

# 다중 서비스 환경을 위한 효율적인 OFDM 부채널 배정 방법

정회원 이 중 훈\*, 송 상 섭\*\*

## Efficient subcarrier allocation for OFDM transmission in multi-services application

Jong-Hoon Lee\*, Sang-Seob Song\*\* *Regular Members*

### 요 약

차세대 이동 통신 기술은 무선 환경에서 고속 및 고품질의 데이터 전송을 요구하고 있다. OFDM 기술은 대역 폭당 전송 속도의 향상과 멀티패스 간섭 등의 방지를 위한 디지털 변조 방식으로 무선 환경에서 다양하게 응용되고 있다. 본 논문에서는 MBMS(Multimedia Broadcast and Multicast Service)와 같이 무선 환경에서 여러 개의 서비스를 동시에 제공하는 다중 서비스 환경에 OFDM을 적용할 때 필요한 부채널 배정에 관한 알고리즘을 제안하였다. 제안한 방법은 각 서비스의 채널 환경을 고려하여 부채널을 분배할 수 있을 뿐만 아니라 서비스의 우선 순위를 고려해서 우선순위가 높은 서비스에 더 많은 부채널 분배가 가능하고 기존의 부채널 배정 알고리즘과 비교할 때 수렴 속도의 향상과 계산량 감소의 장점이 있다.

**key Words** : subcarrier allocation; OFDM; multiservice;

### ABSTRACT

The next-generation mobile communication systems are required to provide high-data-rate and high quality transmission over hostile radio channels. Orthogonal Frequency Division Multiplexing (OFDM) technique regarded as a most promising solution in mobile environment. This paper considers the issue of efficient subcarrier allocation for OFDM system in multi-service application. Proposed algorithm can consider channel gain of each service and divide subchannel. Proposed method considers priority of service and divide more subchannels in high priority service. When proposed algorithm compares with existent algorithm, the convergence rate improves and the calculation amount is decreased.

### 1. 서론

OFDM(Orthogonal Frequency Division Multiplexing)은 무선랜(802.11a/b)이나 WiMax(802.16a), DVB(digital video broadcasting)등에 널리 사용되는 기술로서 기술멀티미디어 서비스에서 요구되는 고속의 데이터 전송을 위한 변조 기술로 각광을 받고 있다. 지금까지 위성에서는 주로 단일

반송파를 이용한 시스템이 적용되고 있지만, 향후 주파수 효율이 높고 대역폭 당 전송속도가 우수한 OFDM이 위성을 통한 서비스에도 적용될 것으로 기대된다. 현재 WiBro의 경우에는 OFDM을 적용한 시스템을 구현하고 있으며, 향후 MBMS(Multimedia Broadcast and Multicast Service)의 경우도 OFDM 시스템의 적용을 고려한 연구가 활발하게 진행되고 있다.

\* 전북대학교 전자정보공학부(jonghoon@chonbuk.ac.kr),

\*\* 전북대학교 전자정보공학부(ssong@chonbuk.ac.kr)

일반적으로 OFDM 시스템에서 부채널을 각 서비스에 배정하는 방법은 크게 두 가지로 나눌 수 있다. 시스템에 주어진 제한된 전력을 이용하여 전송률을 최대화 할 수 있는 방법<sup>[1]</sup>과 정해진 전송률을 유지하면서 사용하는 전력을 최소화 하는 방법<sup>[2]</sup>이 있으며, 이 두 가지 방법을 절충한 방법으로 주어진 전력을 이용하여 전체적인 전송률은 감소하지만 각 사용자에게 균등한 전송률을 제공할 수 있는 방법<sup>[3]</sup>과 같은 다양한 부채널 및 전력 할당 방법들이 존재한다.

본 논문에서는 MBMS와 같이 데이터, 영상, 음악 등 여러 서비스를 동시에 제공할 수 있는 환경에서 적용 가능한 OFDM의 부채널 배정 방법에 대해 연구하였다. 각 서비스의 채널 상태에 관한 정보를 이용하여 모든 서비스에 균등한 전송률을 제공하거나 특정 서비스에 높은 우선순위를 주어 보다 많은 전송률을 제공할 수도 있다. 사업자의 관점에서 보면 이와 같은 부채널 배정 방법을 이용할 경우, 높은 전송속도를 제공하는 서비스와 일반적인 전송속도를 제공하는 서비스를 차등화 하여 다양한 서비스를 제공할 수 있다.

전력과 부채널을 배정하는 과정에서 주어진 전력과 부채널의 채널 이득을 이용하여 최대 전송률을 찾기 위한 다양한 방법들이 연구되고 있다. 일반적으로 서비스를 제공 받는 주체가 단일한 경우에는 water-filling을 이용하여 최대 전송률을 구할 수 있지만, 여러 개의 서비스가 동시에 제공될 경우 전력의 분배와 부채널의 배정을 동시에 고려할 경우 계산량이 많아져서 실제 시스템에 적용하기에는 힘들다. 그래서 전력의 분배와 부채널의 배정을 나누어서 실행하는 것이 보통의 방법이다. 본 논문에서는 이 중에서 부채널의 배정하는 알고리즘에서 다루고 있다.

본 논문에서는 각 서비스에 기본적인 전송률을 보장하기 위해서 부채널 배정을 두 단계로 배정하는 방법을 이용하였다. 첫 번째 단계는 일정량의 부채널을 각 서비스에 일률적으로 배정하는 과정이고 다음 과정은 우선순위를 고려하여 보다 많은 전송률을 제공해야하는 서비스에 남은 부채널을 차례로 배정하는 단계이다. 우선순위가 동일할 경우 모든 서비스가 균등한 전송률을 얻을 수 있도록 부채널을 배정하게 된다. 본 논문에서는 [3]에서 제시한 방법보다 계산량을 줄일 수 있고, 수렴 속도를 개선할 수 있는 향상된 알고리즘을 제안 하였다. 논문의 나머지 부분은 다음과 같이 구성되었다. II장에서

기존의 부채널 배정 알고리즘을 설명하고 III장에서 제안한 알고리즘을 설명하고 기존의 알고리즘과 비교한다. IV장에서는 모의실험의 결과를 설명하고 V장에서 결론을 맺는다.

## II. 시스템 모델과 부채널 할당 방법

위성 또는 기지국을 통해서 OFDM을 이용하여 K가지 서비스가 N개의 부채널을 통해서 제공된다고 가정한다. 또 각 서비스의 채널 정보는 송신단에서 알 수 있고 부채널 배정의 결과는 별도의 제어 채널을 통해서 수신단에 전달된다고 가정한다. 각 부채널에 배정하는 전력은 전체 전송 대역폭 B를 부채널의 개수인 N으로 균등하게 나누어서 제공하는 것으로 가정한다. 그리고 throughput을 계산하기 위해서는 부채널의 대역폭과 채널이득을 곱한 값과 잡음 등을 고려해야 하지만 본 논문에서 사용한 throughput 값은 사용 대역폭에 따라 달라질 수 있는 상대적은 수치이다.

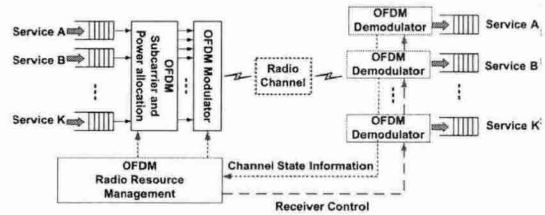


그림 1. 다중 서비스 환경 OFDM의 블록 다이어그램

OFDM에서 제한된 전력을 이용해서 각 서비스별 부채널을 배정하는 문제는 간단한 문제는 아니다. 다양한 방법 중에서 여러 개의 서비스가 동시에 제공될 때 각 서비스의 throughput의 최소값을 최대로 만드는 부채널 배정 문제<sup>[3]</sup>는 다음과 같이 정리할 수 있다.

$$\max_{P_{k,n}} \min_k \sum_{n \in S_k} \frac{B}{N} \log_2 \left( 1 + \frac{P_{k,n} h_{k,n}^2}{N_0 B} \right)$$

subject to

$$\sum_{n=1}^N \sum_{k=1}^K P_{k,n} \leq P_{max} \quad P_{k,n} \geq 0 \quad \text{for all } k, n$$

$S_1, S_2, \dots, S_K$  are disjoint  
and  $S_1 \cup S_2 \cup S_3 \dots \cup S_K \subset \{1, 2, \dots, K\}$

$P_{k,n}$  = 서비스 k, 부채널 n에 분배된 전력  
 $h_{k,n}$  = 서비스 k, 부채널 n의 채널 이득  
 $S_k$  = 서비스 k에 배정된 부채널의 집합  
 $N_0$  = AWGN

부채널과 전력의 분배 문제는 크게 3가지로 요약

될 수 있다. 첫 번째는 각 부채널을 서비스 별로 분배하는 것이고, 두 번째는 각 서비스들 사이의 전력을 어떻게 분배할 것인가 하는 문제이다. 그리고 마지막은 각 서비스에 분배된 부채널들 사이에 배정 받은 전력을 어떻게 분배하는가의 문제이다. 이 문제는 수학적 방법으로 최적값을 찾을 수 없고 모든 경우의 수에 대해서 계산해서 최대값을 찾아야 한다. 최적값을 찾기 위해서는 서비스의 종류를  $K$ 개, 부채널의 총 개수를  $N$ 개라고 한다면 총 계산량은  $K^N$ 이 된다. 예를 들어 4개의 서비스를 128개의 부채널을 사용하여 제공한다고 할 때 최적값을 찾기 위해서는  $K^N = 4^{128} = 1.16 \times 10^{77}$ 의 계산이 필요하다. 본 논문에서는 계산량을 줄이기 위해서 [3]에서의 유사하게 부채널 배정과 전력 분배를 분리하는 방법을 사용하였다.

### Ⅲ. 다중 서비스 환경에서 부채널 배정

MBMS와 같이 여러 종류의 서비스가 동시에 제공되는 환경에서 OFDM을 적용할 경우 각 서비스 별로 부채널과 전력을 배정하는 방법은 다양하다. 각 서비스는 제공하는 내용에 따라서 낮은 전송률로도 서비스 제공이 가능할 경우도 있고, 보다 높은 전송률이 필요한 서비스도 존재할 것이다. 따라서 사업자의 관점에서 다양한 서비스를 개발하고 요금에 따라서 차별화된 서비스를 제공할 수 있도록 하기 위해서는 각 서비스에 최소한의 전송률을 제공하고 우선순위가 높은 서비스에는 보다 많은 부채널을 분배해서 높은 전송률을 제공해 주어야 한다.

본 논문에서 제안한 알고리즘은 각 서비스에게 기본적인 전송률을 제공해 줄 수 있고, 특정 서비스에게 우선순위를 부여하여 더 높은 전송률을 제공할 수 있다. 아래에서 서비스의 종류를  $K$ 개, 부채널의 총 개수를  $N$ 개라고 가정한 경우 제안한 알고리즘을 정리하였다.

1. 초기화  
 $R_k = 0, \Omega_k = \{ \}$   
 $(k = 1, 2, \dots, K \text{ and } A = 1, 2, \dots, N)$
2. for  $k = 1$  to  $K$   
 $|H_{k,n}|$ 를 소트,  $L$ 개의 부채널 선택  
 $\Omega_k = \Omega_k \cup S_k,$   
 $A = A - \{S_k\},$   
 $R_k$ 를 갱신
3. while  $A \neq \{ \}$

```

find  $k, \frac{R_k}{\gamma_k} \leq \frac{R_i}{\gamma_i}$  for all  $i, 1 \leq i \leq K$ 
 $k$ 에 대해서,
if (서비스  $k$ 가 연속적으로  $p+i$ 회 선택)
 $|H_{k,n}|$ 를 소트한 후,
 $(\log_2 N + 1) \times 2^{(i-1)}$ 개의 부채널을 분배
 $k$ 와  $n$ 에 대해서,
 $\Omega_k, A, R_k$  갱신
else
find  $n, |H_{k,n}| \geq |H_{k,j}|$  for all  $j \in A$ 
 $k$ 와  $n$ 에 대해서,
 $\Omega_k, A, R_k$  갱신
end
    
```

여기서  $N$ 은 부채널의 총 개수,  $k$ 는 서비스를 의미하고 모두  $K$ 개의 서비스가 있음을 의미한다.  $R_k$ 는 서비스  $k$ 의 throughput(data rate)이고 식 (1)과 (2)에서 그 계산 방법을 나타내었다. 식 (2)에서 보면  $R_k$ 는 식 (1)과 같은 채널용량의 함수이다. 그리고 식 (1)은 대역폭과 서비스 개수, 부채널 개수 등 다른 조건이 모두 동일한 경우 결국에는 채널 이득,  $|h_{k,n}|$ 의 함수가 된다.  $\Omega_k$ 는 서비스  $k$ 가 점유하는 부채널의 집합을 나타낸다.  $\gamma_k$ 는 서비스  $k$ 의 가중치를 나타낸다.  $\gamma_k$  값을 적절하게 조정함으로써 특정 서비스 우선순위를 부여하여 더 많은 부채널을 배정할 수 있다. 세 번째 단계에서 부채널을 분배해 줄 서비스를 찾기 위해서 각 서비스의 throughput 값을 비교할 때  $\gamma_k$  값이 클수록  $R_k/\gamma_k$ 이 작아지게 된다. 결과적으로  $\gamma_k$  값이 우선순위 값이 된다.

$$C(h_{k,n}) = \frac{B}{N} \log_2 \left( 1 + \frac{\frac{P_{\max}}{N} \cdot h_{k,n}^2}{N_0 \cdot \frac{B}{N}} \right) \quad (1)$$

$$R_k = \sum_{n \in \Omega_k} C(h_{k,n}) \quad (2)$$

위 알고리즘을 살펴보면 크게 두 단계로 이루어져 있다. 첫 번째 단계에서 초기화가 끝나면 각 서비스 별로 채널 이득을 내림차순으로 정리하고 그 중 상위  $L$ 개의 부채널을 각 서비스에게 차례로 배정한다. 그리고  $\Omega_k$ 과  $R_k$ 를 갱신한다. 두 번째 단계에서 각 서비스의 전송률  $R_k$ 를 비교하여 가장 작은 서비스에게 남은 부채널을 차례로 배정한다. 이 과정을 모든 부채널일 모두 배정될 때까지 ( $A = \emptyset$ ) 반복한다.

기존의 알고리즘<sup>[3]</sup>에서는 첫 번째 부채널 배정에

서 각 서비스에게 가장 좋은 부채널 한 개씩을 배정 하였지만 본 논문에서는 그 숫자를  $L$ 개씩 선택 할 수 있도록 해서 기존 알고리즘 보다 반복 횟수를 줄였다. 그리고 남은 부채널도 한 개씩 배정하던 기존의 방법 대신 같은 서비스가 연속으로  $(p+i)$  회 이상 연속으로 부채널을 배정받을 경우에는 한 개가 아니라  $(\log_2 N + 1) \times 2^{(i-1)}$  개씩 배정하도록 하였다.  $(\log_2 N + 1)$ 를 선택하는 이유는 남아있는 부채널에서 채널이득이 가장 큰 부채널을 찾는 연산보다 남아 있는 부채널을 채널이득이 큰 순서로 정렬하는 연산량이 더 작아지기 때문이다.

다음 식 (3)은  $N$ 개의 부채널 중에서 채널 이득이 큰 부채널을 차례로  $s$ 개를 찾아내는데 필요한 연산량이다. 식 (4)는  $N$ 개의 부채널을 채널이득 순서로 정렬하는데 필요한 연산량이다.

$$O_{max} = (N-1) + (N-2) + \dots + (N-s) = sN - \frac{s(s+1)}{2} \quad (3)$$

$$O_{sort} = N \cdot \log_2 N \quad (4)$$

$$O_{max} \geq O_{sort} \quad (5)$$

식 (3)과 (4)를 비교해서 식 (3)의 값보다 식 (4)의 값이 작아지는 경우에 소트 연산을 사용하여 계산량을 줄일 수 있다. 다음 그림 2는 식 (3)과 식 (4)의 값의 차이를 비교한 그래프이다.  $s$ 는 식 (5)를 만족하는 최소 정수이다.

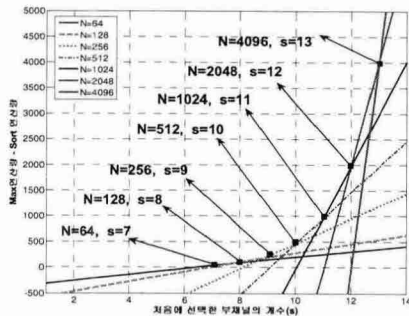


그림 2. Max 연산과 Sort 연산의 계산량 비교

부채널의 개수가  $N$ 일 때  $s$  값은 다음 식 (6)과 같이 표현된다. 식 (3), (4), (5)에서 사용된  $N$ 값은 아직 배정되지 않고 남아있는 부채널의 총 개수를 의미한다.

$$s = \log_2 N + 1 \quad (6)$$

따라서 알고리즘의 첫 번째 단계에서  $L$ 개를 선택

할 때  $L$ 값이 식 (6)의  $s$  값보다 작으면 max 연산을 이용하고, 크면 sort 연산을 이용하는 것이 알고리즘의 계산량을 줄이는데 유리하다.

또 두 번째 단계에서 부채널을 배정할 때 throughput이 작은 서비스에게 연속해서 부채널이 배정되는 경우가 발생할 수 있다. 특히 특정 서비스의 채널이득이 작을 경우에 이러한 경우가 자주 발생한다. 이때 알고리즘의 수렴 속도를 개선하기 위해서 기존 알고리즘에서는 부채널을 한 개씩 배정했지만, 제안한 알고리즘에서는 sort 연산을 이용하여  $(\log_2 N + 1)$ 개를 분배하도록 했다. 그리고 다음에 또 다시 같은 서비스에 부채널을 분배할 경우에는  $(\log_2 N + 1)$ 개의 2의 지수승으로 분배하여 알고리즘의 속도를 개선하였다. 이런 방법으로 부채널을 분배할 경우에 기존의 알고리즘과 비교했을 때와 throughput의 관점에서는 거의 차이가 없고, 알고리즘의 수렴속도는 개선되었고 계산량은 감소하였다.

본 논문에서는  $p$ 의 값을 모의실험을 통해서 관찰한 결과로 “4” 정도의 값이 적절하다고 판단하였다. 예를 들어 부채널의 총 개수가 1024개이고, 각 서비스가 처음 단계에서  $L$ 개씩 부채널을 배정을 받고, 다음 과정에서  $\gamma_k$ 를 고려한  $R_k$ 가 가장 작은 서비스를 찾아내고 그 서비스에 부채널을 한 개씩 할당한다. 그런데 서비스 A가 4번 연속으로 부채널을 배정받는 상황이 발생한 경우에는 5번째에는 부채널 한 개가 아니라 11개를 배정한다. 6번째도 역시 서비스 A가 부채널을 배정받는다면  $22(=11 \times 2)$ 개의 부채널을 배정한다. 이러한 과정의 결과로 특정 서비스의 채널 품질이 불량하거나 우선순위가 높아서 더 많은 부채널을 배정해야하는 경우에 알고리즘의 속도를 높일 수 있다.

#### IV. 모의실험 결과

##### 1. 각 서비스의 채널 환경이 유사한 경우

다음 그림은 각 서비스의 채널 환경이 유사한 경우에 대해서 채널 배정 결과를 나타낸 것이다. 서비스는 5가지이고 부채널의 수는 128개를 가정한 결과이다. 각 서비스의 우선순위는 모두 동일하다. 잡음의 영향은 고려하지 않았고 각 서비스의 채널상태는 임의의 Rayleigh 채널이고, 부채널의 대역폭은 4kHz로 가정하였다.

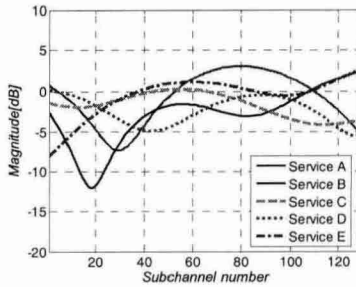


그림 3. 채널이득이 유사한 5개 서비스의 채널 상태

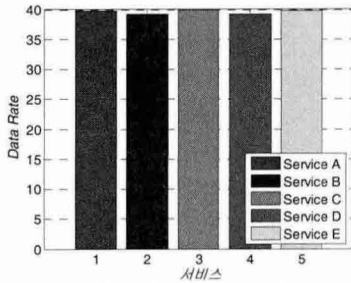


그림 4. 채널이득이 유사한 5개 서비스의 throughput

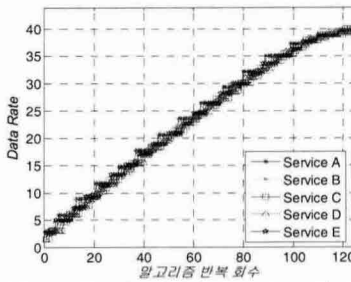


그림 5(a). 기존 알고리즘의 throughput 및 반복 회수

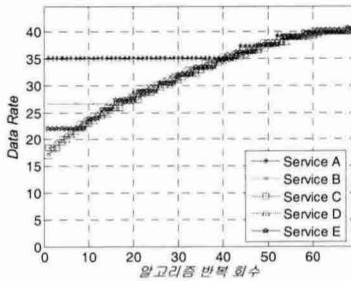


그림 5(b). 제안 알고리즘의 throughput 및 반복 회수 ( $L=12$ )

그림 3은 5가지 서비스의 채널 이득이고 그림 4는 각 서비스의 throughput을 나타낸다. 그림 5는 기존 알고리즘과 제안된 알고리즘에서  $L$ 값을 12로 했을 때 알고리즘 반복회수와 throughput의 변화량을 나타내고 있다. 최종적인 throughput은 약 40으

로 거의 변화가 없지만 알고리즘의 반복 회수가 124회에서 69회로 감소하였고, sort연산을 통해서 17%의 계산량이 감소하였다.

2. 특정 서비스의 채널 환경이 불량한 경우

다음 그림6은 특정 서비스의 채널 환경이 특히 불량한 경우의 채널환경의 예를 보여주고 있다. 그림 7a과 그림 7b는 이러한 채널 환경에서 기존 알고리즘과 제안한 알고리즘을 이용한 채널 배정 결과를 나타낸 것이다. 역시 5개의 서비스와 128개의 부채널을 가정하였다. 그리고 표 1에서는 특정 서비스의 채널 환경이 불량한 경우 기존 알고리즘과 제안한 알고리즘의 계산량 및 throughput을 비교 정리 하였다.

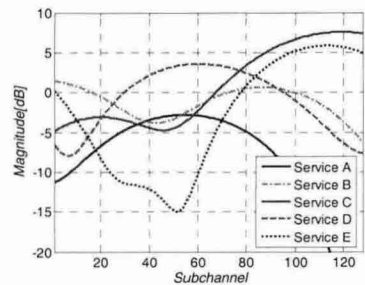


그림 6. 특정 서비스 채널 환경이 불량한 경우의 채널 상태

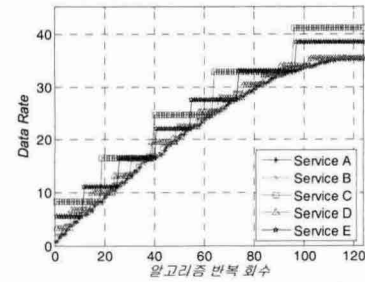


그림 7(a). 기존 알고리즘의 throughput 및 반복 회수 비교 (특정 서비스의 채널 환경이 불량한 경우)

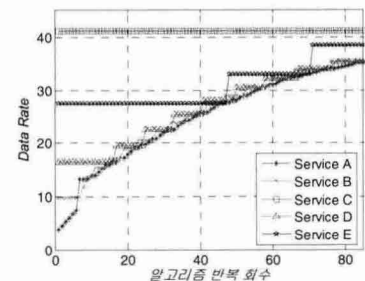


그림 7(b). 제안 알고리즘의 throughput 및 반복 회수 비교 (특정 서비스의 채널 환경이 불량한 경우,  $L=5$ )

표 1. 기존 알고리즘과 비교

서비스	기존 알고리즘		제안한 알고리즘			
			L=5		L=12	
	$\Omega_k$	$R_k$	$\Omega_k$	$R_k$	$\Omega_k$	$R_k$
A	83	35.3	83	35.3	73	33.8
B	20	35.5	20	35.5	19	33.9
C	5	41.2	5	41.2	12	98.3
D	13	35.4	13	35.4	12	38.2
E	7	38.6	7	38.6	12	63.2
Total	186.0		185.9		267.3	
반복회수	128		85		43	
계산량	sort 0+0회 max 128회 (8123)		sort 0+4회 max 80회 (8017)		sort 5+5회 max 38회 (6368)	

그림 6에서 보면 서비스 A의 채널 환경이 다른 서비스에 비해서 매우 불량한 것을 알 수 있다. 이 경우 서비스 A는 다른 서비스와 비슷한 throughput을 제공하기 위해서 128개의 부채널 중에서 80개 이상을 배정받는 것을 알 수 있다. 이 경우 서비스 A에 연속적으로 부채널을 배정하는 경우가 빈번하게 발생하게 된다. 따라서 제안된 알고리즘의 경우 L=5일 경우 서비스 A가 연속적으로 선택되는 부분에서 sort 연산이 4회 이루어지게 된다. 이는 계산량에는 큰 영향을 미치지 않지만 알고리즘의 반복회수를 128회에서 85회로 줄이는 이득을 얻을 수 있다. 또 L=12로 선택했을 경우에는 첫 번째 단계에서 최대값을 찾는 max 연산 대신에 sort 연산을 사용할 수 있다. 결과적으로 기존의 알고리즘에 비해서 각 서비스의 최소 throughput에는 큰 영향을 미치지 않으면서 계산량과 알고리즘 반복회수를 크게 줄일 수 있는 이득을 얻게 된다.

위 그림과 표의 결과에서 알 수 있듯이 L값의 선택이 매우 중요하다. 특히 부채널의 수가 증가함에 따라서 적절하게 선택된 L값은 전체 알고리즘의 계산량과 수렴 속도에 많은 영향을 미치게 된다. 다음에서 L값의 변화에 따른 알고리즘의 계산량에 대해서 알아본다.

### 4.3 L값의 변화에 따른 알고리즘 계산량

부채널의 수가 증가하면 각 서비스가 처음 단계에서 배정받는 부채널의 수(L)가 변함에 따라서 throughput과 알고리즘의 계산량이 크게 달라진다. 따라서 적절한 L값을 선택하는 것이 중요하다. 다음 그림 8은 서비스의 수는 5가지, 부채널의 수가 1024개일 경우의 채널 상태를 나타낸다. 그리고 그림 9에서는 L값에 따른 전체 throughput의 변화를 보여준다.

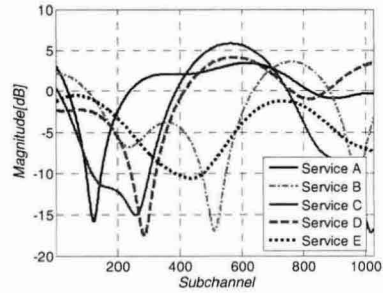


그림 8. 채널 상태(서비스 개수=5, 부채널=1024)

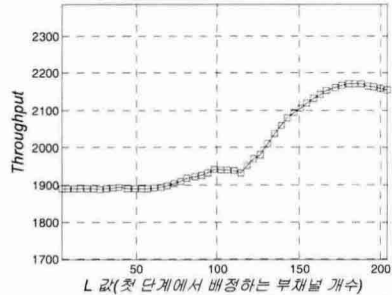


그림 9. L값에 변화에 따른 전체 throughput의 변화

그림 9에서는 첫 번째 단계에서 배정하는 부채널의 개수인 L값을 1에서부터 약 200까지 변화시키면서 각 서비스의 throughput의 변화를 나타낸다. 채널의 상태에 따라서 다소 차이가 있지만 L값이 증가하면 전체 throughput의 값이 증가하는 경향을 보인다. 일반적으로 주어진 전력을 가지고 최대 전송률을 얻기 위해서는 채널 상태가 좋은 서비스에게 더 많은 부채널을 배정하는 것이 유리하다. 이러한 이유로 L값이 증가함에 따라 채널 상태가 좋은 서비스 첫 번째 단계에서 부채널을 점점 더 많이 배정 받게 되어 전체적인 throughput이 증가하게 되는 것이다.

그러나 다음 그림 9와 표2에서 보면 L값이 어느 수준 이상으로 증가하게 되면 각 서비스 사이에서 throughput의 공평성을 상실하게 된다. 즉, L값을 너무 크게 설정하게 되면 각 서비스 사이에 전송률의 변화가 발생한다. 그림 10은 L값의 변화에 따른 각 서비스의 throughput의 변화를 나타낸 것이다. L값이 70전후까지는 각 서비스의 throughput이 380으로 비슷하지만 L값이 90을 초과하게 되면 서비스 C는 throughput이 증가하면서 다른 서비스들과 차이가 발생한다. 그리고 L값이 더 커지게 되면 각 서비스들의 공평성의 균형이 무너지게 된다.

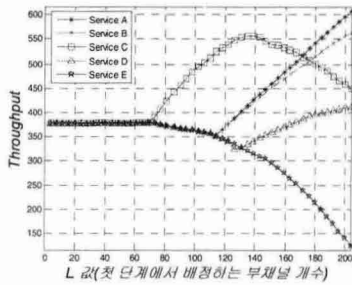


그림 10. L값에 변화에 따른 각 서비스의 throughput의 변화

표 2. L값의 변화에 따른 계산량 비교(N=1024, K=5)

서비스	L=1		L=30		L=70		L=100	
	$\Omega_k$	$R_k$	$\Omega_k$	$R_k$	$\Omega_k$	$R_k$	$\Omega_k$	$R_k$
A	135	378	136	380	137	380	123	365
B	197	378	198	378	188	380	186	364
C	90	378	90	378	72	380	100	489
D	121	378	121	378	140	380	141	364
E	481	378	479	378	487	380	474	363
Total	1,891		1,891		1,900		1,945	
반복	1,024		802		529		210	
계산량	sort 0+0회 max 1024회 (523,776) 기준과동일	sort 5+3회 max 797회 (393,752) 25% 감소	sort 5+4회 max 524회 (212,908) 60% 감소	sort 5+6회 max 205회 (96,044) 81% 감소				

표 2에서는 1024개의 부채널, 5가지 서비스의 경우에 L값이 1, 30, 70, 100일 때 알고리즘의 반복 회수 및 계산량, throughput 등을 정리하였다. L값이 커질수록 알고리즘의 반복 횟수는 줄어들고, 계산량도 감소하는 것을 알 수 있다. 따라서 각 서비스의 공평성을 유지하는 범위 내에서 적절한 L값을 선택할 필요가 있다. 다음은 본 논문에서 L값을 결정하는 방법이다.

$$DR_l = \max(R_{k,l}) - \min(R_{k,l}) \text{ for } k = 1, 2, \dots, K$$

$$AR_l = \left( \sum_{k=1}^K R_{k,l} \right) / K \text{ for } k = 1, 2, \dots, K$$

find  $DR_l > (AR_l * L_{factor})$  then  $L = l$

$R_{k,l}$ 는 처음 단계에서 l개의 부채널을 배정받은 서비스 k의 전송률을 의미한다. 이때에 전송률이 가장 높은 서비스와 가장 낮은 서비스의 전송률의 차이를  $DR_l$ 로 정의하였다. 즉,  $DR_l$ 은  $R_{k,l}$ 의 최대-최소의 간격을 의미한다.  $AR_l$ 은 모든  $R_{k,l}$ 의 평균 값이다.  $DR_l$ 과  $AR_l$ 은 우선순위가 주어졌을 경우에는 우선순위 값으로 나누어진 정규화 된 값을 사용한다.  $L_{factor}$ 는 각 서비스의 전송률의 공평성에 대

한 척도이다. 즉  $L_{factor}$ 가 0.1일 경우 각 서비스의 전송률이 10%까지 차이가 나는 것을 허용하는 범위에서 부채널을 분배한다는 것을 의미한다. 위 그림 10의 경우에는  $L_{factor}$ 가 0.1일 경우 L값은 약 80이다. 즉, 각 서비스가 처음 단계에서 80개씩의 부채널을 배정받게 되면 총 1024개의 부채널 중 405개(=81\*5), 약 40%를 첫 번째 단계에서 배정받게 되어 [3]의 방법과 비슷한 성능을 유지하면서 계산량이 65% 정도 감소하게 된다.

다음 그림 11은 서비스의 종류가 5가지이고 부채널의 개수가 1024개인 Rayleigh 특성 채널을 임의로 30가지를 생성하여 그 표준편차를 그래프로 나타냈다.

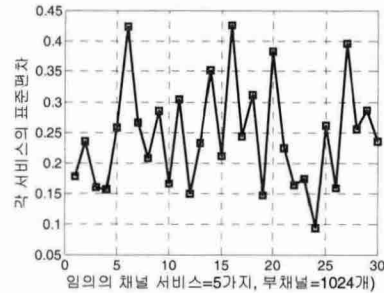


그림 11. 각 서비스의 표준편차(서비스=5, 부채널=1024)

그림 12의 실선 그래프는 채널 이득이 다양한 경우에 각 서비스가 선택할 수 있는 L값을 보여주고 있다.  $L_{factor}$ 가 클수록 더 높은 L값을 선택할 수 있고 L값이 클수록 알고리즘의 계산량은 줄어든다. 그러나 L값이 너무 크게 설정되면 각 서비스 사이의 throughput에 대한 공평성을 상실하게 된다. 그러나 그림 12의 실선으로 그려진 L값은 주어진 채널정보를 가지고 수학적인 계산을 통해서 얻은 값이다. 이 과정에는 많은 계산량과 시간이 필요하므로 실제로 적용하기에는 부적합하다. 따라서 사전에 적절한 L값을 선택하기 위한 방법이 필요하다.

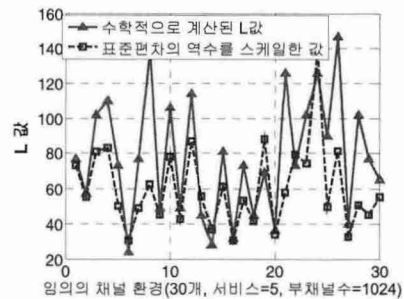


그림 12. 계산된 L값과 표준편차를 이용하여 선택한 L값

본 논문에서는 대략적인  $L$ 값을 사전에 선택하기 위해서 비교적 계산량이 적은 표준편차를 이용하였다. 그림 12의 점선으로 그려진 그래프는 표준편차의 역수를 적절하게 스케일 하여 선택한  $L$ 값을 나타내고 있다( $L_{factor}=0.1$ ). 일반적으로 각 서비스의 채널 이득의 표준편차가 작을수록 더 큰  $L$ 값을 선택할 수 있으므로 표준편차의 역수를 적절하게 스케일링한 값을  $L$ 값으로 사용할 수 있다.

### V. 결 론

본 논문에서는 MBMS와 같이 여러 가지 서비스를 동시에 제공할 경우 OFDM 시스템에서 한정된 부채널을 각 서비스에게 효율적으로 할당하는 방법을 제안하였다. 각 서비스의 전송률을 비슷하게 유지하거나, 특정 서비스에게 우선순위를 부여하여 더 많은 전송률을 얻을 수 있도록 부채널을 분배할 수 있는 알고리즘을 제안하였고 그 성능을 분석하였다.

기존의 알고리즘은 각 서비스에게 최초로 부채널을 한 개씩 배정하고 이후 throughput이 가장 작은 서비스에 부채널을 하나씩 분배하는데 비해서 제안한 알고리즘은 처음 단계에서 각 서비스에게 할당하는 부채널의 수( $L$ )를 적절하게 선택함으로써 알고리즘의 계산량과 반복 횟수를 줄일 수 있다. 그리고 특성 서비스의 채널이 아주 불량하거나 또는 특정 서비스에게 높은 우선순위를 주어서 더 많은 부채널을 할당해야 하는 경우에 동일한 서비스가 연속해서 부채널을 배정받는 경우가 빈번하게 발생하게 되는데, 이 경우 연속해서 같은 서비스가 부채널을 배정받는 경우 선택하는 부채널의 개수를 1개씩이 아니라  $(\log_2 N + 1)$ 개의 2의 지수 승으로 선택하여 계산량을 줄일 수 있었다.

차세대 무선 이동 통신에서는 고속의 데이터를 요구하는 서비스들이 다양하게 제공될 것으로 예상되며 이에 따라 OFDM의 활용도가 높아지고 있다. 본 연구에서 제안된 방법들이 적용되어 다양한 서비스를 효율적으로 제공하는데 기여할 수 있을 것으로 기대된다.

### 참 고 문 헌

[1] J.Jang and K.B Lee, "Transmit power adaptation for mulituser OFDM system,"

IEEE J. Select Area Commun., vol. 21, pp. 171-178, Feb. 2003

- [2] D. Kivanc, G. Li, and H. Kiu, "Computationally efficient bandwidth allocation and power control for OFDMA," IEEE Trans. Wireless Commun. vol. 2 pp. 1150-1158. Nov. 2003
- [3] W. Rhee and J. M. Cioffi, "Increasing in Capacity of Multiuser OFDM System Using Dynamic Subchannel Allocation," in Proc. IEEE International Vehicular Technology Conference, vol. 2, pp. 1085-1089, May 2000
- [4] Zhenping Hu, Guangxi Zhu, Yuan Xia, "Multiuser subcarrier and bit allocation for MIMO-OFDM systems with perfect and partial channel information," Wireless Communications and Networking Conference, 2004.
- [5] Li Zhen, Zhu Geqing, Wang Weihua, Song Junde, "Improved Algorithm of Multiuser Dynamic Subcarrier Allocation in OFDM System," Communication Technology Proceedings, ICCT 2003. April 2003.

### 저 자

이 종 훈(Jong-Hoon Lee)

정회원



1996년 2월 : 전북대학교  
전자공학과 졸업  
1998년 2월 : 전북대학교  
전자공학과 석사  
1999년 3월 ~ 현재 : 전북대학교  
전자공학과 박사과정

<관심분야> 고속 디지털 통신, OFDM

송 상 섭(Sang-Seob Song)

정회원



1980년 2월 : KAIST 전기 및  
전자공학과 석사  
1990년 8월 : University of Manitoba  
공학박사  
1981년 3월 ~ 현재 : 전북대학교  
전자정보공학부 교수

<관심분야> 채널코딩, 초고속 이동통신, 무선 랜