

# Passive Star형 광상호연결망에서의 All-to-all 방송을 위한 송수신 스케줄링 기법

장 석 문<sup>†</sup> · 변 광 준<sup>††</sup> · 예 흥 진<sup>††</sup> · 위 규 범<sup>††</sup> · 홍 만 표<sup>††</sup>

## 요 약

Passive star형 광상호연결망에서는 송신노드에서 특정 파장을 사용하여 패킷을 방송하면 해당 노드에서 선택적으로 수신하는 방식(broadcast-and-select)으로 노드간의 패킷전달이 이루어진다. 이 때, 모든 노드는 임의의 파장에 대하여 가변적인(tunable) 송신기와 고정적인(fixed-wavelength) 수신기를 각각 하나씩 가지고 있다고 가정하고, 전체 노드의 개수와 파장분할다중화(wavelength division multiplexing)에 의해 동시에 사용할 수 있는 파장의 개수를 각각  $N$ 과  $k$ 라 정의한다. 임의의 노드간에 모든 패킷의 길이가 항상 일정하여 하나의 패킷 송수신이 완료되는 시간을 단위시간(unit time)으로 할 때, 송신기의 파장을 다른 파장으로 변환하기 위한 지연시간(tuning delay)을  $\delta$ 로 나타내기로 한다. 본 논문에서는 송신노드와 수신노드가 동일한 경우를 제외한 all-to-all 방송을 위한 최적 송수신 스케줄의 주기(cycle length)는  $\max\left\{\left\lceil\frac{N}{k}\right\rceil(N-1), k\delta+N-1\right\}$  이상임을 증명하고,  $N-1$ 이  $k$ 로 나뉘어 떨어지는 경우이거나 또는  $\left\lceil\frac{N}{k}\right\rceil(N-1) > k\delta+N-1$ 인 경우에 최적의 송수신 스케줄(optimal transmission schedules)을 구하는 새로운 알고리즘을 제안하였다.

## A Transmission Scheduling Algorithm for All-to-all Broadcast in Optical Passive Star Interconnections

Seokmun Chang<sup>†</sup> · Kwang June Byeon<sup>††</sup> · Hong-Jin Yeh<sup>††</sup>  
· Kyunbum Wee<sup>††</sup> · Manpyo Hong<sup>††</sup>

### ABSTRACT

In optical passive star interconnections, all packets are transmitted between nodes in a broadcast-and-select manner. It is assumed that each node has a tunable transmitter and a fixed-wavelength receiver, and that all packet lengths are equal so that each transmission can be done in a unit time. The tuning delay, denoted by  $\delta$ , means the amount of time for a transmitter to change its wavelength to another one. The problem is, given any value of the number of nodes  $N$  and the number of wavelengths  $k$  according to WDM implementations, to find transmission schedules with minimum cycle length for all-to-all broadcast where no one sends any packet to itself. In this paper, we prove that the

\* 이 논문은 1997년도 정보통신부 대학기초 연구지원사업에 의해 작성되었음.

<sup>†</sup> 준 회 원 : 아주대학교 정보 및 컴퓨터 공학부

<sup>††</sup> 중 심 회 원 : 아주대학교 정보 및 컴퓨터 공학부

논문접수 : 1998년 5월 7일, 심사완료 : 1998년 6월 12일

cycle length of optimal transmission schedules should be at least  $\max \left\{ \left\lceil \frac{N}{k} \right\rceil (N-1), k\delta + N-1 \right\}$ . A novel algorithm for optimal transmission schedules is then presented when  $N-1$  is divisible by  $k$ . This algorithm also can be used for any values of  $N$  and  $k$  if the tuning delay  $\delta$  does not affect strictly the cycle length of transmission schedules, i.e.  $\left\lceil \frac{N}{k} \right\rceil (N-1) > k\delta + N-1$ .

### 1. 서 론

광전송은 낮은 데이터 손실율과 함께 넓은 대역폭을 제공할 수 있으므로 고속의 대량 데이터 전송에 효과적으로 활용될 수 있다. 특히, 파장분할다중화(wavelength division multiplexing) 방식에 의한 광전송은 하나의 광케이블을 통하여 1초당 테라( $10^{12}$ )비트급의 데이터 전송이 가능한 여러 개의 독립적인 채널을 동시에 제공할 수 있으며, 이러한 초고속 통신채널은 대규모 병렬(massively parallel) 컴퓨터 시스템에서 프로세서간의 상호연결망을 구성하는 데에 효과적으로 사용될 수 있다[1][7]. 이러한 하나의 예로써 passive star형 광상호연결망에서는 송신노드에서 정해진 파장을 사용하여 패킷을 방송하면 해당 수신노드에서 선택적으로 처리하는 방식(broadcast-and-select)으로 노드간의 패킷전달이 이루어지며, 이를 위해 송신기(transmitter)나 혹은 수신기(receiver)중의 적어도 하나는 파장을 바꾸어가면서 사용할 수 있도록 가변적(tunable)이어야 한다[6].

본 논문에서는 모든 노드가 임의의 파장에 대하여 가변적인 송신기(tunable transmitter)와 고정적인 수신기(fixed-wavelength receiver)를 각각 한 개씩 가지고 있다고 가정하고, 파장분할다중화(wavelength division multiplexing)에 의해 동시에 사용할 수 있는 파장의 개수  $k$ 가 전체 노드의 개수  $N$ 에 비하여 상대적으로 매우 작다고 가정한다. 또한, 모든 패킷의 길이가 일정하여 어떤 파장을 사용하든지 송신노드와 수신노드간의 실제 거리에 관계없이 임의로 주어진 단위시간(unit time)내에 하나의 패킷 송수신이 완료된다고 할 때, 송신기의 파장을 다른 파장으로 변경하는 데에 걸리는 시간 즉, 파장변경에 따른 지연시간(tuning delay)을  $\delta$ 로 나타내기로 한다.

모든 노드간의 패킷전송이 이루어지려면 송신노드가 패킷을 보내기 전에 미리 송신기의 파장을 수신노드의

고정된 파장으로 변경해야 한다. 이 때, 송신기의 파장을 다른 파장으로 변경하는 동안에는 패킷을 보낼 수 없기 때문에 파장변경에 따른 지연시간을 최소화하기 위하여 송신노드의 입장에서는 가급적이면 특정 파장으로 보낼 패킷들을 전부 보낸 뒤에 다른 파장으로 변경하는 것이 필요하다. 또한, 파장분할다중화(WDM)에 의해 사용할 수 있는 파장의 개수는 제한되어 있으므로 같은 파장을 서로 다른 노드간의 패킷전달을 위하여 재사용해야만 한다. 송수신 스케줄(transmission schedules)이란 송신노드와 수신노드간의 패킷 전송시 사용할 파장을 서로 일치시키기 위해 언제 어떤 파장으로 무슨 패킷을 전송할 것인지에 대한 일종의 시간표와 같으며, 스케줄링 알고리즘에 따라 같은 패턴의 패킷들을 전송함에 있어 스케줄의 길이(length)가 달라질 수 있다. 따라서, 최적의 송수신 스케줄을 구하는 문제는 곧 파장변경에 따른 지연시간과 노드간의 패킷전송시간을 최대한 서로 중첩시키기 위한 스케줄링 알고리즘을 찾는 문제와 같다.

All-to-all 방송을 위한 송수신 스케줄을 구하는 문제는 방송시 모든 노드가 자기 자신에게도 패킷을 전송할 것인지의 여부에 따라 크게 두가지 경우로 나누어 볼 수 있다. 우선 모든 노드가 자기 자신에게도 패킷을 전송하면 한번의 all-to-all 방송을 위해  $N^2$ 개의 패킷이 전송되어야 한다. Pieris와 Sasaki[5]는  $N$ 이  $k$ 로 나누어 떨어진다고 가정하여 스케줄링 알고리즘을 제시하고, 임의의  $\delta$ 의 값에 대하여 송수신 스케줄의 길이는 항상  $\max \left\{ \delta + \frac{N^2}{k}, N\sqrt{\delta} \right\}$ 와  $\max \left\{ \delta + \frac{N^2}{k}, k\delta + N - \frac{N}{k} + \frac{N^2}{k^2} \right\}$ 사이에 있음을 증명하였다. 그 뒤, [2]에서는 [5]에서 보여준 스케줄의 길이가 최소값일 때 최적(optimal)임을 증명하였다. 또

한, [3]에서는  $N$ 이  $k$ 로 나누어 떨어지지 않는 경우를  $\delta$ 의 범위에 따라 크게 세가지 경우로 나누어 각각 최적의 송수신 스케줄링 알고리즘을 제안하였으나, 일부 경우에는 최적은 아니지만 스케줄의 길이가 최적 길이의  $\frac{13}{12}$ 배까지 근접한다는 것을 보여 주었다.

만일 모든 노드가 자기자신에게 패킷을 전송하지 않는다고 가정하면 한번의 all-to-all 방송을 위해 전송될 패킷의 개수는  $N^2$ 이 아니라  $N(N-1)$ 으로 줄어들게 된다. [4]에서는 앞에서와 마찬가지로  $N$ 이  $k$ 로 나누어 떨어진다고 가정하여 스케줄링 알고리즘을 제시하고, 여러 번의 all-to-all 방송이 반복적으로 이루어질 때 최적 송수신 스케줄의 주기(cycle length)는  $\max\left\{\frac{N(N-1)}{k}, k\delta + N - 1\right\}$ 임을 증명하였다. 한편, [8]에서는  $\frac{N}{k} = C$ 일 때 모든 노드에서 일정한 간격으로 패킷을 수신할 수 있는 스케줄링 알고리즘을 제안하고, 그러한 최적 송수신 스케줄의 주기는  $\max\left\{C(N-1), \left\lceil \frac{\delta}{C-1} + \frac{1}{C-1} \left\lceil \frac{C-1}{k} \right\rceil \right\rceil\right\}$ 임을 증명하였다.

본 논문에서는 모든 노드가 자기자신에게 패킷을 전송하지 않는다는 가정하에서 임의로 주어진  $N, k, \delta$ 의 값에 대하여 all-to-all 방송을 위한 최적 송수신 스케줄의 주기는  $\max\left\{\left\lceil \frac{N}{k} \right\rceil(N-1), k\delta + N - 1\right\}$ 임을 증명한 뒤에,  $N-1$ 이  $k$ 로 나누어 떨어질 때 최적 송수신 스케줄을 구하는 알고리즘을 제안하였다. 특히, 본 논문에서 제안된 스케줄링 알고리즘은 [4]에서 보여준 알고리즘과는 근본적으로 다를 뿐만 아니라, 임의로  $N, k, \delta$ 의 값이 주어지는 경우에도 사용할 수 있다. 단,  $N-1$ 이  $k$ 로 나누어 떨어지지 않는 경우에는  $\delta$ 의 값이 최적 송수신 스케줄의 주기에 전혀 영향을 주지 않는 조건 즉,  $\left\lceil \frac{N}{k} \right\rceil(N-1) > k\delta + N - 1$ 을 만족할 때에만 최적 송수신 스케줄을 구할 수 있다. 그러나, 이러한 제약조건을  $\delta$ 에 관한 식으로 정리해 보면, 예를 들어  $N=1000, k=10$ 일 때  $\delta \leq 9890.1$ 이듯이,  $k$ 가  $N$ 에 비하여 상대적으로 매우 작은 경우에

는 거의 모든 시스템에서 최적 송수신 스케줄링 구할 수 있다는 데에 의미가 있다.

본 논문의 구성은 다음과 같다. 다음 장에서는 all-to-all 방송을 위한 최적 송수신 스케줄의 주기를 정의한 뒤에, 3장에서는  $N-1$ 이  $k$ 로 나누어 떨어지는 경우에 최적 송수신 스케줄을 구하는 알고리즘을 설명한다. 4장에서는 앞에서 설명한 알고리즘을 확장하여 임의로 주어진  $N, k$ 의 값에 대하여  $\left\lceil \frac{N}{k} \right\rceil(N-1) > k\delta + N - 1$ 일 때 최적임을 증명하고, 결론을 내리기 전에 [4]의 스케줄링 알고리즘과 비교하여 분석한다.

## 2. 최적 송수신 스케줄의 주기

앞으로 본 논문에서는 모든 노드가 자기자신에게 패킷을 전송하지 않는다는 가정하에서의 all-to-all 방송을 단순히 '방송'이라 부르기로 한다. 방송을 위한 최적 송수신 스케줄의 주기를 여러 번의 방송이 반복적으로 계속될 때 한번의 방송에 필요한 평균 시간을 의미한다. 여기에서는 모든 노드의 송신기들을 초기화하는 시간은 고려하지 않기로 한다. 그러나, 송신기가 방송을 끝내고 처음 사용했던 파장으로 초기화하는 데에 필요한 파장변경시간은 방송을 위한 최적 송수신 스케줄의 주기에 포함되어야 한다. 따라서, 실제 방송시 모든 패킷들에 대한 전송이 완료될 때까지 걸리는 시간이 최적 방송 주기보다 짧아질 수 있다.

임의로 주어진  $N, k$ 의 값에 대하여, 각각의 노드에 0부터  $N-1$ 까지의 번호를 부여하고 파장은  $w_i$  (단,  $0 \leq i \leq k-1$ )와 같이 첨자를 사용하여 나타내기로 하자. 이 때, 노드  $s$ 의 수신기가 사용하는 파장  $w_i$ 를  $i = s \bmod k$ 로 정의하면, 수신기의 파장이  $w_i$ 인 노드들의 집합  $R_i$ 의 원소의 개수는  $|R_i| =$

$$\left\lceil \frac{N}{k} \right\rceil \text{ 또는 } \left\lceil \frac{N}{k} \right\rceil - 1 \text{ 이다.}$$

[정리 1] 임의로 주어진  $N, k, \delta$ 의 값에 대하여 방송을 위한 최적 송수신 스케줄의 주기는 항상

$$\max\left\{\left\lceil \frac{N}{k} \right\rceil(N-1), k\delta + N - 1\right\} \text{ 보다}$$

크거나 같다.

(증명) 각각의 노드는 자기자신을 제외한 모든 다른 노드들로부터 정해진 과장으로  $N-1$ 개의 패킷을 전송받게 된다. 이 때, 같은 과장으로 동시에 2개 이상의 패킷이 전송될 수 없으므로 최적 송수신 스케줄의 주기는 과장  $w_j$ 를 사용하여 전송될 모든 패킷의 개수 즉,  $|R_j|(N-1)$ 보다 크거나 같다. 그런데, 집합  $R_j$ 의 정의에 의해

$$\max_{0 \leq j \leq k-1} (|R_j|) = |R_0| = \left\lceil \frac{N}{k} \right\rceil$$

이므로 최적 송수신 스케줄의 주기는 항상

$$\left\lceil \frac{N}{k} \right\rceil (N-1)$$

보다 크거나 같다. 한편, 송신기를 처음에 사용했던 과장으로 초기화하는 것까지 감안한다면 모든 노드들은 한번의 방송을 위하여 과장을  $k$ 번 변경한다. 따라서 임의의 노드에 대하여 최적 송수신 스케줄의 주기는  $k$ 번의 과장변경에 따른 지연시간에 자기 자신을 제외한 다른 노드로 보낼  $N-1$ 개의 패킷전송에 필요한 시간을 더한 값 즉,  $k\delta + N-1$ 보다 크거나 같다. ■

### 3. 최적 송수신 스케줄링 알고리즘

만일  $N-1$ 이  $k$ 로 나누어 떨어지면  $N-1 = kC$ 를 만족하는 정수  $C$ 가 존재하고, 집합  $R_i$ 의 정의에 의해  $|R_0| = C+1$ 이고  $|R_i| = C$  (단,  $1 \leq i \leq k-1$ )이다. 따라서, 집합  $R_0$ 에 속하는 노드들은 과장  $w_0$ 을 사용하여  $C$ 개의 패킷을 전송하고나서  $\delta$ 만큼의 과장변경시간이 지나면 과장  $w_1$ 을 사용하여  $C$ 개의 패킷을 전송하는 방식으로 반복하여 마지막에는 과장  $w_{k-1}$ 을 사용하여  $C$ 개의 패킷을 전송한 다음  $\delta$ 만큼의 과장변경시간이 지나 과장  $w_0$ 을 사용할 수 있는 상태가 된다. 그러나, 그 밖의 다른 노드들의 경우, 즉 집합  $R_i$  (단,  $1 \leq i \leq k-1$ )에 속하는 노드들은 과장  $w_0$ 을 사용할 때와 과장  $w_i$ 를 사용할 때에만 각각  $C+1$ 개와  $C-1$ 개의 패킷을 전송하는 것을 제외

하고는 집합  $R_0$ 에 속하는 노드들의 경우와 마찬가지로이다.

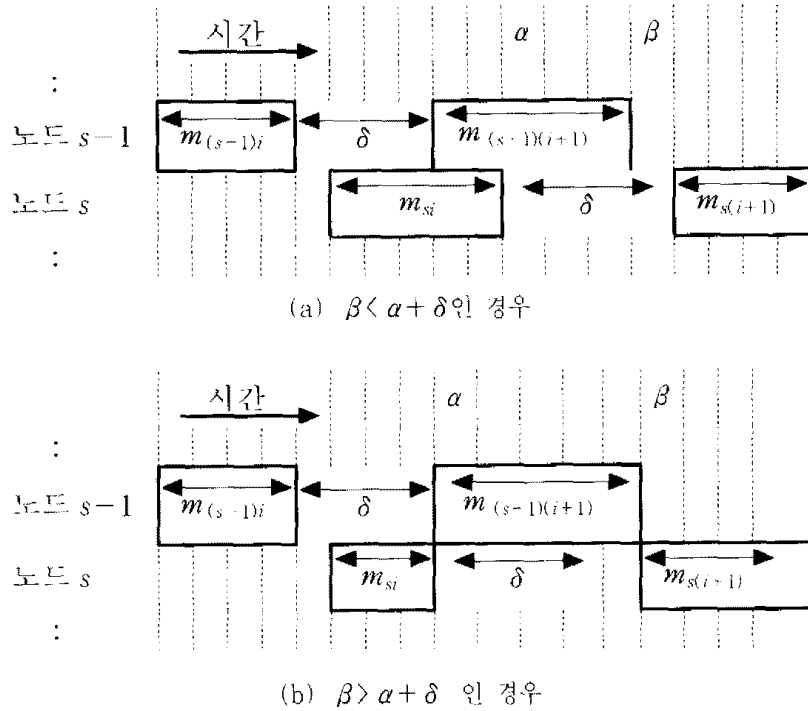
[보조정리 1] 만일  $k | N-1$ 이면 모든 노드가 서로 독립적으로 패킷을 전송한다고 할 때, 각 노드의 송수신 스케줄의 길이는 항상 일정하다.

(증명) 집합  $R_0$ 에 속하는 노드들의 경우에는  $kC + k\delta = k\delta + N-1$ 이고, 그 밖의 집합  $R_i$  (단,  $1 \leq i \leq k-1$ )에 속하는 노드들의 경우에는  $(C+1) + (k-2)C + (C-1) + k\delta = k\delta + N-1$ 이므로 항상 일정하다. ■

노드 0에서 전송할 패킷들을 스케줄함에 있어서  $w_0, w_1, \dots, w_{k-1}$ 의 순서대로 과장  $w_0$ 을 사용하여 보낼 패킷들을 모두 전송하고 과장변경시간  $\delta$ 가 지난 뒤 바로 다음 과장  $w_1$ 을 사용하여 보낼 패킷들을 모두 전송하는 식으로 나중에는 과장  $w_{k-1}$ 을 사용하여 보낼 패킷들을 모두 전송한 뒤에 과장변경시간  $\delta$ 가 지나면 스케줄이 끝나도록 한다. 또한, 과장  $w_0$ 을 사용하여 전송할 패킷들을 스케줄함에 있어서는  $1 \leq s \leq N-1$ 일 때 노드  $s-1$ 에서 보낼 패킷들을 모두 전송하고 나면 바로 이어서 노드  $s$ 에서 패킷들을 전송하기 시작하기로 한다. 이 때, 과장  $w_i$  (단,  $1 \leq i \leq k-1$ )를 사용하여 전송할 패킷들에 대하여 송수신 스케줄이 이미 결정되어 있다고 가정하면, 노드  $s$ 에서 과장  $w_{i+1}$ 을 사용하여 전송할 패킷들을 스케줄하기 위하여 (그림 1)과 같이 노드  $s$ 에서 과장  $w_i$ 을 사용하여 전송할 패킷들을 모두 보낸 시점  $\alpha$ 에 과장변경시간  $\delta$ 를 더한 시점  $\alpha + \delta$ 와 노드  $s-1$ 에서 과장  $w_{i+1}$ 을 사용하여 전송할 패킷들을 모두 보낸 시점  $\beta$ 를 비교하여 더 늦은 시점  $\gamma = \max\{\alpha + \delta, \beta\}$ 부터 패킷들을 보내기 시작한다. 이러한 방식의 스케줄링 알고리즘은 (그림 2)에 제시하였으며, 그에 따른 송수신 스케줄에서는 같은 과장을 사용하여 동시에 2개 이상의 패킷이 전송될 수 없을 뿐만 아니라 각

노드에 장착된 송신기의 파장변경시간을 언제나 보장해 준다. 방송을 위한 최적 송수신 스케줄의 주기를  $L$ 로 나타내기로 한다. 일반적으로,  $N \times k$  행렬  $M = (m_{si})$ 의 원소  $m_{si}$ 를 노드  $s$ 가 파장  $w_i$ 를 사용하여

전송할 패킷의 개수라고 하면,  $s \bmod k = i$ 일 때  $m_{si} = |R_i| - 1$ 이고 그 밖의 경우에는  $m_{si} = |R_i|$ 이다.



(그림 1) 패킷의 전송시점  $\gamma = \max\{\alpha + \delta, \beta\}$  선택  
 (Fig. 1) Choice of a transmission time,  $\gamma = \max\{\alpha + \delta, \beta\}$

```

Transmission Scheduling Algorithm for  $k \mid N-1$ 
Begin
0    $p_0 \leftarrow 1, T(s, t) \leftarrow NULL (0 \leq s \leq N-1, 1 \leq t \leq 2L)$ 
1   For  $s=0$  to  $N-1$  do
2     For  $i=0$  to  $k-1$  do
3       If  $((p_{i-1} + \delta > p_i) \text{ and } (i \neq 0))$  then  $p_i \leftarrow p_{i-1} + \delta$ 
4       For  $j=0$  to  $|R_i|-1$  do
5         If  $((s \equiv i \pmod k) \text{ and } (s \neq i + kj)) \text{ or } (s \not\equiv i \pmod k)$  then {
6            $T(s, p_i) \leftarrow i + kj$ 
7            $p_i \leftarrow p_i + 1$ 
8         }
End
    
```

(그림 2)  $k \mid N-1$  일 때 송수신 스케줄링 알고리즘  
 (Fig. 2) A transmission scheduling algorithm for  $k \mid N-1$

일단의 노드  $s$ 가 파장  $w_0$ 를 사용하여 전송할 패킷들을 처음 보내기 시작한 때부터  $k$ 개의 파장을 전부 사용하여 전송할 패킷들을 모두 보낸 다음 처음에 사용했던 파장  $w_0$ 를 다시 사용할 수 있도록 파장변경을 마칠 때까지 걸린 시간을 노드  $s$ 에 대한 송수신 스케줄의 길이라고 정의한다. 같은 의미에서, 노드에 관계없이 파장  $w_i$ 를 사용하여 전송될 패킷들이 처음 보내지기 시작한 때부터 마지막 남은 패킷이 보내질 때까지 걸린 시간을 파장  $w_i$ 에 대한 송수신 스케줄의 길이라고 정의한다.

[보조정리 2] 만약  $k \mid N-1$ 이면 (그림 2)에서 제시한 알고리즘에 의한 송수신 스케줄에서 각각의 노드에 대한 송수신 스케줄의 길이는  $k\delta + N-1$ 로 모두 같다.

(증명) 알고리즘의 가정에 의해 노드 0에 대한 스케줄

의 길이는  $k(C+\delta)$ 이다. 노드  $s$ 에서 파장  $w_i$ 를 사용하여 전송할 패킷들을 모두 보낸 시점과 노드  $s-1$ 에서 파장  $w_{i+1}$ 을 사용하여 전송할 패킷들을 모두 보낸 시점을 각각  $\alpha_{si}$ 과  $\beta_{(s-1)(i+1)}$ 이라 하자. 이 때, 노드  $s$ 가 파장  $w_{i+1}$ 을 사용하여 전송할 패킷들을 보내기 시작하는 시점을 선택하기 위해 계산할 두 시점간의 차이를  $\theta_{s(i+1)} = (\alpha_{si} + \delta) - \beta_{(s-1)(i+1)}$  (단,  $1 \leq s \leq N-1$ )이라 정의하면,  $\theta_{s(i+1)}$ 의 부호에 따라 패킷을 전송할 시점이 결정된다. 예를 들어 (그림 1)에서  $\theta_{si} = 1$ 일 때 (a)의 경우에는  $\theta_{s(i+1)} = 1$ 이고 (b)의 경우에는  $\theta_{s(i+1)} = -1$ 이 된다. 알고리즘의 가정에 의해 노드 0에 대한 스케줄과 파장  $w_0$ 에 대한 스케줄이 결정되면  $\theta_{0i} = 0$  (단,  $0 \leq i \leq k-1$ )

	0	1	2	3	...	k-2	k-1
k개	C	C	C	C	...	C	C
	C+1	C-1	C	C	...	C	C
	C+1	C	C-1	C	...	C	C
	C+1	C	C	C-1	...	C	C
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	C+1	C	C	C	...	C-1	C
k개	C+1	C	C	C	...	C	C-1
	C	C	C	C	...	C	C
	C+1	C-1	C	C	...	C	C
	C+1	C	C-1	C	...	C	C
	C+1	C	C	C-1	...	C	C
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
k개	C+1	C	C	C	...	C-1	C
	C+1	C	C	C	...	C	C-1
	C	C	C	C	...	C	C
	C+1	C	C	C	...	C-1	C
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	C+1	C	C	C	...	C	C-1
k개	C	C	C	C	...	C	C
	C+1	C-1	C	C	...	C	C
	C+1	C	C-1	C	...	C	C
	C+1	C	C	C-1	...	C	C
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	C+1	C	C	C	...	C-1	C
k개	C+1	C	C	C	...	C	C-1
	C	C	C	C	...	C	C
	C+1	C-1	C	C	...	C	C
	C+1	C	C-1	C	...	C	C
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	C+1	C	C	C	...	C-1	C

(a) 행렬  $M$

	0	1	2	3	...	k-2	k-1
k개	0	0	0	0	...	0	0
	0	1	0	0	...	0	0
	0	2	2	1	...	1	1
	0	1	2	2	...	1	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0	1	1	1	...	2	1
k개	0	1	1	1	...	2	2
	0	0	0	0	...	0	1
	0	1	0	0	...	0	0
	0	2	2	1	...	1	1
	0	1	2	2	...	1	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
k개	0	1	1	1	...	2	1
	0	1	1	1	...	2	2
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0	0	0	0	...	0	1
	0	1	0	0	...	0	0
	0	2	2	1	...	1	1
k개	0	1	2	2	...	1	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0	1	1	1	...	2	1
	0	1	1	1	...	2	2
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0	0	0	0	...	0	1

(b) 행렬  $\Theta$

(그림 3)  $N-1 \mid k$ 일 때의 행렬  $M, \Theta$

(Fig. 3) Matrices  $M, \Theta$  for  $N-1 \mid k$

이와  $\theta_{s_i} = 0$  (단,  $0 \leq s \leq N-1$ )이다. 이때,  $\delta + m_{(s-1)(i+1)} + \theta_{s(i+1)} - \theta_{s_i} + m_{s_i} + \delta$  이라는 사실로부터  $\theta_{s_i}$ 와  $\theta_{s(i+1)}$  사이의 관계식  $\theta_{s(i+1)} = \theta_{s_i} + m_{s_i} - m_{(s-1)(i+1)}$ 을 적용하면 두 개의  $N \times k$  행렬  $M$ ,  $\Theta$ 는 다음과 같다.

그런데, 임의의 노드  $s$ 가 파장  $w_j$ 를 사용하여 패킷을 전송할 때 항상  $\theta_{s_i} \geq 0$ 이면 알고리즘의 진행과정에서 항상  $\gamma = \alpha + \delta$ 이라는 등식이 성립한다. 즉, 하나의 파장을 사용하여 전송할 패킷들을 모두 보낸 다음 파장변경시간  $\delta$ 가 지나면 곧바로 다음 파장을 사용하여 전송할 패킷들을 보내기 시작하므로 임의의 노드  $s$ 에 대한 스케줄의 길이는  $k\delta + N - 1$ 로 모두 같다. ■

**[보조정리 3]** 만일  $k \mid N-1$ 이면 (그림 2)에서 제시한 알고리즘에 의한 송수신 스케줄에서 파장  $w_0$ 에 대한 송수신 스케줄의 길이는  $\lceil \frac{N}{k} \rceil (N-1)$ 이다.

**(증명)** 알고리즘에서 가장한대로 스케줄하면 결과적으로 노드 0부터 노드  $N-1$ 까지 번호순서에 따라 각각의 노드에서 파장  $w_0$ 를 사용하여 전송할 패킷들을 모두 보낸 다음 곧바로 다음 노드가 패킷전송을 시작하게 되므로, 파장  $w_0$ 에 대한 송수신 스케줄의 길이는 곧 파장  $w_0$

을 사용하여 전송할 모든 패킷의 개수 즉,  $\lceil R_0 \rceil (N-1)$ 와 같다. 그런데,  $k \mid N-1$  이면  $\lceil R_0 \rceil = C+1 = \lceil \frac{N}{k} \rceil$  이므로  $\lceil R_0 \rceil (N-1) = \lceil \frac{N}{k} \rceil (N-1)$ 이다. ■

예를 들어,  $N=7, k=3$ 이고  $\delta=4$ 인 경우에 (그림 3)은 (그림 2)에서 제시한 알고리즘에 의한 송수신 스케줄 결과로서, 각 행의 맨처음 번호는 노드 번호를 나타내고 각 열의 맨 위에 있는 숫자는 시간의 흐름을 나타낸다. 따라서,  $T(s, p_i) = q$ 이면 노드  $s$ 가 시간이  $p_i$ 일 때 노드  $q$ 로 파장  $w_j$  (단,  $q \bmod k = j$ )를 사용하여 패킷을 전송하고,  $T(s, p_i)$ 의 값이 정의되지 않으면 노드  $s$ 의 송신기가 파장을 변경하는 중이거나 또는 전송에 사용할 파장을 다른 노드에서 사용하고 있음을 의미한다.

이제 최적의 송수신 스케줄을 구하는 방법을 알아보기로 하자. (그림 2)의 알고리즘으로 구한 송수신 스케줄 결과를 이용하여, 노드  $s$ 에서 패킷을 전송하는 시점이 최적 송수신 스케줄의 주기  $L$ 보다 큰 경우에는  $L$ 만큼 앞당겨 보기로 하자. 즉, 앞에서 주어진 예제의 경우에  $L=18$ 이므로 (그림 4)의 스케줄 결과에서 전송 시점이 1부터 18까지인 부분은 남겨두고  $L$ 보다 큰 부분은 (그림 5)에서 보듯이 왼쪽으로  $L$ 만큼 평행 이동시킨 결과를 얻을 수 있다.

일반적으로  $k \mid N-1$ 일 때 방송을 위한 최적 송수신 스케줄을 구하기 위한 스케줄링 알고리즘은 (그림 6)에 제시되어 있다. (그림 2)에서 제시한 알고리즘과

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
0	3	6					1	4				2	5																						
1			0	3	6				4					2	5																				
2					0	3	6				1	4				5																			
3								0	6				1	4				2	5																
4										0	3	6				1				2	5														
5													0	3	6			1	4					2											
6																0	3					1	4					2	5						

(그림 4)  $N=7, k=3$ 이고  $\delta=4$ 일 때의 송수신 스케줄  
(Fig. 4) A transmission schedule for  $N=7, k=3$  and  $\delta=4$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34		
0	3	6					1	4					2	5																						
1			0	3	6					4					2	5																				
2		5					0	3	6				1	4					5																	
3			2	5					0	6					1	4							2	5												
4					2	5					0	3	6					1						2	5											
5			1	4				2						0	3	6							1	4				2								
6				1	4						2	5					0	3						1	4					2	5					

(그림 5)  $N=7, k=3$ 이고  $\delta=4$ 일 때의 수정된 송수신 스케줄  
 (Fig. 5) A modified transmission schedule for  $N=7, k=3$  and  $\delta=4$

```

Optimal Transmission Scheduling Algorithm for  $k | N-1$ 
Begin
0       $L \leftarrow \max \left\{ \left\lceil \frac{N}{k} \right\rceil (N-1), k\delta + N - 1 \right\}$ 
1       $p_0 \leftarrow 1, T(s, t) \leftarrow NULL (0 \leq s \leq N-1, 1 \leq t \leq L)$ 
2      For  $s=0$  to  $N-1$  do
3          For  $i=0$  to  $k-1$  do
4              If  $((p_{i-1} + \delta > p_i)$  and  $(i \neq 0))$  then  $p_i \leftarrow p_{i-1} + \delta$ 
5              For  $j=0$  to  $|R_i|-1$  do
6                  If  $((s \equiv i \pmod k)$  and  $(s \neq i + kj))$  or  $(s \not\equiv i \pmod k)$  then {
7                      If  $(p_i > L)$  then
8                           $T(s, p_i - L) \leftarrow i + kj$ 
9                      else
10                          $T(s, p_i) \leftarrow i + kj$ 
11                      $p_i \leftarrow p_i + 1$ 
12                 }
End
    
```

(그림 6)  $k | N-1$ 일 때 최적 송수신 스케줄링 알고리즘  
 (Fig. 6) Optimal transmission scheduling algorithm for  $k | N-1$

비교해보면 0행에 최적 방송 주기  $L$ 을 직접 계산하는 부분이 추가되고, 7~10행에서  $L$ 보다 큰 전송시점  $p_i$ 를  $p_i - L$ 로 조정하는 부분이 바뀌어 있음을 알 수 있다.

[정리 2] 만일  $k | N-1$ 이면 (그림 6)의 알고리즘에 따른 송수신 스케줄의 주기는 최적이다.

(증명) 만일  $\left\lceil \frac{N}{k} \right\rceil (N-1) \leq k\delta + N - 1$ 이면 [정

리 1]로부터  $L = k\delta + N - 1$ 이고, 앞의 [보조정리 2]에 의해 모든 노드에 대한 스케줄의 길이가  $k\delta + N - 1$ 이므로 전송시점이  $L$ 보다 큰 부분을  $L$ 만큼  $L$  왼쪽으로 평행이동시켰을 때 전송시점이 보다 작은 부분과 겹치는 부분이 전혀 생기지 않는다. 반대로

$\left\lceil \frac{N}{k} \right\rceil (N-1) > k\delta + N - 1$ 이면 [정리 1]

로부터  $L = \left\lceil \frac{N}{k} \right\rceil (N-1)$ 이고, 앞의 [보



수정된 3에 의해 과장  $w_0$ 에 대한 스케줄링 길이가  $\lceil \frac{N}{k} \rceil (N-1)$ 이므로 전송시점이  $L$ 보다 큰 부분을  $L$ 만큼 왼쪽으로 평행이동시켰을 때 전송시점이  $L$ 보다 작은 부분과 겹치는 부분이 전혀 생기지 않는다. 따라서, (그림 6)의 알고리즘에 따른 송수신 스케줄의 주기는 최적이다. ■

앞의 예제와 마찬가지로  $N=7, k=3$ 일 때,  $\delta=3$ 이면 (그림 7)에서 보듯이  $\delta$ 의 값에 관계없이  $L = \lceil \frac{7}{3} \rceil (7-1) = 18$ 이므로, (그림 5)와 비교해보면 주기와 과장  $w_0$ 에 대한 스케줄은 같으나 그 밖의 과장으로 전송될 패킷들의 전송시점이  $\delta$ 의 값이 줄어든 만큼 앞당겨진 것을 알 수 있다. 같은 조건에서  $\delta=5$ 이면 (그림 8)에서 보듯이  $L = 3 \times 5 + 7 - 1$

일치므로  $\delta$ 의 값에 의해 전파 방송 주기가 결정되며, (그림 5)와 비교해보면 과장  $w_0$ 에 대한 스케줄은 같으나 그 밖의 과장으로 전송될 패킷들의 전송시점이  $\delta$ 의 값이 늘어남 만큼 뒤로 늦춰진 것을 알 수 있다.

#### 4. 일반화된 최적 송수신 스케줄링 알고리즘

앞 장에서 제시한  $k \mid N-1$ 일 때의 최적 송수신 스케줄을 구하는 알고리즘을 임의로 주어진  $N, k, \delta$ 의 값에 대하여 적용하기 위하여 확장해 보기로 하자. 정수  $j$  (단,  $1 \leq j \leq k$ )를  $N=Ck+j$ 로 정의하면  $j=1$ 일 때  $k \mid N-1$ 이고,  $j=k$ 이면  $k \mid N$ 이다. 또한, 집합  $R_j$ 의 정의에  $|R_j| = \dots =$  의하여  $|R_0| = \dots = |R_{j-1}| = C+1$  이고  $|R_{k-1}| = C$ 이다. 이제,  $k \nmid N-1$ 인 경우를 설명하기 위하여  $j \neq 1$  즉,  $2 \leq j \leq k$ 이라고 가정한다.

임의로  $N, k, \delta$ 의 값이 주어졌을 때 방송을 위한

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
0	3	6				1	4				2	5							
1			0	3	6				4				2	5					
2						0	3	6				1	4					5	
3	2	5							0	6				1	4				
4			2	5							0	3	6					1	
5		1	4				2							0	3	6			
6				1	4				2	5								0	3

(그림 7)  $N=7, k=3$ 이고  $\delta=3$  일 때의 최적 송수신 스케줄  
(Fig. 7) An optimal transmission schedule for  $N=7, k=3$  and  $\delta=3$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	3	6						1	4						2	5					
1			0	3	6					4							2	5			
2						0	3	6					1	4							5
3		2	5						0	6					1	4					
4				2	5						0	3	6								1
5	1	4						5						0	3	6					
6			1	4						2	5						0	3			

(그림 8)  $N=7, k=3$ 이고  $\delta=5$ 일 때의 최적 송수신 스케줄  
(Fig. 8) An optimal transmission schedule for  $N=7, k=3$  and  $\delta=5$

스케줄링 알고리즘의 기본 아이디어는 앞 장에서 제시한 스케줄링 알고리즘과 같다. 즉, 노드  $s$ 에 대한 스케줄과 파장  $w_0$ 에 대한 스케줄을 가정하고 노드의 번호순서와 파장의 사용순서  $w_1, w_2, \dots, w_{k-1}$ 에 따라 패킷의 전송시점을 결정한 다음, 최적 송수신 스케줄의 주기  $L$ 을 계산하여 전송시점이  $L$ 보다 큰 부분을  $L$ 만큼 왼쪽으로 평행이동한다. 자세한 설명을 위하여 앞

장에서 정의한대로 두 개의  $N \times k$  행렬  $M = (m_{st})$ 와  $\Theta = (\theta_{st})$ 를 구해보면 다음과 같다.

(그림 9)에서 행렬  $\Theta$ 를 자세히 살펴보면 노드 0을 제외하고  $s \bmod k = 0$ 이거나  $s \bmod k = 1$ 인 노드  $s$ 에 대하여 행렬  $\Theta$ 의 원소들이 -1의 값을 갖게 된다. 이 때,  $\theta_{st} = -1$ 이면 노드  $s$ 의 송신기의 파장을  $w_j$ 로 변경하는 작업이 완료되었음에도 불구하고

		j 개			k-j 개							
j 개		C	C+1	C+1	...	C+1	C+1	C	C	...	C	C
		C+1	C	C+1	...	C+1	C+1	C	C	...	C	C
		C+1	C+1	C	...	C+1	C+1	C	C	...	C	C
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		C+1	C+1	C+1	...	C	C+1	C	C	...	C	C
		C+1	C+1	C+1	...	C+1	C	C	C	...	C	C
k-j 개		C+1	C+1	C+1	...	C+1	C+1	C-1	C	...	C	C
		C+1	C+1	C+1	...	C+1	C+1	C	C-1	...	C	C
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		C+1	C+1	C+1	...	C+1	C+1	C	C	...	C-1	C
		C+1	C+1	C+1	...	C+1	C+1	C	C	...	C	C-1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		C	C+1	C+1	...	C+1	C+1	C	C	...	C	C
		C+1	C	C+1	...	C+1	C+1	C	C	...	C	C
		C+1	C+1	C	...	C+1	C+1	C	C	...	C	C
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		C+1	C+1	C+1	...	C	C+1	C	C	...	C	C
		C+1	C+1	C+1	...	C+1	C	C	C	...	C	C
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		C+1	C+1	C+1	...	C+1	C+1	C-1	C	...	C	C
		C+1	C+1	C+1	...	C+1	C+1	C	C-1	...	C	C
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		C+1	C+1	C+1	...	C+1	C+1	C	C	...	C-1	C
		C+1	C+1	C+1	...	C+1	C+1	C	C	...	C	C-1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		C	C+1	C+1	...	C+1	C+1	C	C	...	C	C
		C+1	C	C+1	...	C+1	C+1	C	C	...	C	C
		C+1	C+1	C	...	C+1	C+1	C	C	...	C	C
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		C+1	C+1	C+1	...	C	C+1	C	C	...	C	C
		C+1	C+1	C+1	...	C+1	C	C	C	...	C	C
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(a) 행렬  $M$

		j 개			k-j 개							
j 개		0	0	0	...	0	0	0	0	...	0	0
		0	0	-1	...	-1	-1	0	0	...	0	0
		0	1	1	...	0	0	1	1	...	1	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	0	0	...	1	0	1	1	...	1	1
		0	0	0	...	1	1	1	1	...	1	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	0	0	...	0	1	2	1	...	1	1
		0	0	0	...	0	0	2	2	...	1	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	0	0	...	0	0	1	1	...	2	1
		0	0	0	...	0	0	1	1	...	2	2
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	-1	-1	...	-1	-1	0	0	...	0	1
		0	0	-1	...	-1	-1	0	0	...	0	0
		0	1	1	...	0	0	1	1	...	1	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	0	0	...	1	0	1	1	...	1	1
		0	0	0	...	1	1	1	1	...	1	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	0	0	...	0	1	2	1	...	1	1
		0	0	0	...	0	0	2	2	...	1	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	0	0	...	0	0	1	1	...	2	1
		0	0	0	...	0	0	1	1	...	2	2
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	-1	-1	...	-1	-1	0	0	...	0	1
		0	0	-1	...	-1	-1	0	0	...	0	0
		0	1	1	...	0	0	1	1	...	1	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		0	0	0	...	1	0	1	1	...	1	1
		0	0	0	...	1	1	1	1	...	1	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(b) 행렬  $\Theta$

(그림 9)  $N \times k$  행렬  $M, \Theta$

(Fig. 9)  $N \times k$  Matrices of  $M, \Theta$

다른 노드에서 파장  $w_i$ 를 사용하고 있으므로 1만큼 기다려야 한다는 것을 의미한다.

예를 들어,  $N=8, k=3$ 이고  $\delta=4$ 인 경우에 (그림 10)은 (그림 6)에서 제시한 알고리즘에 의한 송수신 스케줄 결과를 보여주고 있다. 이 때, (그림 10)의 7번째 열에서 보듯이 노드 0에서 파장  $w_1$ 을 사용하여 패킷을 전송하기 시작하는 부분과 노드 7에서 파

장  $w_1$ 을 사용하여 마지막 패킷을 전송하는 부분이 서로 겹쳐지는 문제가 발생하므로 스케줄을 조정할 필요가 있다. 이러한 문제가 발생하는 근본적인 원인은 알고리즘에서 가정한 노드 0에 대한 스케줄이 일반적으로  $s \bmod k=0$ 를 만족하는 노드  $s$ 에 대한 스케줄과 일치하지 않기 때문이다. 따라서, 노드 0에 대한 스케줄을 (그림 11)의 5행에서 보듯이  $k \nmid N-1$ 일 때

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	3	6					1	4	7	↔					2	5					
1			0	3	6					4	7	↔					2	5			
2						0	3	6					1	4	7						5
3		2	5						0	6						1	4	7			
4				2	5					0	3	6							1	7	
5	4	7					2						0	3	6						1
6			1	4	7					2	5						0	3			
7						1	4					2	5						0	3	6

(그림 10)  $N=8, k=3$ 이고  $\delta=4$ 일 때의 송수신 스케줄 결과  
(Fig. 10) A transmission schedule for  $N=8, k=3$  and  $\delta=4$

**Generalized Optimal Transmission Scheduling Algorithm**

Begin

```

0       $L \leftarrow \max \left\{ \left\lceil \frac{N}{k} \right\rceil (N-1), k\delta + N - 1 \right\}$ 
1       $p_0 \leftarrow 1, T(s, t) \leftarrow NULL \ (0 \leq s \leq N-1, 1 \leq t \leq L)$ 
2      For  $s=0$  to  $N-1$  do
3          For  $i=0$  to  $k-1$  do
4              If  $((p_{i-1} + \delta > p_i)$  and  $(i \neq 0))$  then  $p_i \leftarrow p_{i-1} + \delta$ 
5              If  $((s=0)$  and  $(i=1)$  and  $(N-1 \bmod k \neq 0))$  then  $p_i \leftarrow p_{i-1} + 1$ 
6              For  $j=0$  to  $|R_i|-1$  do
7                  If  $((s=i \bmod k)$  and  $(s \neq i+kj))$  or  $(s \neq i \bmod k)$  then {
8                      If  $(p_i > L)$  then
9                           $T(s, p_i - L) \leftarrow i + kj$ 
10                     else
11                          $T(s, p_i) \leftarrow i + kj$ 
12                      $p_i \leftarrow p_i + 1$ 
13                 }
End
```

(그림 11) 일반화된 최적 송수신 스케줄링 알고리즘  
(Fig. 11) A generalized optimal transmission scheduling algorithm

각 0기 좌상  $w_1$  을 사용하여 패킷을 전송하기 시작하는 시점을 1만큼 늦추도록 조정한다.

**[보조정리4]** 임의로 주어진  $N, k, \delta$  의 값에 대하여, (그림 11)에서 제시한 알고리즘에 의한 송수신 스케줄에서 각각의 노드에 대한 송수신 스케줄의 길이는  $k\delta + N$ 이거나  $k\delta + N - 1$ 이다.

**(증명)** 알고리즘의 가정에 의해  $k$ 번의 지연시간외에 추가로 1만큼 늦추었으므로, 노드 0에 대한 스케줄의 길이는  $k\delta + C + 1 + (j-1)(C+1) + (k-j)C = k\delta + kC + j = k\delta + N$ 이다. 이러한 가정의 변화로 인하여 (그림 9)에서 제시한 행렬  $\Theta$ 의 0번째 행과  $s \bmod k = 0$ 인  $s$ 번째 행이 일치할 뿐만 아니라,  $s \bmod k = 0$  또는  $s \bmod k = 1$ 인 노드  $s$ 에 대한 스케줄의 길이는 모두  $k\delta + N$ 이다. 그러나, 그 밖의 다른 노드에 대한 스케줄의 길이는 앞장의 [보조정리 2]에 의하여 항상  $k\delta + N - 1$ 이다. ■

**[정리3]** 임의로 주어진  $N, k, \delta$  의 값에 대하여,

$\lfloor \frac{N}{k} \rfloor (N-1) > k\delta + N - 1$ 이면 (그림 11)의 알고리즘에 따른 송수신 스케줄의 주기는 최적이다.

**(증명)** 먼저  $\lfloor \frac{N}{k} \rfloor (N-1) \leq k\delta + N - 1$ 이면 [정리 1]로부터  $L = k\delta + N - 1$ 이고, [보조정리 4]에 의하면  $s \bmod k = 0$  또는  $s \bmod k = 1$ 인 노드  $s$ 에 대한 스케줄의 길이는 모두  $k\delta + N$ 이므로 최적 스케줄의 주기  $L$ 보다 커지게 된다. 반대로  $\lfloor \frac{N}{k} \rfloor (N-1) > k\delta + N - 1$ 이면 [정리 1]로부터  $L = \lfloor \frac{N}{k} \rfloor (N-1)$ 이고  $\lfloor \frac{N}{k} \rfloor (N-1) \geq k\delta + N$ 이므로, 전송시점이  $L$ 보다 큰 부분을  $L$ 만큼 왼쪽으로 평행이동 시켰을때 전송시점이  $L$ 보다 작은 부분과 겹치는 부분이 전혀 생기지 않는다. 또한, (그림 10)의 예제에서 보듯이 비록 겹치는 부분이 생기지 않더라도 같은 시간에 동일한 파장으로 2개 이상의 패킷을 전송하는 경우가 생기지 않으므로 (그림 11)의 알고리즘에 따른 송수신 스케줄의 주기는 최적이다. ■

본 논문에서 제안한 알고리즘과 [4]의 알고리즘의 특성을 서로 비교하기 위해  $k | N$ 이고  $\lfloor \frac{N}{k} \rfloor (N-1) > k\delta + N - 1$ 인 경우에 각각의 알고리즘을 사용하여 얻어진 송수신 스케줄을 비교한 결과 최적 방송 주기는 서로 같다는 것을 확인하였다. 그러나, 패킷 전송의 패턴을 자세히 살펴보면 여기에서 제안한 알고리즘에서는 동시에 전송되는 패킷의 개수가 시간의 호

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	4				1	5			2	6			3	7			
1		0	4				5				2	6			3	7	
2	7			0	4			1	5				6				3
3		7				0	4			1	5			2	6		
4			3	7				0				1	5			2	6
5	2	6			3	7			0	4				1			
6			2				3	7			0	4			1	5	
7	5			2	6				3				0	4			1

(그림 12)  $N=8, k=4$ 이고  $\delta=2$ 일 때 구한 송수신 스케줄  
(Fig. 12) Our transmission schedule for  $N=8, k=4$  and  $\delta=2$

같은 노드에서 파장  $w_1$ 을 사용하고 있으므로 (반경) 기다려야 한다는 것을 의미한다.

예를 들어,  $N=8, k=3$ 이고  $\delta=4$ 일 경우에 (그림 10)은 (그림 6)에서 제시한 알고리즘에 의한 송수신 스케줄 결과를 보여주고 있다. 이 때, (그림 10)의 7번째 열에서 보듯이 노드 0에서 파장  $w_1$ 을 사용하여 패킷을 전송하기 시작하는 부분과 노드 7에서 파

장  $w_1$ 을 사용하여 파장의 패킷을 전송하는 부분이 서로 겹쳐지는 문제가 발생하므로 스케줄을 조정할 필요가 있다. 이러한 문제가 발생하는 근본적인 원인은 알고리즘에서 가정한 노드 0에 대한 스케줄이 일반적으로  $s \bmod k=0$ 를 만족하는 노드  $s$ 에 대한 스케줄과 일치하지 않기 때문이다. 따라서, 노드 0에 대한 스케줄을 (그림 11)의 5행에서 보듯이  $k \nmid N-1$  일 때

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	3	6					1	4	7	↔					2	5					
1			0	3	6					4	7	↔					2	5			
2						0	3	6					1	4	7						5
3		2	5						0	6						1	4	7			
4				2	5						0	3	6						1	7	
5	4	7					2							0	3	6					1
6			1	4	7					2	5						0	3			
7						1	4					2	5						0	3	6

(그림 10)  $N=8, k=3$ 이고  $\delta=4$ 일 때의 송수신 스케줄 결과  
 (Fig. 10) A transmission schedule for  $N=8, k=3$  and  $\delta=4$

**Generalized Optimal Transmission Scheduling Algorithm**

Begin

0  $L \leftarrow \max \left\{ \left\lceil \frac{N}{k} \right\rceil (N-1), k\delta + N - 1 \right\}$

1  $p_0 \leftarrow 1, T(s, t) \leftarrow NULL (0 \leq s \leq N-1, 1 \leq t \leq L)$

2 For  $s=0$  to  $N-1$  do

3 For  $i=0$  to  $k-1$  do

4 If  $((p_{i-1} + \delta > p_i) \text{ and } (i \neq 0))$  then  $p_i \leftarrow p_{i-1} + \delta$

5 If  $((s=0) \text{ and } (i=1) \text{ and } (N-1 \bmod k \neq 0))$  then  $p_i \leftarrow p_{i-1} + 1$

6 For  $j=0$  to  $|R_i|-1$  do

7 If  $((s \equiv i \pmod k) \text{ and } (s \neq i + kj)) \text{ or } (s \not\equiv i \pmod k)$  then {

8 If  $(p_i > L)$  then

9  $T(s, p_i - L) \leftarrow i + kj$

10 else

11  $T(s, p_i) \leftarrow i + kj$

12  $p_i \leftarrow p_i + 1$

13 }

End

(그림 11) 일반화된 최적 송수신 스케줄링 알고리즘  
 (Fig. 11) A generalized optimal transmission scheduling algorithm

단, 0가 가장  $u_i$ 를 사용하여 패킷을 전송하기 시작하는 시점을 1만큼 늦추도록 조정한다.

**[보조정리4]** 임의로 주어진  $N, k, \delta$ 의 값에 대하여, (그림 11)에서 제시한 알고리즘에 의한 송수신 스케줄에서 각각의 노드에 대한 송수신 스케줄의 길이는  $k\delta + N$ 이거나  $k\delta + N - 1$ 이다.

**(증명)** 알고리즘의 가정에 의해  $k$ 번의 지연시간외에 추가로 1만큼 늦추었으므로, 노드 0에 대한 스케줄의 길이는  $k\delta + C + 1 + (j-1)(C+1) + (k-j)C = k\delta + kC + j = k\delta + N$ 이다. 이러한 가정의 변화로 인하여 (그림 9)에서 제시한 행렬  $\Theta$ 의 0번째 행과  $s \bmod k = 0$ 인  $s$ 번째 행이 일치할 뿐만 아니라,  $s \bmod k = 0$  또는  $s \bmod k = 1$ 인 노드  $s$ 에 대한 스케줄의 길이는 모두  $k\delta + N$ 이다. 그러나, 그 밖의 다른 노드에 대한 스케줄의 길이는 앞장의 [보조정리 2]에 의하여 항상  $k\delta + N - 1$ 이다. ■

**[정리3]** 임의로 주어진  $N, k, \delta$ 의 값에 대하여,  $\lceil \frac{N}{k} \rceil (N-1) > k\delta + N - 1$ 이면 (그림 11)의 알고리즘에 따른 송수신 스케줄의 주기는 최적이다.

**(증명)** 만일  $\lceil \frac{N}{k} \rceil (N-1) \leq k\delta + N - 1$ 이면 (정리 1)로부터  $L = k\delta + N - 1$ 이고, [보조정리 4]에 의하면  $s \bmod k = 0$  또는  $s \bmod k = 1$ 인 노드  $s$ 에 대한 스케줄의 길이는 모두  $k\delta + N$ 이므로 최적 스케줄의 주기  $L$ 보다 커지게 된다. 반대로  $\lceil \frac{N}{k} \rceil (N-1) > k\delta + N - 1$ 이면 (정리 1)로부터  $L = \lceil \frac{N}{k} \rceil (N-1)$ 이고  $\lceil \frac{N}{k} \rceil (N-1) \geq k\delta + N$ 이므로, 전송시점이  $L$ 보다 큰 부분을  $L$ 만큼 왼쪽으로 평행이동 시켰을때 전송시점이  $L$ 보다 작은 부분과 겹치는 부분이 전혀 생기지 않는다. 또한, (그림 10)의 예제에서 보듯이 비록 겹치는 부분이 생기지 않더라도 같은 시간에 동일한 파장으로 2개 이상의 패킷을 전송하는 경우가 생기지 않으므로 (그림 11)의 알고리즘에 따른 송수신 스케줄의 주기는 최적이다. ■

본 논문에서 제안한 알고리즘과 [4]의 알고리즘의 특성을 서로 비교하기 위해  $k | N$ 이고  $\lceil \frac{N}{k} \rceil (N-1) > k\delta + N - 1$ 인 경우에 각각의 알고리즘을 사용하여 얻어진 송수신 스케줄을 비교한 결과 최적 방송 주기는 서로 같다는 것을 확인하였다. 그러나, 패킷 전송의 패턴을 자세히 살펴보면 여기에서 제안한 알고리즘에서는 동시에 전송되는 패킷의 개수가 시간의 호

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	4				1	5			2	6			3	7			
1		0	4				5				2	6			3	7	
2	7			0	4			1	5				6				3
3		7				0	4			1	5			2	6		
4			3	7				0				1	5			2	6
5	2	6			3	7			0	4				1			
6			2				3	7			0	4			1	5	
7	5			2	6				3				0	4			1

(그림 12)  $N=8, k=4$ 이고  $\delta=2$ 일 때 구한 송수신 스케줄  
(Fig. 12) Our transmission schedule for  $N=8, k=4$  and  $\delta=2$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4			1	5			2	6			3	7		
1	5			2	6			3	7			0	4		
2	6			3	7			0	4			1	5		
3	7			0	4			1	5			2	6		
4		1	5			2	6			3	7			0	
5		2	6			3	7			0	4			1	
6		3	7			0	4			1	5			2	
7		0	4			1	5			2	6			3	

(그림 13)  $N=8, k=4$ 이고  $\delta=2$ 일 때 [4]에서 구한 최적 송수신 스케줄  
 (Fig. 13) An optimal transmission schedule in [4] for  $N=8, k=4$  and  $\delta=2$

류에 따라 변하는 것을 볼 수 있으나, (4)의 알고리즘에서는 모든 노드가 전혀 패킷을 전송하지 않거나 아니면 항상  $k$ 개씩 일정하게 전송한다는 차이점을 발견하였다. 이러한 차이점은 파장 변경을 위해 각 노드에서 발생하는 지연시간을 서로 동기화(synchronization)하는 데에서 비롯된 것으로서, (4)의 알고리즘을  $k \nmid N$ 인 경우에도 사용할 수 있도록 일반화할 수 없는 근본적인 원인이다.

한편, 아래의 (그림 12)과 (그림 13)는  $k \mid N$ 이고

$\lceil \frac{N}{k} \rceil (N-1) \leq k\delta + N - 1$ 일 때 (4)의 알고리즘이 최적의 송수신 스케줄을 구할 수 있는 반면에 여기에서 제안된 알고리즘을 사용하였을 때 최적이지 않은 예제를 보여주고 있다. 그러나, 위의 조건을  $\delta$ 에 관하여 정리하면  $\delta \geq \frac{N(N-1)}{k} - \frac{(N-1)}{k} \approx \left(\frac{N}{k}\right)^2$

이므로, 파장의 개수  $k$ 가 전체 노드의 개수  $N$ 에 비하여 상대적으로 작다는 가정하에서 현실적으로  $\delta$ 의 값이 크지 않으므로 이러한 경우는 많이 발생하지 않을 것으로 본다.

### 5. 결 론

지금까지 앞장에서 제시한 스케줄링 알고리즘들을 정리하면 [정리 2]와 [정리 3]으로 요약될 수 있다. 즉, passive star형 광상호연결망에서의 방송을 위하여 본 논문에서 제안한 스케줄링 알고리즘을 사용하면 임의로 주어진  $N, k, \delta$ 의 값에 대하여  $k \mid N - 1$ 이면  $\delta$ 의 값에 관계없이 항상 최적의 송수신 스케줄을

구할 수 있으며,  $k \nmid N - 1$ 일 때에도 조건  $\lceil \frac{N}{k} \rceil$

$(N-1) > k\delta + N - 1$ 을 만족하는 범위내에서 최적의 송수신 스케줄을 구할 수 있다. 한편, 파장의 개수  $k$ 가 전체 노드의 개수  $N$ 에 비하여 상대적으로 작다고 가정할 때, 위에서 제시한 조건을  $\delta$ 에 관하여 정리하면 대략  $N$ 과  $k$ 의 비의 제곱보다 작다는 것을 의미하므로 현실적으로 대부분의 passive star형 광상호연결망에 제안된 스케줄링 알고리즘을 적용할 수 있다.

향후 과제로는 (4)에서 제안한 알고리즘과 본 논문에서 제안한 알고리즘의 장단점을 서로 보완하여 임의로 주어진  $N, k, \delta$ 의 값에 대하여 아무런 조건없이 최적의 송수신 스케줄을 구할 수 있는 알고리즘을 연구하고 있다. 또한, passive star형이 아닌 다른 구조의 광상호연결망에서의 방송을 위한 송수신 스케줄링 문제나 방송과는 달리 비규칙적인 일반적인 경우에 적합한 송수신 스케줄링 문제를 해결하고자 한다.

### 참 고 문 헌

[1] C. A. Brackett, "Dense Wavelength Division Multiplexing Networks: Principles and Applications," IEEE J. on Selected Areas in Communications, Vol. 8, No. 6, pp. 948-964, Aug. 1990.  
 [2] H. Choi, H-A. Choi, and M. Azizoglu, "Efficient Scheduling of Transmissions in Optical Broadcast Networks," IEEE/ACM Trans. on Networking, Vol. 4, No. 6, pp. 913-920,

Dec. 1996.

- [3] H. Choi, H-A. Choi, and M. Azizoglu, "On the All-to-All Broadcast Problem in Optical Network," Proceedings of Infocom '97, 1997.
- [4] S-K. Lee, A. D. Oh, H. Choi, H-A. Choi, "Optimal transmission schedules in TWDM optical passive star networks," Discrete Applied Mathematics, Vol. 75, No. 1, pp. 81-91, Jan. 1997.
- [5] G. R. Pieris and G. H. Sasaki, "Scheduling Transmissions in WDM Broadcast-and-select Networks," IEEE/ACM Trans. on Networking, Vol. 2, No. 2, pp. 105-110, April, 1994.
- [6] C. Tocci and H. J. Caulfield, Optical Inter-connection: Foundations and Applications, Artech House, Inc., 1994.
- [7] R. J. Vetter and D. H. C. Du, "Distributed Computing with High-Speed Optical Networks," IEEE Computer, pp. 8-18, Feb. 1993.
- [8] 장석문, 예홍진, 위규범, "Passive Star형 광상호 연결망에서의 방송을 위한 동간격 송수신 스케줄링," 아주대학교 정보통신연구소 논문집, 제1권, 제1호, pp. 165-173, 1998년, 2월.



**장 석 문**

1997년 아주대학교 자연과학부 수학교육(학사)  
 1997년~현재 아주대학교 정보과 학과 석사과정  
 관심분야 : 알고리즘, 계산이론



**변 광 준**

1985년 서울대학교 전자계산기 공학과(학사)  
 1987년 Pennsylvania State Univ., Computer Science(석사)  
 1993년 Univ. of Southern California, Computer Science(박사)

1994년~현재 아주대학교 정보 및 컴퓨터공학부 조교수  
 관심분야 : 데이터베이스, 분산계제 컴퓨팅



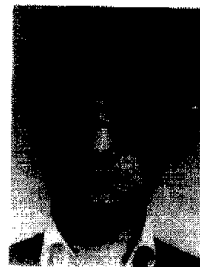
**예 홍 진**

1986년 서울대학교 수학교육과(학사)  
 1988년 아주대학교 전자계산학과(석사)  
 1990년 Univ. Joseph Fourier - INP Grenoble(佛), 응용수학과(D.E.A.)  
 1993년 Univ. Claude Bernard - ENS Lyon(佛), 전자계산학과(박사)  
 1993년~현재 아주대학교 정보 및 컴퓨터공학부 조교수  
 관심분야 : 컴퓨터 산술, 병렬처리



**위 규 범**

1978년 서울대학교 수학과(학사)  
 1984년 University of Wisconsin - sin 전산학과(석사)  
 1992년 Indiana University 전산학과(박사)  
 1993년~현재 아주대학교 정보 및 컴퓨터공학부 조교수  
 관심분야 : 컴퓨터 이론



**홍 만 표**

1981년 서울대학교 계산통계학과(학사)  
 1983년 서울대학교 계산통계학과(석사)  
 1991년 서울대학교 계산통계학과(박사)  
 1983년~1985년 울산공과대학 전자계산학과 전임강사  
 1985년~현재 아주대학교 정보 및 컴퓨터공학부 교수  
 1993년~1994년 미네소타대학 전자공학과 교환교수  
 관심분야 : 병렬처리