

# MF-TDMA 광대역 위성시스템을 위한 최적 타임슬롯 할당 체계 Optimal Timeslot Allocation for MF-TDMA Broadband Satellite Systems

장근녕<sup>1</sup>, 이기동<sup>2</sup>, 이호진<sup>2</sup>

<sup>1</sup> 신라대학교 경영학부

<sup>2</sup> 한국전자통신연구원 무선방송연구소

## Abstract

본 논문에서는 MF-TDMA(Multi Frequency - Time Division Multiple Access) 체계를 이용하는 광대역 위성 시스템의 리턴 링크를 대상으로 최적 타임슬롯 할당 체계를 제시한다. 이를 위해 먼저 clear-sky RCST와 rain-fade RCST가 존재하고 복수의 서비스 및 지연 클래스가 존재하는 광대역 위성 시스템을 대상으로 MF-TDMA 체계 하에서 최적 타임슬롯 할당 모형을 정형화한다. 다음으로 3단계로 구성된 최적 타임슬롯 할당 체계를 제시한다. Phase 1에서는 clear-sky TRF 타임슬롯들의 집합과 rain-fade TRF 타임슬롯들의 집합을 결정하고, Phase 2에서는 각 RCST의 각 서비스/지연 클래스에 할당되는 타임슬롯의 수를 결정하고, Phase 3에서는 각 RCST의 각 서비스/지연 클래스에 할당된 타임슬롯들의 스케줄을 결정한다. 마지막으로 불균등 수요 데이터를 활용하여 제시한 최적 타임슬롯 할당 체계의 성능을 분석한다. 성능 분석 결과 제시한 최적 타임슬롯 할당 체계는 짧은 시간 내에 최적해 또는 최적해와 아주 가까운 해를 안정적으로 구한다는 사실을 확인할 수 있었다.

## 1. 서론

최근 정지궤도위성을 이용한 위성통신서비스 분야는 급격한 편화를 맞이하고 있다. 전 세계적으로 가장 많이 사용되던 C 및 Ku 대역의 주파수가 고갈되어 위성의 사용 주파수가 C 및 Ku 대역에 비하여 여러 가지 장점이 있는 Ka 대역으로 이동하고 있다. 또한, 최종사용자는 단순한 데이터서비스나 음성서비스를 넘어 인터넷 등 멀티미디어 서비스를 요구하고 있다.

이에 따라 미국, 유럽, 일본 등 위성기술 관련 선진국에서는 Ka 대역을 활용하여 멀티미디어 서비스를 제공하는 광대역 위성시스템 개발을 본격적으로 추진하고 있다[3]. 이들 위성 선진국에서는 자체 보유한 Ka 대역 위성을 이용하여 각종 실험 및 시험서비스를 실시하여 Ka 대역 위성시스템 및 지구국 개발이 성숙 단계에 있으며 이러한 기술을 이용하여 상용 서비스 개발에 박차를 가하고 있다.

이러한 광대역 위성시스템의 효율적인 구현을 위해서는 해결해야 할 여러 가지 과제가 남아 있는데, 그 중에 가장 중요한 부분 중의 하나가 위성 액세스 자원의 효율적인 활용을 위한 다중접속 체계의 개발이다. 현재 개발중인 다중접속 프로토콜에는 MF-CDMA (Multi-Frequency Code Division Multiple Access) 방식, MF-TDMA (Multi-Frequency Time Division Multiple Access)

방식 등이 있다.

광대역 위성시스템 개발 분야에서 널리 활용되고 있는 ETSI의 DVB-RCS(Digital Video Broadcasting - Return Channel via Satellite) 표준에서는 MF-TDMA 방식을 사용하고 있다[1]. 따라서 이 방식 하에서 위성 액세스 자원을 가장 효율적으로 활용할 수 있는 방안을 찾는 것이 중요하다고 할 수 있고, 이에 대한 연구가 전 세계적으로 이루어지고 있다[2,4-8].

하지만 아직까지는 광대역 위성시스템에서 요구되는 많은 사항을 제대로 고려하는 방안이 개발되지 못하고 있다. 즉, 복수 클래스의 멀티미디어 서비스가 존재하고, 이들 각 클래스의 특성과 QoS 요구사항을 제대로 반영하면서 자원을 최적으로 활용할 수 있는 방안은 제시되지 못하고 있다.

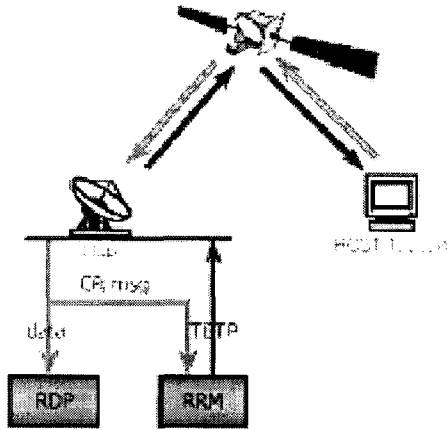
본 논문에서는 먼저 clear-sky RCST와 rain-fade RCST가 존재하고 복수의 서비스 및 지연 클래스가 존재하는 광대역 위성 시스템을 대상으로 MF-TDMA 체계 하에서 최적 타임슬롯 할당 모형을 정형화한다. 다음으로 3단계로 구성된 최적 타임슬롯 할당 체계를 제시한다. Phase 1에서는 clear-sky TRF 타임슬롯들의 집합과 rain-fade TRF 타임슬롯들의 집합을 결정하고, Phase 2에서는 각 RCST의 각 서비스/지연 클래스에 할당되는 타임슬롯의 수를 결정하고, Phase 3에서는 각 RCST의 각 서비스/지연 클래스에 할당된 타임슬롯들의 스케줄을 결정한다. 마지막으로 제시한 최적 타임슬롯 할당 체계의 성능을 분석한다.

본 논문은 다음과 같이 구성되어 있다. 제2절에서는 광대역 위성 시스템 모형을 설명한다. 제3절에서는 clear-sky RCST와 rain-fade RCST가 존재하고 복수의 서비스 및 지연 클래스가 존재하는 광대역 위성 시스템을 대상으로 MF-TDMA 체계 하에서 최적 타임슬롯 할당 모형을 정형화한다. 제4절에서는 3단계로 구성된 최적 타임슬롯 할당 체계를 제시하고, 제5절에서는 테스트를 통해 제시된 할당 체계의 성능을 분석한다.

## 2. 시스템 모형

### 2.1 리턴 링크 모형

본 연구에서는 [그림 1]과 같이 하나의 지구국(earth station; HUB), 하나의 GEO 위성, RCST라 불리는 복수의 단말로 구성된 대화형 광대역 위성 액세스 네트워크를 대상으로 연구를 수행한다. RCST는 clear-sky RCST(clear-sky 지역에 있는 RCST)와 rain-fade RCST(rain-fade 지역에 있는 RCST)로 구분되어진다.



[그림 1] 시스템 모형

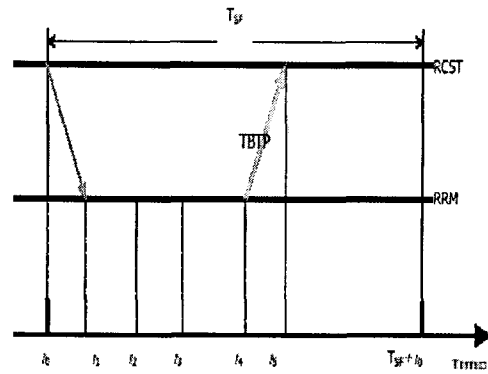
리턴 링크(return link; 위성을 거쳐 RCST에서 HUB로의 링크)에서의 다중접속체(multiple access scheme)는 MF TDMA에 근거를 두고 있다. 리턴 링크에 할당되는 라디오 자원은 다수의 RCST에 의해 공유된다. RCST는 RRM(realtime resource manager)에 CR(capacity request) 메시지를 전송하고, 이 메시지를 받은 RRM은 TBTP(terminal burst time plan)를 생성하여 RCST에 전송한다. TBTP를 받은 RCST는 이를 통해 자신에게 어떤 타임슬롯이 할당되었는지를 파악하게 된다. 이러한 프로시저는 매 슈퍼프레임(superframe)마다 실시된다.

RCST는 필요할 때마다 RRM에 CR 메시지를 전송한다. RRM은 RCST가 슈퍼프레임 단위 시간  $[t_0 - T_{sf}, t_0]$  동안 요청한 수요를 추적한다([그림 2] 참조). RRM은 RCST가 요청한 수요를 이용하여  $t_2$  시점까지 최적 할당량을 결정하고,  $t_3$  시점까지 TBTP 테이블을 생성한다. RCST는 생성된 TBTP를  $t_5$  시점에 받게 되고, 이 테이블에 따라 자신에게 할당된 타임슬롯을 기다리게 된다. 시간 간격  $[t_5, T_{sf} + t_0]$  가 너무 짧으면, RCST는 스케줄을 제대로 읽을 수 없게 되고, 따라서 이 스케줄을 다음 슈퍼프레임에서 사용할 수 없게 된다. RCST에게 수신한 TBTP 테이블을 읽을 수 있는 충분한 시간을 제공하기 위해서는 타임슬롯 할당 시간을 최대한 줄일 필요가 있다.

### 2.2 슈퍼프레임의 구조

슈퍼프레임의 구조는 시스템에 따라 다양하게 정의될 수 있지만, 본 논문에서는 아래와 같이 정의하고 논문을 전개한다. 물론 본 논문에서 제시되는 내용은 슈퍼프레임의 구조에 관계없이 쉽게 적용될 수 있다.

하나의 슈퍼프레임은 시간 주파수 도메인(time frequency domain)에서 특정한 시간 주파수 블록(time frequency block)  $T_{sf} \times W_{sf}$  ( $T_{sf} = 3.047760 \text{sec}$   $W_{sf} = 22.4 \text{MHz}$ ) 으로 정의되



[그림 2] Time Diagram

고, 이 시간 주파수 블록(time frequency block)  $T_{sf} \times W_{sf}$  는 4개의 시간 주파수 블록  $T_{sf} \times W_f$  ( $W_f = 5.6 \text{MHz}$ ) 로 나누어진다. 블록  $T_{sf} \times W_f$  는 clear sky 프레임(clear sky 타임슬롯) 또는 rain fade 프레임(rain fade 타임슬롯)을 생성하기 위해 사용된다.

블록  $T_{sf} \times W_f$  가 clear sky 프레임을 생성하기 위해 사용되는 경우, 특정한 시간 주파수 블록  $T_{sf}^c \times W_f$  ( $T_{sf}^c = 190.485 \text{ms}$ ) 으로 정의되는 16개의 clear sky 프레임으로 나누어진다. 이 clear sky 프레임은 하나의 CSC(common signaling channel) 타임슬롯, 하나의 ACQ(acquisition) 타임슬롯, 두 개의 SYNC(synchronization) 타임슬롯, 1940개의 TRF(traffic) 타임슬롯으로 구성되어진다.

블록  $T_{sf} \times W_f$  가 rain fade 프레임을 생성하기 위해 사용되는 경우, 특정한 시간 주파수 블록  $T_{sf}^r \times W_f$  ( $T_{sf}^r = 304.776 \text{ms}$ ) 으로 정의되는 10개의 rain fade 프레임으로 나누어진다. 이 rain fade 프레임은 8개의 시간 주파수 블록  $T_{sf}^r \times W_f'$  ( $W_f' = 0.7 \text{MHz}$ ) 으로 나누어지고, 각각의 블록은 하나의 CSC(common signaling channel) 타임슬롯, 하나의 ACQ(acquisition) 타임슬롯, 두 개의 SYNC(synchronization) 타임슬롯, 248개의 TRF(traffic) 타임슬롯으로 구성되어진다.

### 3. 최적 타임슬롯 할당 모형

여기에서는 clear sky 타임슬롯들의 집합과 rain fade 타임슬롯들의 집합이 주어졌다는 가정 하에서, 최적 타임슬롯 할당문제(timeslot assignment problem: TAP)를 수리적으로 정형화한다. 수리모형 정형화에서는 여러 가지 형태의 클래스(RCST 클래스, 서비스 클래스(service class), 지연 클래스(delay class))를 고려한다. RCST는 서로 다른 전송 모드를 사용하는 clear sky RCST와 rain fade RCST로 구분되어진다. 서비스 클래스는 e mail, 채팅, FTP 등과 같은 서비스 형태별로

나누어진다. 지연 클래스는 요청된 타임슬롯의 지연 정도에 따라 나누어진다.

타임슬롯 할당 문제를 정형화하기 위해 먼저 다음과 같은 매개변수를 정의한다.

- $R$  : 활성(active) RCST의 집합 ( $R = R_c \cup R_r$ )
- $R_c$  : 활성 clear-sky RCST의 집합
- $R_r$  : 활성 rain-fade RCST의 집합
- $C_1$  : 서비스 클래스(service class)의 집합
- $C_2$  : 지연(delay class)의 집합
- $S$  : TRF 타임슬롯의 집합 ( $S = S_c \cup S_r$ )
- $S_c$  : clear-sky TRF 타임슬롯의 집합
- $S_r$  : rain-fade TRF 타임슬롯의 집합
- $N_c$  : 한 개의 clear-sky 타임-주파수 블록  $T_{sp} \times W_f$  에 의해 생성되어지는 clear-sky TRF 타임슬롯의 수
- $N_r$  : 한 개의 rain-fade 타임-주파수 블록  $T_{sp} \times W_f$  에 의해 생성되어지는 rain-fade TRF 타임슬롯의 수
- $\overline{N_c}$  : clear-sky TRF 타임슬롯의 총 수
- $\overline{N_r}$  : rain-fade TRF 타임슬롯의 총 수
- $F_c^t$  : 각 clear-sky 타임-주파수 블록  $T_{sp} \times W_f$  에 의해 생성되어지는  $t$  번째 clear-sky TRF 타임슬롯들의 집합 ( $t = 1, \dots, N_c$ )
- $F_r^t$  : 각 rain-fade 타임-주파수 블록  $T_{sp} \times W_f$  에 의해 생성되어지는  $t$  번째 rain-fade TRF 타임슬롯들의 집합 ( $t = 1, \dots, N_r$ )
- $d_j^{kl}$  : RCST  $j$  의 서비스/지연 클래스 ( $k, l$ ) 에 의해 요청된 TRF 타임슬롯의 수 ( $k \in C_1, l \in C_2$ )
- $w_j^{kl}$  : RCST  $j$  의 서비스/지연 클래스 ( $k, l$ ) 에 대한 가중치
- $Q_j$  : RCST  $j$  의 서비스/지연 클래스 ( $k, l$ ) 에 할당될 수 있는 최대 타임슬롯 수 ( $Q_j \leq \min\{N_c, N_r\}$ )
- $a_j^{kl}$  : 요청된 수요량에 대한 최소 타임슬롯 할당 수의 비율
- $m_j$  : RCST  $j$  에 할당되어야 하는 최소 타임슬롯 수

TAP의 목적은 다음과 같이 정의되는 가중 패널티(weighted penalty)의 합을 최소화하는 것이다.

$$\sum_{j \in R} \sum_{k \in C_1} \sum_{l \in C_2} w_j^{kl} \cdot \max(0, d_j^{kl} - \sum_{i \in S} x_{ij}^{kl})$$

여기서  $x_{ij}^{kl}$  은 타임슬롯 할당 여부를 나타내는 이진 정수(binary integer) 변수로,  $x_{ij}^{kl}$  가 1이면 TRF 타임슬롯  $i$  가 RCST  $j$  의 서비스/지연 클래스 ( $k, l$ ) 에 할당되었다는 것을 의미하고  $x_{ij}^{kl}$

가 0이면 그렇지 않다는 것을 의미한다.  $\sum_{i \in S} x_{ij}^{kl}$  는 RCST  $j$  의 서비스/지연 클래스 ( $k, l$ ) 에 할당된 타임슬롯의 총 수를 나타내고, 따라서 ( $d_j^{kl} - \sum_{i \in S} x_{ij}^{kl}$ ) 는 RCST  $j$  의 서비스/지연 클래스 ( $k, l$ ) 에서 타임슬롯 할당으로 충족되지 않은 타임슬롯의 수를 나타낸다.

가중치  $w_j^{kl}$  는 각 클래스별 GOS(grade of service)를 반영한다. 이 값은 평균 대기 시간(average waiting time) 등의 여러 가지 요인에 의해 결정된다. 일반적으로 rain-fade RCST의 GOS가 clear-sky RCST의 GOS보다 더 중요하기 때문에, rain-fade RCST의 가중치가 clear-sky RCST의 가중치보다 크다. 본 연구에서는 rain-fade RCST에 절대적인 우선권을 주어, rain-fade RCST의 가중치가 서비스/지연 클래스에 관계없이 clear-sky RCST의 가중치보다 월등히 큰 것으로 가정한다. 즉, rain-fade RCST에 우선적으로 타임슬롯이 할당된다.

이제, TAP는 다음과 같은 수리모형으로 정형화되어진다.

(TAP)

$$\text{Min} \sum_{j \in R} \sum_{k \in C_1} \sum_{l \in C_2} w_j^{kl} (d_j^{kl} - \sum_{i \in S} x_{ij}^{kl}) \quad (1)$$

$$\text{s.t.} \sum_{i \in S} x_{ij}^{kl} \leq d_j^{kl}, \quad \forall j \in R, k \in C_1, l \in C_2 \quad (2)$$

$$\sum_{i \in S} \sum_{k \in C_1} \sum_{l \in C_2} x_{ij}^{kl} \leq 1, \quad \forall i \in S \quad (3)$$

$$\sum_{i \in S} \sum_{k \in C_1} \sum_{l \in C_2} x_{ij}^{kl} \leq Q_j, \quad \forall j \in R \quad (4)$$

$$z_j \sum_{i \in S_c} \sum_{k \in C_1} \sum_{l \in C_2} x_{ij}^{kl} + (1 - z_j) \sum_{i \in S_r} \sum_{k \in C_1} \sum_{l \in C_2} x_{ij}^{kl} = 0, \quad \forall j \in R \quad (5)$$

$$\sum_{i \in F_c^t} \sum_{k \in C_1} \sum_{l \in C_2} x_{ij}^{kl} \leq 1, \quad \forall j \in R, t = 1, \dots, N_c \quad (6)$$

$$\sum_{i \in F_r^t} \sum_{k \in C_1} \sum_{l \in C_2} x_{ij}^{kl} \leq 1, \quad \forall j \in R, t = 1, \dots, N_r \quad (7)$$

$$\sum_{i \in S} x_{ij}^{kl} \geq a_j^{kl} d_j^{kl}, \quad \forall j \in R, k \in C_1, l \in C_2 \quad (8)$$

$$\sum_{i \in S} \sum_{k \in C_1} \sum_{l \in C_2} x_{ij}^{kl} \geq m_j, \quad \forall j \in R \quad (9)$$

$$z_j = 1, \quad \forall j \in R_r \quad (10)$$

$$z_j = 0 \text{ or } 1, \quad \forall j \in R_c \quad (11)$$

$$x_{ij}^{kl} = 0 \text{ or } 1, \quad \forall i \in S, j \in R, k \in C_1, l \in C_2 \quad (12)$$

여기서  $z_j$  는 이진 정수 변수로,  $z_j$  가 1이면 RCST  $j$  에 rain-fade 타임슬롯이 할당된다는 것을 나타내고,  $z_j$  가 0이면 RCST  $j$  에 clear-sky 타임슬롯이 할당된다는 것을 나타낸다.

제약식 (2)는 RCST  $j$  의 서비스/지연 클래스 ( $k, l$ ) 에 할당되는 타임슬롯의 수는 수요량  $d_j^{kl}$  보다 많아서는 안 된다는 것이다. 제약식 (3)은 각 타임슬롯은 두 개 이상의 클래스 ( $j, k, l$ ) 에 할당

되어서는 안 된다는 것이다. 제약식 (4)는 RCST  $j$ 의 서비스/지연 클래스  $(k, l)$ 에 할당되는 타임슬롯의 수는 최대용량  $Q_j$ 보다 많아서는 안 된다는 것이다. 제약식 (5)는 하나의 RCST가 clear-sky 타임슬롯과 rain-fade 타임슬롯을 같이 사용해서는 안 된다는 것이다. 제약식 (6)과 (7)은 하나의 RCST에 동일한 타임 도메인(time domain)을 사용하는 타임슬롯이 두 개 이상 할당되어서는 안 된다는 것이다. 제약식 (8)은 RCST  $j$ 의 서비스/지연 클래스  $(k, l)$ 에 할당되는 타임슬롯의 수는 최소 요구량  $a_j^{kl}, d_j^{kl}$  이상이어야 한다는 것이다. 클래스  $(j, k, l)$  별로 수요(demand) 대 할당량(allocation amount)을 어느 정도 이상의 수준으로 유지하기 위하여  $a_j^{kl}$  값을 도입하는 것이 시스템 운용에 필요하다. 임계값  $a_j^{kl}$ 가 0.5라면, RCST  $j$ 의 서비스/지연 클래스  $(k, l)$ 에 그 수요인  $d_j^{kl}$ 의 50%에 해당되는 타임슬롯은 최소한 할당되어야 한다는 것을 의미한다. 이 값은 서비스 제공업자의 정책에 따라 정해진다. 제약식 (9)는 RCST  $j$ 에 할당되는 타임슬롯의 수는 최소 요구량  $m_j$  이상이어야 한다는 것이다. RCST가 추가적인 자원 요청을 할 때, CR 메시지 등을 원활하게 송신하도록 하기 위해서는 클래스에 관계없이 무조건 할당되어야 하는 최소량이 필요하다. 제약식 (10)은 rain-fade RCST에는 rain-fade 타임슬롯만을 할당해야 한다는 것이다.

TAP는 비선형 이진정수계획문제(nonlinear binary integer programming problem)로, NP-complete이다. 따라서 이 문제를 직접 푸는 것은 매우 어렵다. TAP를 보다 쉽게 풀기 위해, 다음과 같이 RCST  $j$ 의 서비스/지연 클래스  $(k, l)$ 에 할당되는 타임슬롯의 수를 결정하는 문제로 단순화한다.

(RTAP)

$$\text{Min} \sum_{j \in R} \sum_{k \in C_1} \sum_{l \in C_2} w_j^{kl} (d_j^{kl} - y_j^{kl}) \quad (13)$$

$$\text{s.t. } y_j^{kl} \leq d_j^{kl}, \quad j \in R, k \in C_1, l \in C_2 \quad (14)$$

$$\sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl} \leq Q_j, \quad j \in R \quad (15)$$

$$\sum_{k \in C_1} (1 - z_j) \sum_{l \in C_2} y_j^{kl} \leq \bar{N}_c \quad (16)$$

$$\sum_{k \in C_1} z_j \sum_{l \in C_2} y_j^{kl} \leq \bar{N}_r \quad (17)$$

$$y_j^{kl} \geq a_j^{kl} d_j^{kl}, \quad j \in R, k \in C_1, l \in C_2 \quad (18)$$

$$\sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl} \geq m_j, \quad j \in R \quad (19)$$

$$z_j = 1, \quad j \in R_r \quad (20)$$

$$z_j = 0 \text{ or } 1, \quad j \in R_c \quad (21)$$

$$y_j^{kl}: \text{nonnegative integer.} \quad (22)$$

여기서  $y_j^{kl}$ 은 RCST  $j$ 의 서비스/지연 클래스

$(k, l)$ 에 할당되는 타임슬롯의 수를 나타내는 변수이다. RCST에 할당되는 타임슬롯의 수가 결정되면 TAP의 제약식을 만족하는 타임슬롯 스케줄링을 쉽게 구할 수 있기 때문에, RTAP와 TAP는 동등한 문제(equivalent problem)이다. 그런데 RTAP 역시 비선형 정수계획문제로, NP-complete이다. 따라서 이 문제 역시 직접 다루는 것은 어렵다.

RTAP의 경우  $z_j$  변수로 인해 비선형 문제로 정형화되어 있다. 따라서 이 변수를 제외할 수 있다면 보다 쉽게 문제를 풀 수 있을 것으로 사료된다. 이에 따라 RTAP의 제약식 (16)과 (17)을 완화하여 RTAP를 아래의 RRTAP로 완화한다.

(RRTAP)

$$\text{Min} \sum_{j \in R} \sum_{k \in C_1} \sum_{l \in C_2} w_j^{kl} (d_j^{kl} - y_j^{kl})$$

$$\text{s.t. } (14) - (15)$$

$$\sum_{j \in R} \sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl} \leq \bar{N}_c + \bar{N}_r$$

$$\sum_{j \in R} \sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl} \leq \bar{N}_r$$

$$(18) - (19)$$

$$(22).$$

RRTAP에서는 RTAP에서와는 달리 clear-sky RCST의 경우 clear-sky 타임슬롯과 rain-fade 타임슬롯을 동시에 사용할 수 있도록 하고 있다. 따라서 RRTAP의 최적 목적식 값은 RTAP의 최적 목적식 값보다 작거나 같다.

**Proposition 1.** RRTAP의 최적 목적식 값은 RTAP의 하한(lower bound)이다.

Proof. 생략.

RRTAP는 rain-fade RCST에 우선권을 주어 rain-fade RCST에 우선적으로 타임슬롯을 할당한다는 가정에 따라 두 개의 subproblem으로 나누어 단계적으로 풀 수 있다. 첫 번째 subproblem RRTAP<sub>r</sub>은 rain-fade RCST만을 고려하는데, rain-fade RCST에 우선적으로 rain-fade 타임슬롯을 할당하는 것이다. 두 번째 subproblem RRTAP<sub>c</sub>는 clear-sky RCST만을 고려하는데, rain-fade RCST에 할당하고 남은 rain-fade 타임슬롯과 clear-sky 타임슬롯을 clear-sky RCST에 할당한다.

(RRTAP<sub>r</sub>)

$$\begin{aligned} & \text{Min } \sum_{j \in R_r} \sum_{k \in C_1} \sum_{l \in C_2} w_j^{kl} (d_j^{kl} - y_j^{kl}) \\ & \text{s.t. } y_j^{kl} \leq d_j^{kl}, \quad j \in R_r, k \in C_1, l \in C_2 \\ & \quad \sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl} \leq Q_j, \quad j \in R_r \\ & \quad \sum_{j \in R_r} \sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl} \leq \overline{N}_r - \sum_{j \in R_c} \sum_{k \in C_1} m_j' \\ & \quad y_j^{kl} \geq a_j^{kl} d_j^{kl}, \quad j \in R_r, k \in C_1, l \in C_2 \\ & \quad \sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl} \geq m_j, \quad j \in R_r \\ & \quad y_j^{kl}: \text{nonnegative integer.} \end{aligned}$$

여기서  $m_j' = \max \{m_j, \sum_{k \in C_1} \sum_{l \in C_2} a_j^{kl} d_j^{kl}\}$  은 RCST  $j$  의 최소 요구량을 나타낸다.

(RRTAP<sub>c</sub>)

$$\begin{aligned} & \text{Min } \sum_{j \in R_c} \sum_{k \in C_1} \sum_{l \in C_2} w_j^{kl} (d_j^{kl} - y_j^{kl}) \\ & \text{s.t. } y_j^{kl} \leq d_j^{kl}, \quad j \in R_c, k \in C_1, l \in C_2 \\ & \quad \sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl} \leq Q_j, \quad j \in R_c \\ & \quad \sum_{j \in R_c} \sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl} \leq \overline{N}_c + \overline{N}_r \\ & \quad y_j^{kl} \geq a_j^{kl} d_j^{kl}, \quad j \in R_c, k \in C_1, l \in C_2 \\ & \quad \sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl} \geq m_j, \quad j \in R_c \\ & \quad y_j^{kl}: \text{nonnegative integer.} \end{aligned}$$

여기서  $\overline{N}_c$  은 rain-fade RCST에 할당되지 않고 남은 rain-fade 타임슬롯의 수를 나타내는데, 이 값은 subproblem RRTAP<sub>r</sub>의 최적해로부터 구해진다.

**Proposition 2 (Optimality Condition No. 1).**  
 $\overline{N}_r' = 0$  이면, RRTAP의 최적해(RRTAP<sub>r</sub>과 RRTAP<sub>c</sub>의 최적해)는 RTAP의 최적해이다.  
Proof. 생략.

**Proposition 3 (Optimality Condition No. 2).**  
 $\sum_{j \in R_c} y_j \leq \overline{N}_c$  이면, RRTAP의 최적해(RRTAP<sub>r</sub>과 RRTAP<sub>c</sub>의 최적해)는 RTAP의 최적해이다.  
Proof. 생략.

#### 4. 최적 타임슬롯 할당 체계

여기에서는 3단계로 구성된 최적 타임슬롯 할당 체계를 제시한다. Phase 1에서는 clear-sky TRF 타임슬롯들의 집합인  $S_c$  와 rain-fade TRF 타임슬롯들의 집합인  $S_r$  을 결정한다. 이 집합들  $S_c$  와  $S_r$  은 집합  $R_c$  및  $R_r$  과 수요 행렬

$\{d_j^{kl}\}$  에 의해 결정된다. Phase 2와 Phase 3에서는 주어진  $S_c$  와  $S_r$  하에서 TAP 문제를 푼다. Phase 2에서는 각 RCST의 각 서비스/지연 클래스에 할당되는 타임슬롯의 수를 결정하고(RTAP를 풀고), Phase 3에서는 각 RCST의 각 서비스/지연 클래스에 할당된 타임슬롯들의 스케줄을 결정한다.

#### 4.1 Phase 1: 타임슬롯 집합 결정

Phase 1에서는 clear-sky TRF 타임슬롯들의 집합인  $S_c$  와 rain-fade TRF 타임슬롯들의 집합인  $S_r$  을 결정한다. 이들 집합들은 clear-sky 프레임을 생성하는데 사용되는 타임-주파수 블록  $T_{sf} \times W_f$  의 수인  $B_c$  와 rain-fade 프레임을 생성하는데 사용되는 타임-주파수 블록  $T_{sf} \times W_f$  의 수인  $B_r$  에 의해 정해진다. 제2절에 제시된 슈퍼 프레임 구조 하에서  $B_c$  와  $B_r$  의 최대값과 최소값은 각각 4와 0이다. 따라서  $(B_c, B_r)$  이 가질 수 있는 값은 (4,0), (1,3), (2,2), (1,3), (0,4) 등이다.

$(B_c, B_r)$  은 이용가능한 총 타임슬롯의 수, clear-sky RCST와 rain-fade RCST간의 형평성(fairness), 최소 요구량(minimal requirement) 등의 기준을 사용하여 정해진다. 즉, clear-sky RCST와 rain-fade RCST간의 형평성 조건(fairness condition)과 최소 요구량 조건(minimal requirement condition)을 만족하면서 이용가능한 총 타임슬롯의 수를 최대화할 수 있도록 정해진다. 먼저, 최소 요구량 조건은 다음과 같다.

$$\begin{aligned} \sum_{j \in R_c} m_j' & \leq \overline{N}_c \quad (-N_c \times B_c) \\ \sum_{j \in R_r} m_j' + \sum_{j \in R_c} m_j' & \leq \overline{N}_r \quad (-N_r \times B_r \times 8). \end{aligned}$$

형평성 조건은 clear-sky RCST의 평균 GOS에 대한 rain-fade RCST의 평균 GOS의 비율이 주어진 형평성 비율(fairness ratio) 이상이어야 한다는 것이다. clear-sky RCST의 평균 GOS와 rain-fade

RCST의 평균 GOS를 각각  $\frac{\sum_{j \in R_c} y_j}{\sum_{j \in R_c} d_j}$  와

$\frac{\sum_{j \in R_r} y_j}{\sum_{j \in R_r} d_j}$  로 정의하고 rain-fade RCST에 우선권을 줄 경우,

$$\begin{aligned} \sum_{j \in R_r} y_j & - \min \left\{ \sum_{j \in R_r} d_j, \overline{N}_r - \sum_{j \in R_c} m_j' \right\} \\ \sum_{j \in R_c} y_j & - \min \left\{ \sum_{j \in R_c} d_j, \overline{N}_c + \overline{N}_r - \sum_{j \in R_r} y_j \right\} \end{aligned}$$

이 된다. 여기서  $y_j = \sum_{k \in C_1} \sum_{l \in C_2} y_j^{kl}$  이고  $d_j = \sum_{k \in C_1} \sum_{l \in C_2} d_j^{kl}$  이다.

$B_r$  값이 증가함에 따라 clear-sky RCST의 평균 GOS에 대한 rain-fade RCST의 평균 GOS의

비율은 개선되고 이용가능한 총 타임슬롯의 수는 감소한다. 따라서 clear-sky RCST와 rain-fade RCST간의 형평성 조건과 최소 요구량 조건을 만족하면서 총 타임슬롯의 수를 최대화하는  $(B_c, B_r)$ 은 형평성 조건과 최소 요구량 조건을 만족하는 최소의  $B_r$ 로부터 얻어질 수 있다 (따라서  $B_c - 4 - B_r$ ).

형평성 조건에서 사용되는 형평성 비율(fairness ratio)을  $FR_0$ 이라고 할 때, 타임슬롯 집합을 결정하기 위한 프로시저 TSD는 다음과 같이 정리된다.

**Procedure TSD** (Timeslot Set Determination)  
생략.

#### 4.2 Phase 2: 타임슬롯 할당량 결정

Phase 2에서는 각 RCST의 각 서비스/지연 클래스에 할당되는 타임슬롯의 수를 최적화하기 위해서 RTAP를 푼다. 이를 위해 subproblem RRTAP<sub>r</sub>과 subproblem RRTAP<sub>c</sub>를 아래의 프로시저 TAD<sub>r</sub>과 TAD<sub>c</sub>에 의해 순차적으로 해결한다. 서비스 클래스와 지연 클래스는 각각 1, ...,  $|C_1|$  과 1, ...,  $|C_2|$  와 같이 나누고, 숫자가 클수록 우선순위가 높다고 가정한다. 또한, 서비스 클래스가 지연 클래스보다 우선한다고 가정한다.

**Procedure TAD<sub>r</sub>** (Timeslot Amount Determination)  
생략.

**Procedure TAD<sub>c</sub>** (Timeslot Amount Determination)  
생략.

**Proposition 4.** 프로시저 TAD<sub>r</sub>(TAD<sub>c</sub>)는 subproblem RRTAP<sub>r</sub> (RRTAP<sub>c</sub>)의 최적해를 구한다.  
Proof. 생략.

프로시저 TAD<sub>r</sub>과 TAD<sub>c</sub>에 의해 구한 해가 Proposition 2 또는 Proposition 3의 조건을 만족하는 경우 RTAP의 최적해가 구해진 것이다.

한편  $R_c^0$ 를 clear-sky 타임슬롯이 할당되는 clear-sky RCST들의 집합으로 정의했을 때, 프로시저 TAD<sub>c</sub>에 의해 구한 각 clear-sky RCST에 대한 할당량  $y_j$ 를 이용하여  $\sum_{j \in R_c^0} y_j - \overline{N}_c$  인  $R_c^0$ 를 구할 수 있다면 프로시저 TAD<sub>r</sub>과 TAD<sub>c</sub>에 의해 구한 해(RRTAP의 최적해)가 RTAP의 최적해가 된다.

**Proposition 5 (Optimality Condition No. 3).** 프로시저 TAD<sub>c</sub>에 의해 구한 각 clear-sky RCST에 대한 할당량  $y_j$ 를 이용하여  $\sum_{j \in R_c^0} y_j - \overline{N}_c$  인  $R_c^0$ 가 존재하면 프로시저 TAD<sub>r</sub>과 TAD<sub>c</sub>에 의해 구한 해(RRTAP의 최적해)가 RTAP의 최적해이다.  
Proof. 생략.

아래의 프로시저 OC는 프로시저 TAD<sub>r</sub>과 TAD<sub>c</sub>에 의해 구한 해가 RTAP의 최적해인지를 Optimality Condition No. 1 - 3을 통해 알아보는

프로시저이다.

**Procedure OC** (Optimality Check)  
생략.

프로시저 OC를 통해 최적해로 판명되지 않은 경우에는 subproblem RRTAP<sub>r</sub>과 RRTAP<sub>c</sub>의 최적해로부터 RTAP의 실행가능 해를 아래의 프로시저 FS에 의해 구한다.  $R_c^0$ 가 clear-sky 타임슬롯이 할당되는 clear-sky RCST들의 집합을 나타내므로,  $R_c - R_c^0$ 는 rain-fade 타임슬롯이 할당되는 clear-sky RCST들의 집합을 나타낸다.

**Procedure FS** (Feasible Solution)  
생략.

**Proposition 6 (Optimality Condition No. 4).** 프로시저 FS에 의해 구한 실행가능 해의 목적식 값이 프로시저 TAD<sub>r</sub>과 TAD<sub>c</sub>에 의해 구한 해의 목적식 값과 동일할 수 있고, 이 경우 최적  $R_c^*$ 가 사용되었다면 프로시저 FS에 의해 구한 해는 RTAP의 최적해가 된다.  
Proof. 생략.

#### 4.3 Phase 3: 타임슬롯 스케줄링

Phase 3에서는 Phase 2에서 구해진 각 RCST의 각 서비스/지연 클래스별 타임슬롯 할당량만큼 특정 타임슬롯을 지정한다. 이를 위해 rain-fade 타임슬롯과 clear-sky 타임슬롯을 시간과 주파수에 따라 번호를 매긴다. clear-sky 타임슬롯과 rain-fade 타임슬롯 각각에 대해서 가장 낮은 주파수와 가장 이른 시간대의 타임슬롯을 0번으로 가장 높은 주파수와 가장 늦은 시간대의 타임슬롯을  $\overline{N}_c - 1$ 과  $\overline{N}_r - 1$ 로 번호를 매긴다.  $B_c$ 가 3이요  $B_r$ 이 1인 경우, clear-sky 타임슬롯은 0번에서 93119번까지의 번호가 부여되고, rain-fade 타임슬롯은 0번에서 19839번까지의 번호가 부여된다. 타임슬롯 스케줄링은 프로시저 TS에 의해 순차적으로 이루어진다.

**Procedure TS** (Timeslot Scheduling)  
생략.

#### 5. 성능 분석

여기에서는 실험 데이터를 이용하여 성능을 분석하고자 한다. 성능 분석에 사용되는 주요 데이터는 다음과 같다. 먼저 서비스 클래스 수와 지연 클래스 수는 각각 5라고 가정하였고, rain-fade RCST의 수와 clear-sky RCST의 수는 각각 150과 90으로 가정하였다. 다음으로 각 RCST의 서비스/지연 클래스에 대한 수요  $d_j^{kl}$ 는 다음과 같이 구하였다.

<표 1> 서비스 클래스별 하한, 상한, 주기의 예

서비스 클래스 $k$	하한	상한	주기(period)
1	19	100	400
2	19	60	100
3	19	300	30
4	19	450	24
5	19	800	12

$l=1$  이면,

$\frac{1}{\text{주기}}$  의 확률로  
일양분포  $U(\text{하한}, \text{상한})$  으로부터  
 $a_j^{kl}$  생성

$(1 - \frac{1}{\text{주기}})$  의 확률로  
 $a_j^{kl} - \text{하한}$ ,

$l \geq 2$  이면,  
 $a_j^{kl} - \frac{1}{l^2} \times a_j^{kl}$ .

또한,  $a_j^{kl}$  은 다음과 같이 가정하였다.

$$k > \frac{|C_1|}{2} + 1 \text{ 이면, } a_j^{kl} = 0.7,$$

$$k \leq \frac{|C_1|}{2} + 1 \text{ 이면, } a_j^{kl} = 0.5.$$

마지막으로,  $Q_j$  와  $m_j$  는 각각 2480과 96으로 가정하였고, Phase 1에서 사용되는 형평성 비율  $FR_0$  은 1.0으로 가정하였다.

제시한 최적 타임슬롯 할당 체계를 이용해 구한 타임슬롯 할당 결과를 정리하면 <표 2>와 같다. 먼저 Phase 1에서 rain-fade 타임슬롯의 수와 clear-sky 타임슬롯의 수가 각각 39680개와 62080개로 결정되었다. 다음으로 Phase 2를 수행한 결과 Optimality Condition No. 2를 만족하는 것으로 나타났고, 따라서 Gap은 rain-fade RCST와 clear-sky RCST 모두 없다. Optimality Condition No. 2를 만족했다는 것은 clear-sky RCST에 할당된 타임슬롯 모두를 clear-sky 타임슬롯만으로 충족시킬 수 있다는 것을 의미한다. 마지막으로 요청한 수요를 얼마나 충족시키는가를 나타내는 충족률은 전체 RCST의 평균 충족률, rain-fade RCST의 평균 충족률, clear-sky RCST의 평균 충족률 모두 1.0인 것으로 나타났다. Gap과 충족률 결과를 통해 타임슬롯에 여유가 있는 경우라는 것을 알 수 있다.

다음으로 서비스 클래스별 주기가 감소함에 따

<표 2> 타임슬롯 할당 결과

타임슬롯 수	$\overline{N_r}$	39680 ( $B_r - 2$ )
	$\overline{N_c}$	62080 ( $B_c - 2$ )
Optimality Condition 만족 여부	Optimality Condition	No. 2 만족
Gap	RF	0.0*
	CS	0.0*
충족률	전체	1.0**
	RF	1.0***
	CS	1.0****

\* (목적식 값 - 하한) / 하한

$$** \sum_{k \in K} \sum_{l \in L} \sum_{j \in J} \frac{y_j^{kl}}{a_j^{kl}} / (|R| \times |C_1| \times |C_2|)$$

$$*** \sum_{k \in K} \sum_{l \in L} \sum_{j \in J} \frac{y_j^{kl}}{a_j^{kl}} / (|R_r| \times |C_1| \times |C_2|)$$

$$**** \sum_{k \in K} \sum_{l \in L} \sum_{j \in J} \frac{y_j^{kl}}{a_j^{kl}} / (|R_c| \times |C_1| \times |C_2|)$$

라(발생확률이 증가함에 따라, 수요가 증가함에 따라) 타임슬롯 할당 결과에 어떤 변화가 있는지를 살펴보면, <표 3>과 같다. 표에서 기준값은 <표 1>에 제시된 값을 의미한다. 표에서 알 수 있듯이 서비스 클래스별 주기가 감소함에 따라 예상대로 충족률은 감소한다. 한편, 서비스 클래스별 주기에 관계없이(수요의 많고 적음에 관계없이) 제시한 최적 타임슬롯 할당 체계는 최적해를 구하거나 Gap이 아주 작은 실행가능 해를 구한다는 것을 알 수 있다.

마지막으로 제시한 최적 타임슬롯 할당 체계의 계산 시간을 살펴본다. 계산 시간은 각 유형의 문제를 10000번 수행하고 그 평균값으로 구하였다. 서비스 클래스 수와 지연 클래스 수가 각각 5이고, rain-fade RCST의 수와 clear-sky RCST의 수가 각각 150과 90인 경우를 대상으로 구한 계산 시간 결과가 <표 4>에 정리되어 있는데, 계산 시간은 주기에 관계없이(수요에 관계없이) 평균적으로 8ms에 미치지 못하는 것으로 나타났다.

## 6. 결론

본 논문에서는 먼저 clear-sky RCST와 rain-fade RCST가 존재하고 복수의 서비스 클래스(service class) 및 지연 클래스(delay class)가 존재하는 광대역 위성 시스템을 대상으로 MF-TDMA 체계 하에서 최적 타임슬롯 할당 문제(timeslot assignment problem: TAP)를 수리적으로 정형화하였다. TAP는 비선형 이진정수계획문제(nonlinear binary integer programming problem)로, NP-complete이다. 따라서 이 문제를 직접 푸는 것은 매우 어렵기 때문에 각 RCST의 각 서비스/지

<표 3> 서비스 클래스별 주기가 변하는 경우의 타임슬롯 할당 결과

주기	B <sub>r</sub>	B <sub>c</sub>	OC	Gap		충족률		
				RF	CS	전체	RF	CS
기본값	2	2	2*	0.0**	0.0**	1.0***	1.0****	1.0*****
기본값/2	3	1	2	0.0	0.0	1.0	1.0	1.0
기본값/3	3	1	-	0.0	.000002	.99999	1.0	.99998
기본값/4	3	1	4	0.0	0.0	.95161	1.0	.87096
기본값/5	3	1	1	0.0	0.0	.79924	.79928	.79917

\* 만족하는 Optimality Condition

\*\* (목적식 값 - 하한) / 하한

$$\dots \sum_{k \in R} \sum_{i \in C_1} \sum_{j \in C_2} \frac{y_i^{kj}}{d_j^{kj}} / (|R| \times |C_1| \times |C_2|)$$

$$\dots \sum_{k \in R} \sum_{i \in C_1} \sum_{j \in C_2} \frac{y_i^{kj}}{d_j^{kj}} / (|R_r| \times |C_1| \times |C_2|)$$

$$\dots \sum_{k \in R} \sum_{i \in C_1} \sum_{j \in C_2} \frac{y_i^{kj}}{d_j^{kj}} / (|R_c| \times |C_1| \times |C_2|)$$

연 클래스에 할당되는 타임슬롯의 수를 결정하는 문제 RTAP로 단순화하였다. 그런데 이 문제 역시 비선형 정수계획문제로 풀기 어렵기 때문에 제약식을 완화하여 RRTAP로 단순화하였다.

다음으로 3단계로 구성된 최적 타임슬롯 할당 체계를 제시하였다. Phase 1에서는 clear-sky TRF 타임슬롯들의 집합과 rain-fade TRF 타임슬롯들의 집합을 결정한다. 이 집합들은 clear-sky RCST와 rain-fade RCST의 수와 수요에 의해 결정된다. Phase 2와 Phase 3에서는 TAP 문제를 푸는데, Phase 2에서는 각 RCST의 각 서비스/지연 클래스에 할당되는 타임슬롯의 수를 결정하고(RTAP를 풀고), Phase 3에서는 각 RCST의 각 서비스/지연 클래스에 할당된 타임슬롯들의 스케줄을 결정한다.

마지막으로 불균등 수요 데이터를 활용하여 제시한 최적 타임슬롯 할당 체계의 성능을 분석하였다. 성능 분석 결과 제시한 최적 타임슬롯 할당 체계는 빠른 시간 내에 최적해 또는 최적해와 아주 가까운 해를 안정적으로 구한다는 사실을 확인할 수 있었다.

### 참고문헌

- [1] ETSI, Digital video broadcasting (DVB): Interaction channel for satellite distribution systems, ETSI EN 301 790 (v.1.2.2), 2000.
- [2] Z. Jiang, Y. Li and V.C.M. Leung, "A predictive demand assignment multiple access protocol for broadband satellite networks supporting internet applications," IEEE ICC, 2002.4.
- [3] H.-J. Lee, S. K. Shin, D.-G. Oh, and J.-M. Kim, "Development of satellite broadband interactive multimedia access technology,"

<표 4> 계산 시간

주기	시간 (ms)
기본값	6.015*
기본값/2	6.184
기본값/3	7.068
기본값/4	7.854
기본값/5	5.921

\* Pentium III PC 930MHz

APSCC, 2002.2.

- [4] K.-D. Lee, Y.-H. Cho, S. J. Lee, and H.-J. Lee, "Optimal design of superframe pattern for DVB-RCS return link," ETRI Journal, vol. 24, no. 3, pp. 251-254, 2002.6.
- [5] Y. Li, Z. Jiang and V.C.M. Leung, "Performance evaluations of PRR-CFDAMA for TCP traffic over geosynchronous satellite links," IEEE WCNC'02, 2002.3.
- [6] M. Mobasseri and V.C.M. Leung, "A new buffer management scheme for multimedia terminals in broadband satellite networks," Thirty-Fifth Hawaii International Conference on System Sciences, 2002.1.
- [7] M. Mobasseri and V.C.M. Leung, "Bandwidth Assignment for VBR Traffic in Broadband Satellite Networks," IEEE CCECE'00, 2000.5.
- [8] J. M. Park, U. R. Savagaonkar, E. K. P. Chong, H. J. Siegel, and S. D. Jones, "Efficient Resource Allocation for QoS Channels in MF-TDMA Satellite Systems," IEEE MILCOM, 2000.10.