

그래프 채색에 의한 타임 슬롯 할당 알고리즘

Time Slot Assignment Algorithm with Graph Coloring

권보섭
안동대학교 컴퓨터교육과

BoSeob Kwon(bxkwon@andong.ac.kr)

요약

위성 통신 분야에서 널리 사용되는 시분할 다중 스위칭 시스템은 많은 저대역폭 가입자들로부터 발생되는 트래픽을 반복되는 프레임에 타임 슬롯을 할당해야 한다. 본 논문에서는 타임 슬롯 할당을 위한 새로운 방법을 제안한다. 기존의 방법인 네트워크 흐름 모델을 사용하지 않고 새로운 방법인 그래프 채색방법을 사용하여 효율적인 타임 슬롯 할당 알고리즘을 제안하였다. 제안된 알고리즘은 주어진 트래픽의 프레임 길이가 2의 멱승일 경우 트래픽을 정확히 반으로 나누어 할당한다. 분할된 트래픽의 프레임 길이가 1이 될 때까지 이 과정을 계속적으로 반복해 분할한다. 제안된 알고리즘의 시간 복잡도는 프레임의 길이가 L 이고 스위치 크기가 N 인 경우에는 기존의 네트워크 흐름 모델을 사용한 최적의 타임 슬롯 할당 알고리즘의 시간 복잡도는 $O(N^{4.5})$ 인데 반해 $O(NL \log_2 L)$ 이다.

■ 중심어 : | 그래프 채색 | 타임 슬롯 | 다중 스위치 | 시간복잡도 |

Abstract

A simple Time Division Multiplexing(TDM) switching system which has been widely in satellite networks provides any size of bandwidth for a number of low bandwidth subscribers by allocating proper number of time slots in a frame. In this paper, we propose a new approach based on graph coloring model for efficient time slot assignment algorithm in contrast to network flow model in previous works. When the frame length of an initial matrix of time slot requests is 2^s 's power, this matrix is divided into two matrices of time slot requests using binary divide and conquer method based on the graph coloring model. This process is continued until resulting matrices of time slot requests are of length one. While the most efficient algorithm proposed in the literature has time complexity of $O(N^{4.5})$, the time complexity of the proposed algorithm is $O(NL \log_2 L)$, where N is the number of input/output links and L is the number of time slot allotted to each link in the frame.

■ keyword : | Graph Coloring | Time Slot | Time Complexity | Frame |

I. 서론

최근 사용자에게 전역 커버리지(coverage), 자연스러

운 방송(broadcast)능력 및 광대역 접근(broadband access)의 장점을 가진 위성통신이 세계 어디서나 통신이 가능하게 하는 이동통신 서비스의 수단으로 주목받

* 본 논문은 2004년 안동대학교 특별학술연구지원사업에 의하여 연구되었습니다.

접수번호 : #080327-001

접수일자 : 2008년 03월 27일

심사완료일 : 2008년 05월 02일

교신저자 : 권보섭, e-mail : bxkwon@andong.ac.kr

고 있으며, 인터넷의 성능향상을 위한 강력한 수단으로 제안되고 있다[1]. 그 중에서도 시간지연에 따른 통화 지연 발생이 적은 저궤도(Low Earth orbit : LEO) 위성통신 시스템은 하나의 네트워크로 동작하며 각 위성이 하나의 스위치로 동작한다. 서로 인접한 위성은 위성간 링크(Intersatellite links)로서 연결된다[1].

다수의 LEO 위성통신 시스템이 동작 혹은 개발 중에 있지만 대부분의 이 시스템들은 많은 저대역폭 가입자들로부터 발생하는 트래픽을 적은 수의 고속 스위치 링크에 집중시킬 필요성 때문에 시분할 다중(TDM : Time-Division Multiplex) 스위칭 시스템을 널리 사용하고 있다. 이 시스템의 스위칭과 전송은 연속적으로 반복되는 프레임에 따라 동작되며, 기본 단위시간인 타임 슬롯(time slot)으로 구성되어 있다. 각 타임 슬롯은 트래픽이 충돌없이 스위치 될 수 있도록 스케줄링되어야 한다. 따라서 위성통신 시스템에서의 목적은 트랜스폰더(transponder)가 최대효율을 가지도록 주어진 트래픽을 스위칭하고 전송할 수 있도록 하는 것이다.

입력 링크 i 로부터 출력 링크 j 로 한 프레임 당 요구하는 타임 슬롯 수를 $d_{ij}(1 \leq i, j \leq N)$ 이라 하면, $D = [d_{ij}]^{N \times N}$ 은 입출력 링크간에 요구되는 대역폭을 의미하는 트래픽 행렬로 정의할 수 있다. TDM 스위칭 시스템에서의 이 트래픽 행렬을 충돌없이 스위치 될 수 있도록 하기 위해서는 스위치의 입출력 링크에 2개 이상의 패킷이 동시에 할당되지 않는다는 물리적 제약조건이 존재한다.

타임 슬롯 할당(Time Slot Assignment : TSA) 문제는 트래픽 행렬에 물리적 제약조건을 위배하지 않고 존재하는 수 많은 타임 슬롯 할당 방법 중에서 가장 작은 프레임의 길이를 가지고 주어진 트래픽을 할당하는 방법을 찾는 것이다. TSA 알고리즘은 [3][4]에서 정형화한 이후 많은 연구가 이루어졌고, 또한 최근에는 대규모 센서 네트워크[5][12]에서 다시 주목하기 시작했다. 지난 수 십년간 $N \times N$ 크기의 TDM 스위칭 시스템에서 최적의 TSA를 구하는 알고리즘들이 제안되어져 왔으나[6-8], 그중에 가장 효율적인 알고리즘[7]에서 $O(N^{4.5})$ 으로 알려져 왔다. 이 알고리즘은 네트워크 호

름 모델을 사용하여 프레임의 길이와 관계없이 최적의 알고리즘으로 알려져 있다. 그러나 본 논문에서는 만약 프레임의 길이(L)가 작은 경우 그래프 채색을 사용하여 이보다 더 최적의 알고리즘을 구할 수 있는 방법을 제시하고자 한다.

II. 문제 정의

본 장에서는 TSA 문제에 필요한 용어들을 정의하고 이를 사용하여 TSA 문제를 정의하기로 한다.

스위치는 N 개의 입출력 가지는 $N \times N$ 크기의 논블록킹(Nonblocking) 스위치이다. 입력 i 로부터 출력 j 로 한 프레임 당 요구되는 타임 슬롯의 수를 $d_{ij}(1 \leq i, j \leq N)$ 라 하면, $D = [d_{ij}]^{N \times N}$ 은 입출력 간에 요구되는 대역폭을 의미하는 트래픽 행렬로 정의할 수 있으며 TSA 알고리즘은 주어진 트래픽 행렬 D 에 대하여 전송 가능 스케줄(feasible transmission schedule)을 찾는 문제로 정의할 수 있다.

먼저 TSA 알고리즘에 관련된 정의와 용어들은 처음 이 문제를 형식화한 [3]을 기초로 하고 있으며 이후 많은 문헌들은 이들을 인용하고 있다. [3]에서 사용된 정의와 용어는 다음과 같다.

[정의 1] $N \times N$ 트래픽 행렬 D 에 대하여 모든 원소들의 합을 T , 각 행(열)의 원소들의 합을 $r_i(c_j)$ 로 정의한다[3]. 즉,

$$T = \sum_{i=1}^M \sum_{j=1}^N d_{ij} \tag{1}$$

$$r_i = \sum_{j=1}^N d_{ij} \tag{2}$$

$$c_j = \sum_{i=1}^M d_{ij} \tag{3}$$

[정의 2] 기본적인 TSA 문제는 $N \times N$ 트래픽 행렬 D 가 주어지면 $D = \sum_{i=1}^h S_i$ 을 만족하면서 $N \times N$ 크

기인 행렬 $S = \{S_1, S_2, \dots, S_i, \dots, S_h\}$ 의 열을 구한다. 여기서 S_i 는 $N \times N$ 의 크기인 스위칭 행렬 (Switching matrix)이며, h 는 임의의 음이 아닌 정수이다.

[정의 3] $N \times N$ 스위칭 행렬 S_i 은 다음과 같은 조건은 만족한다.

- i) 모든 원소들은 음수가 아닌 정수를 가진다.
- ii) 각 행과 열에 대하여 원소가 0이 아닌 정수는 오직 하나 혹은 0개를 가진다.

아래는 S_i 의 한 예들을 나타내었다.

$$\begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}, \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 1 \\ 2 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

[정의 4] $N \times N$ 스위칭 행렬 S_i 내의 원소 중에 가장 큰 값을 LS_i 이다.

S_i 내의 원소 중에 가장 큰 값 LS_i 의 의미는 스위칭 행렬 S_i 에 대해서 전송에 필요한 최소한의 타임 슬롯 요구량을 의미한다. 앞에서 서술한 바와 같이 한 타임 슬롯에 $N \times N$ 스위치는 하나의 입출력에 대하여 2개 이상의 패킷이 동시에 들어오거나 나가지 않는다는 물리적 제약조건이 존재하기 때문이다.

[정의 5] 스위칭 행렬 $S_i(1 \leq i \leq h)$ 와 각 S_i 에 대한 LS_i 와 프레임 길이 L 과의 관계는 다음과 같다.

$$L = \sum_{i=1}^h LS_i \quad (4)$$

[정의 6] 트래픽 행렬 D 에 대하여 최소의 프레임 길이 L 은 다음과 같다[3].

$$L = \max(\lceil T/N \rceil, \max(r_1, r_2, \dots, r_i, \dots, r_N, c_1, c_2, \dots, c_i, \dots, c_N)) \quad (5)$$

최소 프레임 길이 L 은 $N \times N$ 스위치에서 트래픽 행렬 D 가 주어지면 전송 가능한 최소한의 타임 슬롯 요구량이며 최소한의 프레임 길이가 된다.

다음은 트래픽 행렬 D 가 식(6)과 같이 주어질 경우 $r_1, r_2, r_3, c_1, c_2, c_3, T, L$ 값들을 보여 주고 있다.

$$D = \begin{pmatrix} 5 & 2 & 2 \\ 1 & 6 & 1 \\ 3 & 0 & 5 \end{pmatrix} \quad (6)$$

$$\cdot r_1 = \sum_{j=1}^N d_{1j} = 9, \quad r_2 = \sum_{j=1}^N d_{2j} = 8,$$

$$r_3 = \sum_{j=1}^N d_{3j} = 8,$$

$$\cdot c_1 = \sum_{i=1}^N d_{i1} = 9, \quad c_2 = \sum_{i=1}^N d_{i2} = 8,$$

$$c_3 = \sum_{i=1}^N d_{i3} = 8$$

$$\cdot T = \sum_{i=1}^N \sum_{j=1}^N d_{ij} = 25$$

$$\cdot L = \max(\lceil T/N \rceil, \max(r_1, r_2, \dots, r_i, \dots, r_N, c_1, c_2, \dots, c_i, \dots, c_N)) = \max(\lceil 25/3 \rceil, \max(9, 8, 8, 9, 8, 8)) = 9$$

기존의 TSA 알고리즘을 위에서 사용된 용어로 정의하면 다음과 같다. [정의 1]을 만족하는 트래픽 행렬 D 가 주어지면 그로부터 [정의 6]에 의한 최소 프레임 길이 L 을 계산한 후, [정의 3]을 만족하는 스위칭 행렬 h 개를 구하는 문제이다. 이 때 스위칭 행렬 S_i 에 대한 LS_i 반드시 [정의 4]와 [정의 5]를 만족해야 한다.

이와 같이 기존의 TSA 알고리즘은 [3][4]에서 형식화된 [6]에서 $O(N^5)$ 의 알고리즘을 구하고 이후 [8]에서 최적의 알고리즘이라고 알려진 $O(N^{4.5})$ 을 제안하였다. 그러나 [1]에서 LEO 위성 클러스터(Cluster)들이 트리-구조를 가진다면 $O(N^4)$ 의 알고리즘들도 제안되고 있다. 또한 최근에 센서 네트워크[5][12] 및 광 네트워크 분야[13]에서 새로운 관점에 TSA 문제를 재조명하고 있다[2][10].

[정의 7] 스위칭 행렬의 [정의 3] 조건 1에서 모든 원소가 0 혹은 1만 가지는 행렬을 (0,1)-스위칭 행렬 Su_i 로 정의한다. 즉, 스위칭 행렬 중에서 각 행과 열에 대하여 그 합이 1을 넘지 않는 행렬이므로 1 타임슬롯에서 물리적 제약조건을 만족하면서 전송 가능한 행렬이다[1].

스위칭 행렬의 열 $S = \{S_1, S_2, \dots, S_i, \dots, S_h\}$, (0,1)-스위칭 행렬의 열 $Su = \{Su^1, Su^2, \dots, Su^L\}$ 과 트래픽 행렬 D 는 다음과 같은 관계를 가진다.

$$D = \sum_{i=1}^h S_i = \sum_{i=1}^L Su^i \quad (7)$$

$$L = \sum_{i=1}^h LS_i \quad (8)$$

지금까지 정의된 용어들을 사용하여 $N \times N$ 스위치에 대한 TSA 문제를 다시 정의하면 다음과 같다.

[$N \times N$ 스위치에 대한 TSA 문제]

[정의 1]로 정의된 $N \times N$ 트래픽 행렬 D 가 주어지면, 다음 식(9)에서 식(11)을 만족하는 (0,1)-스위칭 행렬 $Su = \{Su^1, Su^2, \dots, Su^i, \dots, Su^L\}$ 의 열을 구하는 문제로 정의할 수 있다.

$$D = \sum_{i=1}^L Su^i \quad (9)$$

$$r_i^k = \sum_{j=1}^N su_{ij}^k \leq 1, 1 \leq i \leq N \quad (10)$$

$$c_j^k = \sum_{i=1}^N su_{ij}^k \leq 1, 1 \leq j \leq N \quad (11)$$

식(12)는 식(6)의 트래픽 행렬 D 에 대한 $L = 9$ 개인 (0,1)-스위칭 행렬 $Su^1, Su^2, Su^3, \dots, Su^9$ 의 예를 나타내고 있다. 여기서는 어떠한 (0,1)-스위칭 행렬이라도 식(9)에서 식(11)은 반드시 만족해야 한다.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad (12)$$

본 논문에서는 기존의 TSA 알고리즘들이 트래픽 행렬 D 가 주어지면 최소 프레임 길이 L 을 갖는 스위칭 행렬 $S_i (1 \leq i \leq h)$ 을 구하는데 비해 제안된 TSA 알고리즘은 트래픽 행렬 D 에 대해 (0,1)-스위칭 행렬 $Su^i (1 \leq i \leq L)$ 을 구하는 문제로 TSA 문제를 재정의 하였다.

III. 그래프채색에 의한 TSA 알고리즘

$N \times N$ 스위치에 대한 TSA문제는 트래픽 행렬 D 을 적당하게 분해(Decomposition)해야 한다. 먼저 [8]에서 제안한 최적의 알고리즘은 네트워크 흐름을 이용한 트래픽 네트워크로 주어진 트래픽 행렬 D 을 분해한다. [8]에서 제안한 TSA 알고리즘은 이 분해를 반복적으로 호출한다. 다음은 [8]에서 제안한 분해 알고리즘과 TSA 알고리즘을 나타내었다.

Decomposition Algorithm (T, L_1)

step 1 T 와 L_1 에 대응하는 트래픽 네트워크 G_{T,L_1} 을 생성한다.

step 2 G_{T,L_1} 에서 순환(circulation)과 그 에 대응되는 트래픽 행렬 T_1 을 구한다.

step 3 $T_2 \leftarrow T - T_1$

이 알고리즘에 사용되는 트래픽 네트워크와 순환(circulation)은 그래프 이론 중에 네트워크 흐름 모델에서 사용되는 용어이다[8].

TSA Algorithm (T, L)

step 1 if $L = 1$, stop

step 2 *Decomposition Algorithm* ($T, \lfloor L/2 \rfloor$) 호출,
*Decomposition Algorithm*의 결과로서 프레임 길이
 $\lfloor L/2 \rfloor$ 인 트래픽 행렬 T_1 과 $\lfloor L/2 \rfloor$ 인 트래픽 행렬 T_2
 을 생성한다.

step 3 *TSA Algorithm* ($T_1, \lfloor L/2 \rfloor$) 호출

step 4 *TSA Algorithm* ($T_2, \lfloor L/2 \rfloor$) 호출

우리는 분해를 기존의 TSA 알고리즘에서의 네트워크 흐름을 사용하지 않고 그래프 채색 방법을 사용한 새로운 방법의 TSA 알고리즘을 제안한다. 이를 위해 L 의 값이 2의 멱승으로 가정하고 주어진 트래픽 행렬 D 를 분해하고자 한다.

최적의 L 값은 [정의 6]에 서술한 바와 같이 주어진 트래픽을 분해하기 전에 이미 알려진다. 만약 최적의 L 값이 2^l (l 은 음이 아닌 정수)로 가정하면 주어진 트래픽은 다음과 같은 제약조건으로 형식화할 수 있다.

$$\max(\lceil T/N \rceil, \max(r_1, r_2, \dots, r_i, \dots, r_N, c_1, c_2, \dots, c_i, \dots, c_N)) = 2^l \quad (13)$$

$$T = \sum_{i=1}^N \sum_{j=1}^N d_{ij} \leq N \cdot 2^l \quad (14)$$

$$r_i = \sum_{j=1}^N d_{ij} \leq 2^l \quad (15)$$

$$c_j = \sum_{i=1}^N d_{ij} \leq 2^l \quad (16)$$

식(13)에서 식(16)을 만족하는 트래픽 행렬 D 을 공평하게 분해한다는 것은 식(17)에서 식(20)을 만족하는 트래픽 행렬 $D^k = [d_{ij}^k]_{N \times N}$, $k = \{0, 1\}$ 으로 분해 하는 것이라 할 수 있다.

$$\max(\lceil T^k/N \rceil, \max(r_1^k, r_2^k, \dots, r_i^k, \dots, r_N^k, c_1^k, c_2^k, \dots, c_i^k, \dots, c_N^k)) \leq 2^{l-1} \quad (17)$$

$$T^k = \sum_{i=1}^N \sum_{j=1}^N d_{ij}^k \leq N \cdot 2^{l-1} \quad (18)$$

$$r_i^k = \sum_{j=1}^N d_{ij}^k \leq 2^{l-1} \quad (19)$$

$$c_j^k = \sum_{i=1}^N d_{ij}^k \leq 2^{l-1} \quad (20)$$

아래는 주어진 트래픽 행렬 D 을 공평하게 D_0 과 D_1 로 분해한 한 예를 보여주고 있다.

$$D = \begin{pmatrix} 4 & 2 & 2 \\ 1 & 6 & 1 \\ 3 & 0 & 5 \end{pmatrix}$$

$$D_0 = \begin{pmatrix} 2 & 1 & 1 \\ 0 & 3 & 1 \\ 2 & 0 & 2 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 0 \\ 1 & 0 & 3 \end{pmatrix}$$

이 예에서 트래픽 행렬 D 가 식(13)에서 식(16)을 만족하면 두개의 행렬로 분해된 D_0 과 D_1 은 식(17)에서 식(20)을 만족함을 알 수 있다.

다음은 그래프 채색의 방법을 사용하여 트래픽 행렬 D 을 공평하게 분해하는 제안된 *TwoPartition* 함수에 대해 기술하였다. [그림 1]은 이 함수의 분해 모습이고 알고리즘은 아래와 같다.

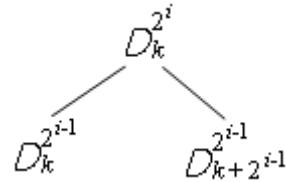


그림 1. *TwoPartition* 함수의 분해

TwoPartition($D_k^{2^i}, i, k$)

step 1 $D' \leftarrow D_k^{2^i}, D_0 = D_1 \leftarrow \lfloor d_{ij}/2 \rfloor_{N \times N}$

step 2 $D' \leftarrow D' - (D_0 + D_1)$

step 3 D' 에서 각 행(열)에 있는 v 들 끼리 서로 쌍으로 연결한다

/* 0이 아닌 원소를 v 라 하자 */

step 4 *while*(D' 에 채색 되지 않은 v 가 존재) {

- 4.1 v 에 빨강색을 채색 한다.
- 4.2 v 에 연결된 간선을 따라 채색된 원소나 해당 간선의 끝을 만날 때 까지 파랑과 빨강을 교대로 채색한다.
- } /* D' 을 모든 검색한다. */

step 5 $D_k^{2^{i-1}} \leftarrow D_0 + \{D'$ 내에 있는 빨강으로 채색된 원소},

$$D_{k+2^{i-1}}^{2^i} \leftarrow D_1 + \{D'$$
내에 파랑색으로 채색된 원소}

TwoPartition 함수는 주어진 트래픽 행렬 $D_k^{2^i}$ 을 공평하게 D_0 와 D_1 으로 나눈다. 이때 $D_k^{2^i}$ 의 원소 값이 짝수이면 그 원소의 값의 반을 [그림 2]의 (b)처럼 D_0 와 D_1 에 해당하는 행과 열에 저장된다. 그러나 원소의 값이 홀수인 경우는 그 원소 값보다 크지 않는 최대 짝수 값의 반을 D_0 와 D_1 에 해당하는 행과 열에 저장하고, 그 나머지 값은 [그림 2]의 (b)의 D' 에 해당 행과 열에 저장한다. 따라서 D' 의 각 원소 값은 0 혹은 1이다.

[그림 3]은 8×8 TDM 스위칭에서 D' 의 원소의 값이 1인 것만 점으로 표시하고 이를 바탕으로 채색 과정을 보인 것인 한 예이다. step 3은 점으로 표시된 D' 행렬에서 행 0, 열 0부터 차례대로 서로의 짝을 연결한다. 여기서 점을 정점(vertex)으로 연결선을 간선(edge)로 표시하면 그래프 이론을 사용할 수 있다. [그림 3]은 사이클 혹은 트리인 서브 그래프의 집합으로 구성됨을 볼 수 있으며, 사이클인 경우는 반드시 정점들의 개수가 짝수이다. 그래프 이론에 의하면 짝수의 정점을 가지는 사이클은 2색의 채색이 가능하며, 트리는 정점의 개수와 무관하게 2색으로 채색이 가능하므로 [그림 3]의 모든 점들을 2색으로 표시할 수 있다.

[그림 2]의 (c)에서는 D' 을 step 4 수행 후의 모습을 보여주고 있다. 그 결과 원소 값들이 빨강(R)에 해당하는 행렬과 파랑(B)에 해당하는 행렬로 분할되고 이를 step 5을 실행하며 [그림 2]의 (d)와 같이 된다.

step 5의 수행 후 두 행렬의 결과를 보면 두 행렬의 각각의 행과 열의 차는 기껏해야 1이며, 이는 식(13)에

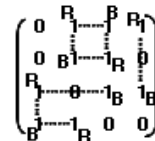
서 식(16)을 만족하는 트래픽 행렬 D 가 주어지면 식(17)에서 식(20)을 만족하는 두개의 행렬 D_1 과 D_2 으로 공평하게 분해할 수 있음을 보여주고 있다.

$$D_0^{2^3} = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 0 & 3 & 1 & 4 \\ 3 & 0 & 3 & 1 \\ 3 & 1 & 2 & 2 \end{pmatrix}$$

(a) 주어진 트래픽 행렬

$$D' = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \quad D_0 = D_1 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

(b) Step 2 수행 후 D', D_0, D_1 행렬



(c) Step 4 수행 후 D' 행렬에 대한 채색 결과

$$D_0^{2^2} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad D_4^{2^2} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 2 & 0 & 2 \\ 1 & 0 & 2 & 1 \\ 2 & 0 & 1 & 1 \end{pmatrix}$$

(d) Step 5 수행 후 행렬들

그림 2. *TwoPartition* 함수에서의 행렬들

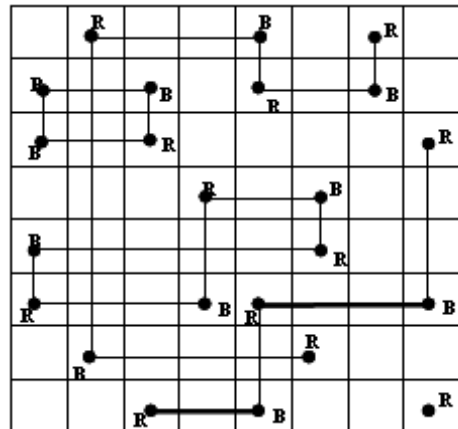


그림 3. 8×8 스위칭에서 홀수 원소 값들에 대한 채색과정

이와 같은 방법으로 $l (= \log_2 L)$ 번째 까지 반복하면 (0,1)-스위칭 행렬 Su^0 에서 Su^7 까지를 구할 수 있

다. 그림 4은 프레임 길이 $L = 2^3$ 으로 주어지는 트래픽 행렬 D 에 대해서 l 번째까지 반복하여 스위칭 행렬을 구하는 과정을 도식적으로 그렸다.

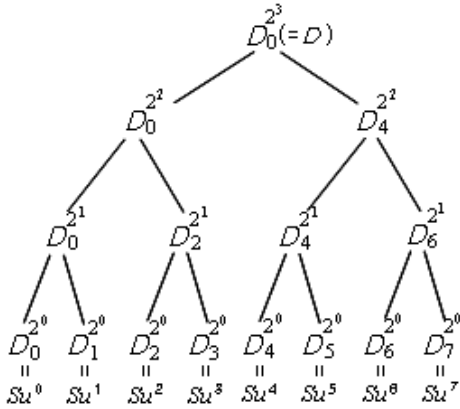


그림 4. $TSA(D_0^{2^3}, 3, 0)$ 함수 호출 과정

주어진 트래픽 행렬 D 는 $D_0^{2^l}$ 의 의미를 가진다. 즉, 프레임의 길이를 2^l 로 가정했으므로 $TSA(D_0^{2^l}, l, 0)$ 을 호출하여 TSA 알고리즘이 시작된다.

아래는 $TSA(D_k^{2^i}, i, k)$ 알고리즘을 순환함수로 기술하였다.

Proposed $TSA(D_k^{2^i}, i, k)$

step 1 if($i = 0$) $Su^k = D_k^{2^i}$, stop

step 2 $TwoPartition(D_k^{2^i}, i, k)$ 을 호출하여

$D_k^{2^{i-1}}$ 와 $D_{k+2^{i-1}}^{2^{i-1}}$ 를 구한다.

step 3 Proposed $TSA(D_k^{2^{i-1}}, i-1, k)$ 호출

step 4 Proposed $TSA(D_{k+2^{i-1}}^{2^{i-1}}, i-1, k+2^{i-1})$

호출

여기서 $D_k^{2^i}$ 의 의미는 i 번째 TSA을 호출하며, 할당되는 타임 슬롯은 k 번째부터 $k + 2^{i-1}$ 번째까지 할당

됨을 의미하고, 초기값은 i 은 l 이고, k 은 0으로부터 시작한다.

이제 제안된 알고리즘의 시간 복잡도를 계산하기 위해서는 먼저 $TwoPartition$ 함수의 시간 복잡도를 계산해야 한다. D^l 은 앞에서 설명했듯이 서브 그래프의 집합으로 생각할 수 있다. 우리는 이 서브 그래프의 정점의 합을 m 개로 가정하자. 이들의 서브 그래프들을 모두 연결 리스트(linked list)로 구현할 수 있다. 따라서 $TwoPartition(D_k^{2^i}, i, k)$ 함수의 각 step마다 시간 복잡도는 정점의 개수에 비례하므로 $O(m)$ 이다.

두 번째로 TSA 알고리즘의 시간 복잡도를 계산하기 위해 $TSA(D_k^{2^i}, i, k)$ 을 고려해 보자. 여기서 트래픽 행렬 $D_k^{2^i}$ 에서의 원소의 개수는 최대 $N \cdot 2^i$ 개이며 $D_k^{2^i}$ 의 개수는 2^{l-i} 이므로 $TSA(D_k^{2^i}, i, \cdot)$ 가 처리해야 할 원소의 개수는 최악의 경우에도 $N \cdot 2^i \cdot 2^{l-i} = N \cdot 2^l$ 을 넘지 않는다. 또한 앞에서 살펴본 $TwoPartition$ 함수의 시간 복잡도는 원소의 개수에 의해 결정되므로 $TSA(D_k^{2^i}, i, \cdot)$ 의 시간 복잡도는 $O(N \cdot 2^l) = O(NL)$ 이 된다.

따라서 프레임 길이가 L 인 트래픽 행렬 D 가 주어지면 $TSA(D_0^{2^l}, l, 0)$ 를 호출하게 되며 시간 복잡도는 $O(NL \cdot l) = O(NL \cdot \log_2 L)$ 이 된다.

$N \times N$ 스위치에서 시간복잡도가 $O(N^{4.5})$ 인 알고리즘이 최적(optimal)으로 알려져 있다. 그래프 채색방법을 사용한 제안된 TSA 알고리즘과 최적의 알고리즘과 단순하게 비교하면 $O(N^{3.5}/L \cdot \log_2 L)$ 의 성능 향상이 이루어졌다고 볼 수 있다.

IV. 결론

우리는 본 논문을 통해 시분할 다중 스위치에서 L 이 2의 제곱 꼴일 때 $O(NL \log_2 L)$ 의 시간 복잡도를 가지는 최적의 알고리즘을 제안하였다. 지금까지 제안된 가장 효율적인 알고리즘은 네트워크 흐름 모델을 적용한

알고리즘[6-8]으로 $O(N^{4.5})$ 의 시간 복잡도를 가진다. 그러나 제안하는 알고리즘은 새로운 방법인 그래프 채색방법을 사용하여 $L \leq N^3$ 인 경우 최소한 $O(\sqrt{N}/\log_2 N)$ 의 시간 복잡도를 개선하였다.

본 논문에서 제안한 알고리즘은 시분할 다중스위치에서만 고려하였지만 최근에 많은 연구가 되고 있는 센서 네트워크[5] 등에도 적용가능하다. 또한, L 이 2의 제곱꼴이 아닌 경우에도 이 알고리즘을 약간 변경하여 쉽게 TSA 문제를 해결할 수 있고, 병렬 알고리즘으로도 발전가능하다[9].

참 고 문 헌

[1] M. A. Bonuccelli, F. Martelli, and S. Pelagatti, "Optimal Packet Scheduling in tree-structured LEO Satellite Cluster," *Mobile Networks and Application*, Vol.9, pp.289-295, 2004.

[2] K. T. Yiu, "Scheduling Satellite-Switched Time-Division Multiple Access With General Switching Modes," *IEEE/ACM Trans.*, Vol.12, No.4, 2004(8).

[3] K. Y. Eng and A. S. Acampora, "Fundamental Conditions Governing TDM Switching Assignments in Terrestrial and Satellite Networks," *IEEE Trans. on Commun.*, Vol.35, pp.755-761, 1987(7).

[4] S. C. Liew, "Comments on 'Fundamental Conditions Governing TDM Switching Assignments in Terrestrial and Satellite Networks'," *IEEE Trans. on Commun.*, Vol.37, pp.187-189, 1989(2).

[5] G. Haigang and L. Ming, "A Two Level TDMA Scheduling Protocol with Intra-cluster Coverage for Large Scale Wireless Sensor Network," *IJCSNS*, Vol.6, No.2B, 2006(2).

[6] M. A. Bonuccelli, "A Fast Time-slot Assignment Algorithm for TDM Hierarchical Switching System," *IEEE Trans. on Commun.*, Vol.37, pp.870-874, 1989(8).

[7] M. A. Bonuccelli, I. Gopal, and C. K. Wong, "Incremental Time-Slot Assignment in SS/TDMA Satellite Systems," *IEEE Trans. on Commun.*, Vol.39, pp.1147-1156, 1991(7).

[8] S. Chalasani and A. Varmar, "An improved Time-slot Assignment Algorithm for TDM Hierarchical Switching Systems," *IEEE Trans. on Commun.*, Vol.41, pp.312-317, 1993(2).

[9] S. Chalasani and A. Varmar, "Parallel Algorithms for Time-slot Assignment in TDM Switching System," *IEEE Trans. on Commun.*, Vol.41 pp.1736-1747, 1993(11).

[10] Isaac Keslassy, Murali Kodialam, T. V. Lakshman, and Dimitrios Stiliadis, "On Guaranteed Smooth Scheduling for Input-Queued Switches," *IEEE/ACM Trans.*, Vol.13, No.6, 2005(12).

[11] K. T. Yiu, "A unified Algorithmic Framework for Variable-Rate TDM Switching Assignments," *IEEE/ACM Trans.*, on Networking, Vol.9, No.5, 2001(10).

[12] S. Kulkarni, "TDMA service for sensor network," In 24th Int. Conf. on Distributed Computing Systems(ICDCS04), ADSN Workshop, pp.604-609, 2004(3).

[13] S. Suresh, J. Eric, Harder, and H. A. Choi, "Scheduling multi-rate sessions in time division multiplexed wavelength-routing networks," *IEEE Journal on Selected Areas in Communications*, Vol.18, No.10, 2000(10)

저 자 소 개

권 보 섭(BoSeob Kwon)

정회원



- 1983년 2월 : 경북대학교 전자공학
학과(공학사)
- 1990년 8월 : 충남대학교 전자공
학과(공학석사)
- 1997년 2월 : 한국과학기술원 전
산학과(공학박사)

- 1983년 1월 ~ 1985년 8월 : LG전자 연구원
- 1985년 9월 ~ 1998년 3월 : 한국전자통신연구원 선
임연구원
- 1998년 3월 ~ 현재 : 안동대학교 컴퓨터교육과 교수
<관심분야> : 컴퓨터교육, 차세대통신망, 센서네트워
크