 CDMA 시스템에서 기지국의 송신전력에 기반을 둔 하향링크 호 수락 제어가 연구되었다. [3]에서는 기지국의 평균 송신 전력 및 패킷의 지연 시간이 고려된 호 수락 제어 방법이 제안되었다. 본 연구에서 시스템은 비실시간 서비스를 위한 무한 대기 큐를 가진다. 그리고, 각 다양한 클래스에 따른 고정 채널 할당과 가변 채널 할당에 대한 분석을 한다. 고정 채널 할당이란, 실시간 서비스 user와 비실시간 서비스 user 두 종류의 서비스 클래스가 존재할 때 각 채널을 전용으로 사용하는 경우를 나타낸다. 그래서, 많은 비실시간 서비스 user가 채널을 사용 중 일때는 비실시간 서비스 user가 모두 서비스 될 때까지 실시간 user의 호는 차단되어 drop된다. 그러나, 가변 채널 할당은 많은 비실시간 서비스 user가 채널을 사용 중이더라도, 새롭게 발생된 실시간 서비스 user가 비실시간에 할당된 채널을 점유할 수 있도록 한 것이다. 이렇게 하면, 고정 채널 할당 보다 가변 채널 할당할 경우 user의 호 차단 확률은 줄어들고, 채널의 utilization이 높은 것은 명백하다. 즉, 이 논문은 downlink에서 기지국에 할당된 채널을 다양한 서비스 user에

따른 가변적 호 수락 제어를 함으로써, 채널의 사용 효율을 극대화 할 수 있음을 보였다. 그러기 위해 가변채널할당과 고정채널할당에 따른 state diagram을 정의하였고, 실시간 서비스와 비실시간 서비스 두 가지 서비스가 존재하는 경우 stationary state 확률을 구했다. 그리고, 이를 n개의 다양한 서비스 클래스가 있는 상황으로 수학적인 일반화를 하였다. 그리고, 두 경우의 blocking 확률과 채널 utilization을 구했다.

### 2. system model

이 논문은 유-무선 채널에서 고속 데이터 통신에 적합한 방식으로 최근 활발히 연구되고 있는 OFDM(Orthogonal Frequency Division Multiplexing)방식을 사용한 Y-4G시스템을 대상으로 기술되었다. 시스템 구성은 채널의 총 용량을 20MHz로 두었고, OFDM symbol dutation은 4 μs이다. 총 subcarrier 수 52개이고, 이 중에서 데이터 서비스를 위한 subcarrier가 48개이다.



[그림2-1] system model