

WDM 기술을 이용한 고속 LAN에서의 매체 접근 프로토콜 설계

정희원 백선욱*

A media access protocol for high-speed WDM LAN

Seonuck Paek* Regular Member

요약

본 논문에서는 WDM LAN에서 지속적으로 동작하는 소자를 이용하면서도 효율적인 매체 접근 제어 알고리즘을 제안한다. 제안된 기법에서 각 노드는 두 개의 가변 소자를 갖고 있어서 하나를 통해 데이터를 수신하는 동안 나머지 하나는 다른 채널로 조율하는 파이프라인 방식을 사용함으로써, 느린 소자로 인해 WDM의 큰 대역폭을 활용하지 못하는 문제를 해결한다. 또한, 가변 조율 소자의 조율 시간을 하나의 서브프레임으로 설정하고 하나의 서브프레임 내에는 여러 개의 타임 슬롯을 두도록 하여 패킷의 전송 지연 시간이 가변 소자의 조율 시간보다 더 짧은 경우에도 효과적으로 사용할 수 있도록 하였다. 트래픽 행렬을 이러한 서브프레임에 효과적으로 스케줄링하는 알고리즘을 제안하였으며, 이러한 스케줄링에 기반한 매체 접근 제어 프로토콜을 설계하고 지연시간 성능을 시뮬레이션으로 분석하였다. 분석 결과 제안된 알고리즘은 다양한 트래픽 상황하에서도 좋은 성능을 보임을 확인하였다.

ABSTRACT

In this paper, we design a new media access control algorithm for star-coupler WDM LAN. Each node have two slow tunable receivers and these two receivers are pipelined: while one receiver is receiving data from one channel, the other is tuned to another channel. Tuning time is configured as a subframe and each subframe consists of multiple time slots so that multiple packets are transmitted in a subframe. An efficient scheduling algorithm is designed and its delay performances are analyzed by simulation. The proposed algorithm shows efficient performance under various traffic pattern.

I. 서론

최근에 WDM(Wavelength Division Multiplexing) 기술을 LAN에 적용시키는 분야에 관한 연구가 활발히 진행되고 있다^{[1][2][3][4][5]}. 이러한 WDM LAN에서 각 노드는 여러 개의 병렬 채널을 이용할 수 있게 됨으로써 기존의 단일 채널 통신망에 비해 고속 전송이 가능하다. WDM LAN에서 각 채널은 보통 여러 개의 타임 슬롯으로 구성된다. 단일 채널 LAN과 마찬가지로 WDM 같은 다중 채널 LAN에서도 타임 슬롯들을 각 노드의 트래픽 요구량에 맞게 할당하는 매

체 접근 제어 프로토콜이 필요하다. WDM과 같은 다중 채널 구조에서 이용 가능한 채널은 여럿이지만 각 노드는 보통 한정된 수의 송신장치와 수신 장치를 갖는다. 발신 노드로부터 목적지 노드로 성공적인 패킷 전송이 이루어지기 위해서는 3 가지 형태의 충돌(collision)을 방지해야 한다. 먼저, 같은 채널에 두 노드가 동시에 데이터를 전송하게 되는 채널 충돌(channel conflict)을 방지해야 한다. 또한, 송신 장치의 수 보다 더 많은 채널들이 동시에 송신 노드에 할당되는 송신지 충돌(source conflict)을 해결해야 한다. 마지막으로 목적지 노드에 수신 장치의 수보다

* 상명대학교 전자계산학과(paeksu@smuc.sangmyung.ac.kr)

논문번호 : 98306-0720, 접수일자 : 1998년 7월 20일

* 이 논문은 1996년도 한국학술진흥재단의 공모과제 연구비에 의해 연구되었음

많은 채널을 통해 동시에 패킷이 도착하게 되는 목적지 충돌(destination conflict)을 해결해야 한다^[1].

WDM LAN에서 노드에 송신 장치나 수신 장치가 각각 하나밖에 없을 경우에는 1)하나의 채널로 데이터 송/수신 2) 새로운 채널로 조율 3) 새로운 채널로 데이터 송/수신 하는 과정을 반복하며 채널을 사용한다. 현재까지 WDM LAN에서의 매체 접근 제어 프로토콜에 대해 많은 연구가 진행되어왔는데, 대부분이 하나의 채널에서 다른 채널로 송수신 장치를 조율시키는데 걸리는 시간이 매우 짧다고 가정하고 있다^{[1][2][3]}. 그러나, 현재 개발되고 있는 광소자들의 조율 시간은 수 백 ms에서 수 ms까지 다양하며 특히, 조율 속도가 빠른 소자일수록 고가이며 안정성이나 채널 선택 범위가 좁은 실정이다^[4]. 한편, 최근의 연구 [7][8][9]에서는 채널 사이의 조율시간의 영향을 줄일 수 있는 프로토콜들을 제안하고 있다. 그러나, 이들 프로토콜에서도 조율 시간 동안의 낭비로 인한 대역폭 손실은 피할 수 없다.

이러한 조율 시간 동안의 대역폭 낭비를 줄일 수 있는 방법은 각 노드에서 송수신 장치를 여러 개 사용하여 파이프라인(pipeline) 방식으로 동작시키는 것이다^[10]. 파이프라인 기법이 가능하기 위해서는 최소한 두 개의 가변 소자가 필요하다. 예를 들어, 각 노드가 하나의 고정 송신기와 두 개의 가변 수신기를 갖는 구조라면 하나의 가변 수신기가 패킷을 수신하고 있는 동안 나머지 하나는 다른 채널로 조율하는 방식으로 파이프라인 기법을 적용할 수 있다. 그러나, 타임 슬롯(=패킷) 단위로 파이프라인 동작을 수행하기 위해서는 한 타임 슬롯 시간 내에 조율이 가능한 소자가 요구되는데, 타임 슬롯의 길이는 WDM 채널의 대역폭 측면에서 보면 상당히 짧다고 볼 수 있으므로 타임 슬롯(=패킷) 단위의 파이프라인에서도 상대적으로 빠른 소자가 필요하다. 본 연구에서는 빠른 송수신 소자를 가정하였던 지금까지의 연구와는 달리 빠른 소자를 필요로 하지 않으면서 파이프라인 방식의 장점을 살리는 구성에서의 매체 접근 프로토콜을 제안한다. 예를 들어 보통 1-10 ns의 조율시간이 걸리는 acousto-optic 소자를 고려하는데, 그 이유는 가격이 싸며, 동기화가 쉽고 또한, 조율범위가 넓은 장점 때문이다^[6].

느린 송수신 소자를 사용할 경우 하나의 채널에서 다른 채널로 조율하는데 걸리는 시간은 하나의 패킷을 전송하는데 필요한 타임 슬롯의 길이에 비해 일반적으로 길다. 본 논문에서는 이러한 문제를 해결하기 위해서 수신 장치의 조율 시간을 하나의 서브프

레이밍으로 설정하고 각 서브프레임을 여러 개의 타임 슬롯으로 구성함으로써 짧은 패킷 전송 시간과 긴 조율 시간이라는 문제를 해결하고자 한다. [11]에서는 느린 송수신 장치를 사용한 파이프라인 구조에서 회선 교환 트래픽을 스케줄하는 문제를 다룬 바 있다. LAN 환경에서의 데이터 트래픽은 random한 특성을 지니므로 본 연구에서는 서브프레임 단위의 파이프라인 구조에서 패킷 교환용 트래픽을 스케줄하는 문제를 다룬다.

본 논문의 구성은 다음과 같다. 먼저 II절에서는 본 논문에서 제안한 통신망 구성과 노드 구성에 대해 기술하며 III절에서는 제안된 통신망 구조에서 스케줄링 알고리즘에 대해 그리고 IV절과 V절에서는 제안된 스케줄링 알고리즘의 성능 분석 결과를 기술한다.

II. 통신망과 노드의 구성

1. 통신망 구성

WDM LAN에서 현재 가장 활발히 연구되고 있는 토폴로지는 그림 1과 같은 성형 토폴로지이다. 이 토폴로지는 망의 신뢰도나 확장성 측면에서 다른 토폴로지보다는 우수하다고 알려져 있다^[1]. 따라서 본 연구에서도 그림 1과 같은 토폴로지를 가정한다. 여기서 노드의 수는 N , 채널(=파장)의 수는 $C+1$ 이라 하자. $C+1$ 채널 중 하나의 채널은 제어 채널로서 트래픽 요구량 등의 제어 정보 전송을 위해 사용하며, 나머지 C 개의 채널은 데이터 채널로 데이터 전송을 위해 사용한다. 또한, 각 노드는 제어 채널에 고정되어 있는 송신기 하나와 고정 수신기 하나를 가지고 있어서 제어 채널에서의 송수신은 목적지 충돌없이 항상 가능하도록 한다. 또한, 데이터 채널 접근을 위해 고정 송신 장치 하나와 조율가능한 가변 수신기를 2 개씩 가지고 있다. 제어 채널과 데이터 채널은 각각 일정 크기의 타임 슬롯으로 구성되어 있으며, 데이터 채널들의 타임 슬롯은 경계에서 서로 동기화되어 있다고 가정한다(제어 채널의 제어용 타임 슬롯의 크기와 데이터 채널의 타임슬롯의 크기가 반드시 같을 필요는 없다). 제어 채널은 N 개의 제어용 타임 슬롯을 한 단위로 구성되는데, 이 타임 슬롯들은 각 노드에 고정적으로 할당되어 있다. 각 노드는 데이터 전송에 앞서 자신이 보낼 데이터 패킷이 있음을 제어 채널의 제어용 타임 슬롯을 통해 다른 노드들에게 알리고, 각 노드는 그 정보들을 트래픽 행렬 형태로 유지, 관리한다. 각 노드는 제어 채널의

정보로부터 분산 방식으로 같은 스케줄링 알고리즘을 적용시킴으로써 동일한 스케줄 결과에 도달할 수 있다.

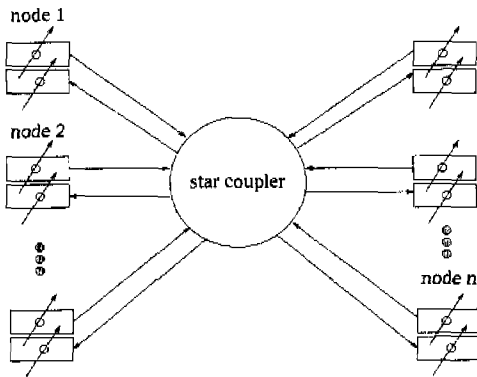


그림 1. WDM 성형 coupler
Fig 1. WDM Star topology

2. 노드의 구성

각 노드는 데이터 채널 사용을 위해 하나의 고정 송신기와 두 개의 가변 수신기를 사용한다. 두 개의 가변 수신기는 파이프라인으로 동작시키는데, 하나의 수신기가 하나의 채널(=파장)으로부터 데이터를 수신하고 있는 동안 나머지 하나의 수신기는 다른 채널로 조율하여 다음 번 수신 준비를 한다. 본 연구에서는 느린 조율속도를 갖는 수신기와 상대적으로 짧은 패킷 길이를 가정하고 있으므로, 수신기가 하나의 채널에서 다른 채널로 조율하는데 드는 시간 길이를 하나의 서브프레임으로 구성하고, 또한, 각 서브프레임은 여러 개의 타임 슬롯으로 구성한다. 이렇게 함으로써 타임 슬롯 단위의 파이프라인이 아니라 서브프레임 단위의 파이프라인으로 동작하게 된다. 또한, 서브프레임 여러 개를 연결하여 하나의 큰 프레임(예를 들어 125 μ s 크기의 프레임)을 구성할 수도 있다. 그림 2는 8개의 노드로 구성된 통신망에서 사용가능한 파장의 수가 4개일 때의 스케줄링의 예를 보여주고 있다. 그림에서 각 노드의 하나의 수신장치는 현재의 서브프레임이 끝나야 다른 파장으로 조율을 시작할 수 있다. 이렇게 함으로써 각 노드의 하나의 수신장치는 동일 서브프레임 내에서는 두 번 다시 조율이 되는 일이 없도록 해야 한다. 그림 2는 하나의 서브프레임이 4 개의 타임 슬롯으로 구성되며, 또한 4개의 서브프레임이 하나의 프레임을 이루는 구조의 예를 보여주고 있다. 여기서 전체 프레임의 길이는 음성 등의 실시간 정보 전송을 위해 125 μ s 길이로 정해질 수 있다.

	subframe-0			subframe-1			subframe-2			subframe-3		
	0	1	2	3	0	1	2	3	0	1	2	3
W0(T0, T4)		D2	D7	D4		D1			D5		D6	
W1(T1, T5)	D3	D5	D4	D6			D6		D3			D4
W2(T2, T6)	D1	D0	D0	D1		D7	D7		D1	D1	D4	D3
W3(T3, T7)					D2	D3	D2	D2	D2	D0	D0	D5
												D1

그림 2. 프레임/서브프레임 구조
Fig 2. Structure of Frame/Subframe

그림 2의 시스템은 W1, W2, W3, W4의 4 개의 데이터 채널을 가지고 있으며, 노드의 수는 8 인 경우이다. 여기서 각 노드의 송신기는 자신에게 할당된 채널에 고정 할당되는데, 노드의 수가 채널의 수보다 더 많기 때문에 하나의 채널에 여러 개의 노드가 고정 할당되게 된다. 채널 W0에는 노드 0번과 4번의 송신기가 고정되어 있으며, W3에는 노드 3번과 노드 7번의 송신기가 고정 할당된다. W1과 W2도 마찬가지로 해석할 수 있다. Dk는 노드 k에서의 수신장치를 나타낸다. 그림 2에서 서브프레임 0번의 0번 타임 슬롯에서 노드 2의 가변 수신기가 W0에 조율되어 있음을 보여주고 있으므로 노드 0이나 노드 4가 전송하는 데이터를 수신할 수 있음을 나타내고 있다. subframe-0에서 W3는 더 이상 스케줄될 수 없어서 완전히 낭비될 수 밖에 없는데, 그 이유는 8개의 모든 노드의 수신기들이 이미 W0, W1, W2에 이미 조율되어 있어 W3로의 조율이 불가능하기 때문이다. 또한, 노드 2의 수신기는 이미 W0에 조율되어 있기 때문에 다른 채널(W1,W2,W3)로는 노드 2에게 데이터를 보낼 수 없음을 나타낸다. 그러나 subframe-1의 0 번과 3번 타임 슬롯에서는 채널 W3을 통해 노드 3이나 노드 7이 전송하는 데이터를 수신할 수 있다는 것을 보여주고 있다. 그림 3은 노드 2의 두 개의 수신기 Ra와 Rb가 파이프라인 방식으로 동작하는 예를 보여주고 있다.

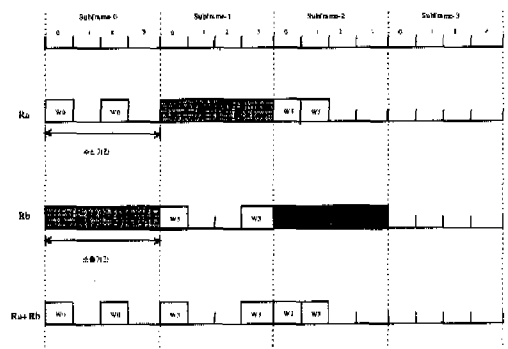


그림 3. 노드 2에서의 수신기의 pipelining
Fig 3. pipelining of receivers in node 2

III. 타임 슬롯 할당 스케줄링 알고리즘

1. 스케줄링 알고리즘

본 절에서는 그림 2와 같은 구조의 프레임에 트래픽을 스케줄하는 알고리즘에 대해 기술한다. 각 노드는 별도의 제어 채널을 통해 각 노드가 보내고자 하는 패킷에 관한 정보(예를 들어 패킷의 목적지 등)를 망에 있는 다른 모든 노드들에게 알리며, 각 노드는 이 정보를 수신하여 트래픽 행렬 $B = \{b_{ij}\}$ 를 유지한다. 여기서 b_{ij} 는 노드 i 가 노드 j 에게 전송하고자 하는 패킷의 수를 나타낸다. 각 노드는 이 행렬로부터 스케줄링 알고리즘을 분산 방식으로 동작시키며 그 스케줄 결과에 맞추어 자신의 가변 수신기를 조율한다. 스케줄을 할 때 고려해야 할 것은 가능하면 많은 양의 트래픽이 프레임 안에 스케줄될 수 있도록 하는 것이다. 이를 위해 본 논문에서는 가능하면 많은 양의 트래픽이 하나의 서브프레임 내에 수용될 수 있도록 하는 스케줄링 알고리즘을 설계한다. 스케줄링 시의 고려 사항은 다음과 같다.

첫째, I절에서 설명한 채널 충돌이 없어야 한다. 즉, 채널을 공유하는 노드들이 같은 타임슬롯에 패킷을 전송하도록 스케줄되는 일이 없어야 한다.

둘째, 목적지 충돌이 없어야 한다. 즉, 하나의 서브프레임 내에서 하나의 노드가 두 개 이상의 채널로부터 데이터를 수신하도록 스케줄되는 일이 발생하지 않아야 한다.

셋째, 각 노드는 타임 슬롯 길이 단위의 패킷을 전송한다고 가정하는데, 하나의 서브프레임에 있는 타임 슬롯의 전체 개수는 노드의 수와 같거나 적게 구성한다. 이것은 각 노드가 하나의 서브프레임에서 최대 한 번 씩만 전송 기회를 가질 수 있도록 하기 위해서이다. 실제로 한 서브프레임에 있을 수 있는 타임 슬롯의 수는 가변 수신기의 조율 시간과 타임 슬롯의 길이에 의해 결정된다.

이상과 같은 가정으로부터 본 논문에서는 [10]의 MRS 알고리즘을 본 논문에서 제안한 서브프레임 구조에 적용시킨 새로운 스케줄링 알고리즘을 제안한다. 먼저 각 노드는 $B = \{b_{ij}\}$ 로부터 트래픽 존재 행렬 $D = \{d_{ij}\}$ 를 계산하는데, b_{ij} 가 1 이상이면 d_{ij} 의 값은 1이 되고 그렇지 않으면 0이 된다. 또한, $D = \{d_{ij}\}$ 로부터 $C \times N$ 의 채널 부하 행렬 $L = \{l_{ij}\}$ 를 유지하는데, l_{ij} 는 채널 i 를 통해 목적지 j 로 전송하고자 하는 패킷 수의 합을 나타낸다(여기서 C 는 데이터채널의 수). 그림 2에서는 두 개의 노드가 하나

의 채널을 공유하는데, 노드 0과 노드 4가 채널 W0를 공유하므로 $l_{0,3} = d_{0,3} + d_{4,3}$ 과 같이 계산된다.

본 논문에서 제안하는 스케줄링 알고리즘은 그림 4에 나타나 있다. 그림 4는 하나의 서브프레임 내에서의 스케줄링 알고리즘만을 기술하고 있는데, 이 알고리즘을 여러 번 반복하면 프레임 전체에 대한 스케줄 결과를 얻을 수 있다. 또한 K 는 하나의 채널을 공유하는 노드의 수를 나타내는데, $K = \lfloor N/W \rfloor$ 로 표현된다. 또한, 아래의 알고리즘에서 r_j^i 는 행렬 $L^* = \{l_{ij}^*\}$ 의 i 행에서 크기가 a 인 원소들의 개수를 나타내며, c_j^i 는 j 열에서 크기가 a 인 원소의 개수를 나타낸다. 또한, T_{ij} 를 한 서브프레임 시간 동안의 타임 슬롯 개수를 나타낸다고 하자. 그림 2의 서브프레임 구조에서 한 채널로 최대 4개까지 전송이 가능하므로 $T_{ij} = 4$ 가 된다. 또한 그림 3에서 단계 3과 13의 f_j 는 현재의 서브프레임에서 채널 i 의 T_{ij} 개의 타임 슬롯 중 아직 할당되지 않은 타임 슬롯의 수를 의미한다.

```

=====
1. compute D from B;
2. compute L from D;
3. a = K; set f_j = T_ij for i = 1, 2, ..., C;
4. Compute C x N matrix L* = {l_ij*} such that
   l_ij* = min(l_ij, f_j);
5. Find [r_1^a, r_2^a, ..., r_C^a] and [c_1^a, c_2^a, ..., c_N^a] from
   matrix L*;
6. If [r_1^a, r_2^a, ..., r_C^a] does not contain any non-zero element {
   a = a - 1;
   goto 5;
}
7. Find a non-zero smallest element in [r_1^a, r_2^a, ..., r_C^a] and a
   non-zero smallest element in [c_1^a, c_2^a, ..., c_N^a];
8. If the smaller of the two elements is a row sum,
   denote its position as p and goto 9;
   else
   denote the position of the column sum as q and goto 10;
9. Find q such that c_q^a = min_k {c_k^a | l_pk = a}, then goto 11;
10. Find p such that r_p^a = min_k {r_k^a | l_kp = a}, then goto 11;
11. schedule l_pq* into the subframe;
12. l_pq* = l_pq* - l_pq* /* adjust L */
13. f_p = f_p - l_pq* /* adjust # of available time slots in channel
   p */ j = 1, 2, ..., C
14. l_pj* = 0 for j = 1, 2, ..., N /* adjust L* */
   l_pj* = min(l_pj*, f_p) for j = 1, 2, ..., N;
15. goto 4;
=====

```

그림 4. 스케줄링 알고리즘
Fig 4. Scheduling Algorithm

2. 스케줄링 예

본 절에서는 1절에서 기술한 스케줄링 알고리즘을 예를 들어 기술한다. 그림 2의 프레임 및 채널 할당과 같이 노드의 수가 8이고 채널의 수는 4이며 T0와 T4는 W0에 고정되어있고 T1과 T5는 W1에, T2와 T6는 W2에 그리고 T3와 T7은 W3에 송신기가 고정되어 있다고 가정한다. 또한, 서브프레임의 길이 $T_{sf}=4$ 인 경우를 가정한다. 이때 그림 5와 같은 채널 부하 행렬 L 을 예로 들어 스케줄해보자. 그림 5에서 (W0, 3)에 해당하는 항목은 값이 2인데, 이것은 W0를 공유하는 노드들(노드 0과 노드 4)이 노드 3으로 전송할 패킷이 2 개 있다는 의미이다. 즉, 노드 0도 전송할 패킷이 있고 노드 4도 전송할 패킷이 있다는 의미이다. 그림 5에서 c_j^i 는 i 행에서 크기가 2인 원소들의 개수를 뜻하며, r_i^j 은 크기가 1인 원소들의 개수를 나타낸다. c_j^i , r_i^j 은 j 열에서 크기가 2인 원소들의 개수와 크기가 1인 원소들의 개수를 각각 나타낸다.

	0	1	2	3	4	5	6	7	f_i	r_i^j	r_i
W0	1	1	1	2	0	1	1	1	4	1	
W1	2	0	0	1	1	2	1	0	4	2	
W2	2	1	1	0	0	2	1	4	2		
W3	2	2	2	2	2	1	1	4	6		
c_j^i	3	1	1	2	1	2	1	0			
r_i^j											

그림 5. 채널 부하 행렬 L
Fig 5. Load matrix of channels

먼저 c_j^i 와 r_i^j 의 원소들 중 가장 작은 원소를 찾으면 c_j^i 중에서 $j=1,2,4,6$ 일 때 최소값 1을 가지며, r_i^j 중에서는 $i=W0$ 일 때 최소값 1을 갖는다. 최소값 1을 갖는 $c_1^0, c_2^0, c_4^0, c_6^0$ 중에서 임의로 하나를 선택해야 하는데, 이 선택은 [13]에서 제시된 방법처럼 각 노드가 같은 seed로부터 시작하는 난수 발생기를 유지한다면 각 노드는 동시에 같은 난수를 발생시킬 수 있고 이 난수로부터 $c_1^0, c_2^0, c_4^0, c_6^0$ 중의 하나를 같이 선택할 수 있다. 또는 각 노드가 선택을 시작하는 열을 라운드 로빈 방식으로 선택할 수 있다. 만일 c_1^0 를 선택했다고 하면 1 열의 원소 중에서 항목의 크기가 2인 항 (W3, 1)을 선택하면, W3를 통해 목적지 노드 1로 2 개의 패킷을 전송하도록 스케줄하는 것이 된다. 즉, 노드 3과 노드 7이 W3을 공유하므로 노드 3이 하나, 노드 7이 하나의 패킷을 전송하도록 스케줄하는 것이 된다. 노드 1의 수신 장치

는 이제 채널 W3에 스케줄되었으므로 이번 서브프레임에서는 다른 채널로부터는 데이터를 더 이상 수신할 수 없다. 따라서, 1번 열을 지운다. 또한, W3의 사용 가능한 타임 슬롯이 2로 줄었으므로 $f_3=2(=4-2)$ 로 갱신한다. 갱신된 후의 그림이 그림 6에 나타나 있다.

	0	1	2	3	4	5	6	7	f_i	r_i^j	r_i
W0	1		1	2	0	1	1	1	4	1	
W1	2		0	1	1	2	1	0	4	2	
W2	2		1	1	0	0	2	1	4	2	
W3	2		2	2	2	1	1	2	6		
c_j^i	3		1	2	1	2	1	0			
r_i^j											

그림 6. 채널 부하 행렬(계속 1)
Fig 6. Load matrix of channels(cont. 1)

그림 6에서도 그림 5에서 진행한 과정과 마찬가지로 하면 되는데, 예를 들어 (W3, 2)를 선택했다고 하자. 이 내용을 반영하면 그림 7과 같이 된다. 그림 7에서 채널 W3은 더 이상 할당할 슬롯이 남아있지 않으므로 W3행의 내용을 모두 지운다. 한 서브프레임의 타임 슬롯 길이가 4이므로 4개 모두 다 할당되었으므로 이번 서브프레임에서 W3에는 더 이상 타임 슬롯을 할당할 수 없기 때문이다.

	0	1	2	3	4	5	6	7	f_i	r_i^j	r_i
W0	1			2	0	1	1	1	4	1	
W1	2			1	1	2	1	0	4	2	
W2	2			1	0	0	2	1	4	2	
W3											
c_j^i	2			1	0	1	1	0			
r_i^j											

그림 7. 채널 부하 행렬 (계속 2)
Fig 7. Load matrix of channels(cont. 2)

이러한 과정을 반복하면 각각 그림 7에서 (W0, 3), (W1, 5)가 순서대로 스케줄되어 그림 8과 같이 갱신된다.

	0	1	2	3	4	5	6	7	f_i	r_i^j	r_i
W0	1				0		1	1	2	0	
W1	2				1		1	0	2	1	
W2	2				0		2	1	4	2	
W3											
c_j^i	2				0		1	0			
r_i^j											

그림 8. 채널 부하 행렬 (계속 3)
Fig 8. Load matrix of channels(cont. 3)

이상의 과정을 계속하여 그림 8에서 (W2, 6)을 선택하면 그림 9와 같이 된다.

	0	1	2	3	4	5	6	7	f_i	r_i^1	r_i^2
W0	1				0			1	2	0	
W1	2				1			0	2	1	
W2	2				0			1	2	1	
W3											
c_i^1	2				0			0			
c_i^2											

그림 9. 채널 부하 행렬 (계속 4)
Fig 9. Load matrix of channels(cont. 4)

그림 9에서는 r_i^2 의 항들이 c_i^2 의 항들보다 더 적은 값을 가지므로 (W1, 0)를 스케줄하면 되는데, 스케줄 후의 그림은 그림 10에 나타나 있다.

	0	1	2	3	4	5	6	7	f_i	r_i^1	r_i^2
W0								1	2	0	1
W1											
W2								1	2	0	1
W3											
c_i^1											
c_i^2								2			

그림 10. 채널 부하 행렬(계속 5)
Fig 10. Load matrix of channels(cont. 5)

그림 10에서는 r_i^1 나 c_i^2 의 항 중 0이 아닌 것이 더 이상은 없으므로 r_i^1 과 c_i^2 으로부터 지금까지의 과정을 반복하는데, (W0, 1)을 스케줄하면 모든 스케줄이 완료된다. 그림 4의 트래픽 부하 행렬을 스케줄한 최종 결과를 타임 슬롯의 길이가 4인 서브프레임에 표시하면 그림 11과 같다.

	0	1	2	3
W0(T0, T4)	D3	D3	D7	
W1(T1, T5)	D5	D5	D0	D0
W2(T2, T6)	D6	D6		
W3(T3, T7)	D1	D1	D2	D2

그림 11. 서브프레임 스케줄링
Fig 11. Scheduling of Subframe

그림 11의 서브프레임은 더 이상 스케줄될 수 없는데, W0의 3번 슬롯과 W2의 2, 3번 슬롯은 비어있긴 하지만 이미 모든 노드의 수신기들이 다른 채널로 조율하도록 스케줄되어 있으므로 더 이상의 스케줄이 일어날 수 없다. 즉, 그림 4에서 0번 노드로 W2나 W0를 통해 전송하고자 하는 트래픽은 있으나, 0번의 수신기는 이미 W3에 스케줄되어 있으므로 W2나 W0를 통한 0번 노드로의 스케줄은 불가능하다(목적지 충돌 문제). 현재까지의 스케줄 결과를 그림 5에 반영하면 그림 12와 같다.

	0	1	2	3	4	5	6	7	f_i	r_i^1	r_i^2
W0	1	1	1	0*	0	1	1	0*			
W1	0*	0	0	1	1	0*	1	0			
W2	2	1	1	1	0	0*	0*	1			
W3	2	0*	0*	2	2	2	1	1			
c_i^1											
c_i^2											

그림 12. 새로운 채널 부하 행렬
Fig 12. load matrix of channels after scheduling

그림 12에서 * 표시는 그림 5에서 10까지의 스케줄링 결과로 처리된 부분을 의미한다. 또한, 노드 4는 아무런 패킷도 수신할 수 없음을 보여주는데, 그 이유는 4번 노드가 조율할 필요가 있는 W1 채널이나 W3 채널의 타임 슬롯들이 이미 0번 5번 노드와 1번 2번 노드에 이미 할당되어 있기 때문에 여분의 타임 슬롯이 없기 때문이다.

그림 12에 대해 위의 스케줄링 과정을 반복하면 두 번째 서브프레임에 대한 스케줄 결과를 얻을 수 있다.

IV. 시간 복잡도 분석

본 절에서는 III절에서 제안된 알고리즘(그림 4)의 시간 복잡도(time complexity)를 분석하고자 한다. 단계 1과 2에서는 각각 $O(C)$ 과 $O(CM)$ 의 시간이 필요하다. 단계 3에서 $[r_i^1, r_i^2, \dots, r_i^N]$ 와 $[c_i^1, c_i^2, \dots, c_i^N]$ 를 계산하기 위해서는 $O(NC)$ 의 시간이 필요하며, 단계 5에서는 $O(C+M)$ 의 시간이 필요하다. 단계 7과 8 및 12에서는 각각 $O(N)$, $O(C)$, $O(C+M)$ 의 시간이 필요하다. 위의 각 단계 중에서 가장 시간이 많이 걸리는 것은 단계 2와 단계 3의 $O(NC)$ 이다. 또한, 최악의 경우에 단계 3에서 13가지가 N 개의 노드 모두에 대해 수행되는 경우를 생각하면 $O(N)$ 번 수행된다(N 개 노드의 수신 장치가 조율해야 할 채널이 결정되면 해당 서브프레임에서는 더 이상 스케줄될 수 없음). 따라서, 그림 3에 제시되어 있는 알고리즘의 시간 복잡도는 $O(N^2C)$ 가 된다.

만일 $C \ll N$ 인 경우라면 제안된 알고리즘의 시간 복잡도는 $O(N^2)$ 에 가까워진다. 어떤 스케줄링 알고리즘이던지 $N \times N$ 의 트래픽 행렬 $B = \{b_{ij}\}$ 를 스케줄하기 위해서는 $N \times N$ 개의 원소를 한번씩은 처리해야 하므로, $O(N^2)$ 은 $N \times N$ 트래픽 행렬 스케줄링 알고리즘의 시간 복잡도의 하위 임계치(lower bound)가 된다. 즉, $C \ll N$ 인 경우라면 본 논문에서 제시된

알고리즘의 시간 복잡도는 최적 스케줄링 알고리즘의 시간 복잡도에 근사하게 된다.

V. 성능 분석

본 절에서는 본 논문에서 제안된 프로토콜의 성능을 시뮬레이션을 이용하여 분석해 보고자 한다. 시뮬레이션 라이브러리는 event-driven 방식인 SMPL을 사용하였다^[12]. 아래의 시뮬레이션에서 구한 값들은 BMS(batch mean analysis)^[12] 방식으로 구하였는데, 시뮬레이터 시스템의 초기의 불안정한 값들의 영향을 제거하였으며, 95%의 신뢰도를 만족하도록 수행시켰다.

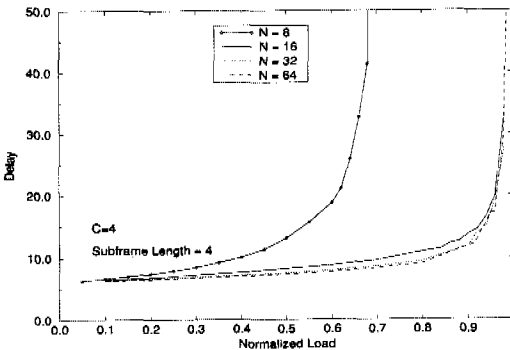


그림 13. 노드 수의 변화에 따른 지연 시간 성능
Fig 13. Delay vs. load under various node numbers

먼저, 그림 13은 채널의 개수가 4이고 서브프레임의 길이가 4인 경우에 부하에 따른 지연 시간을 보여주고 있다. 여기서 모든 노드에는 평균 도착율이 1인 포아송 분포에 따라 패킷이 도착한다고 가정하였으며, 도착한 패킷의 목적지는 자신을 제외한 $N-1$ 개의 노드 중에서 같은 확률로 결정된다고 가정하였다(대칭 구성). 여기서 지연 시간은 노드에 도착한 패킷이 스케줄되어 전송되기 시작하는 시간에서 도착 시작을 뺀 값을 나타낸다. 또한, 부하는 채널 용량에 대해 정규화시켜 표현하였다.

그림 13에서 알 수 있듯이 같은 조건일 때 노드 수가 많아질수록 성능이 좋아지는 것을 볼 수 있다. 그것은 채널 부하 행렬 L 에서 열의 길이가 길어지고, 값이 1인 항들이 넓게 분포하게 되므로 상대적으로 목적지 충돌이나 송신지 충돌의 영향을 줄일 수 있음을 보여준다. 노드의 수 $N = 8$ 인 경우는 0.7 정도의 최대 처리율이 나오지만 $N = 16$ 이상인 경우에는 0.95 이상의 처리율이 나오는 것을 알 수 있다. 또한, N 이 16보다 큰 경우의 성능은 큰 차이가 없

음을 알 수 있다. 저 부하에서의 지연시간은 a)패킷이 노드에 도착해서 스케줄링이 시작될 때까지 기다리는 시간에 b)스케줄링의 결과에 따라 목적지 노드가 자신의 가변 수신기를 조절하는데 걸리는 시간을 더한 값이 된다. a)의 값은 서브프레임 길이의 1/2 정도가 되고, b)의 값은 하나의 서브프레임의 길이가 된다. 따라서 그림 12에서 저부하에서의 지연 시간은 $2 + 4 = 6$ 슬롯 정도가 됨을 알 수 있다.

그림 14는 $N = 32, C = 4$ 인 구성에서 서브프레임의 길이(=타임 슬롯의 수)가 변할 때의 지연시간 성능에 대해 보여주고 있는데, 그림 14에서도 그림 13과 같이 대칭 구성을 가정하였다. 즉, 각 노드에는 같은 비율로 패킷이 도착하며 그들의 목적지 분포는 자신을 제외한 $N-1$ 노드 중에서 임의로 선택한다고 가정하였다.

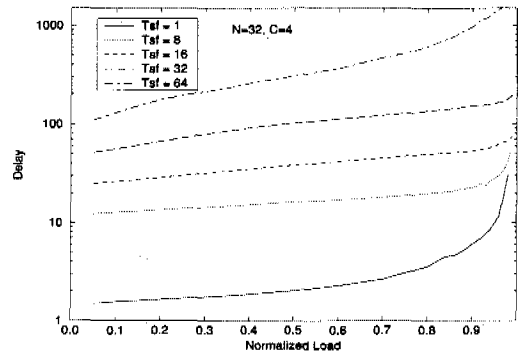


그림 14. 서브프레임의 길이의 변화에 따른 성능
Fig 14. Delay vs. load under various subframe length

그림 14에서 알 수 있듯이 서브프레임의 길이가 커지면 지연시간도 같이 커진다는 것을 알 수 있는데, 그 이유는 서브프레임의 길이가 커지면 불안정한 스케줄링으로 인해 낭비되는 부분이 많아지기 때문이다(그림 11 참조). 또한, 서브프레임의 길이가 길면 스케줄링될 때까지 기다리는 시간도 길어지고 스케줄링 결과에 따라 목적지가 수신기를 조절하는데 걸리는 시간도 상대적으로 길어지기 때문이다.

그림 15는 비대칭적 트래픽 분포에 대한 지연 시간 성능을 보여주고 있는데, 망내에 하나의 hot-spot이 존재하여 다른 노드보다 더 많은 트래픽을 생성하는 경우와 같은 비대칭 분포를 가정하고 있다. 이러한 예는 망 내에 하나의 서버가 있어서 다른 노드(클라이언트)보다 더 많은 트래픽을 발생하는 경우를 들 수 있다. 그림 14에서 보통의 노드(클라이언트)들은 채널 용량에 비해 0.5의 트래픽을 발생시킨다고 가정하였다. 그림에서 알 수 있듯이 서버가 클라이언

트 하나보다 8배(4.0)의 트래픽을 발생시키더라도 클라이언트의 성능에서는 큰 영향이 없음을 알 수 있다. 그러나 서버의 지연시간은 이 시점부터 급격히 증가하기 시작한다.

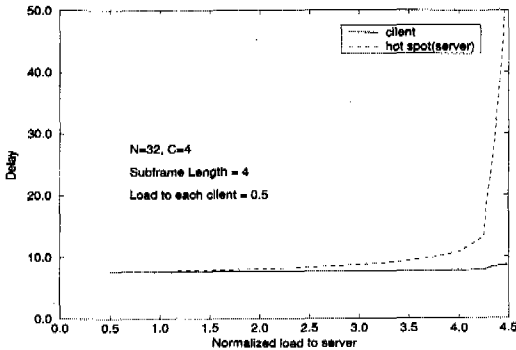


그림 15. hot-spot이 있는 경우의 성능
Fig 15. Delay vs. load under hot-spot condition

그림 16도 비대칭적 트래픽 분포에 대한 지연 시간 성능을 보여주고 있는데, 32개의 노드를 4개의 다른 집합으로 나누고 각 집합에 속한 노드에 도착하는 패킷의 도착율이 각각 $\lambda-0.2$, $\lambda-0.1$, $\lambda+0.1$, $\lambda+0.2$ 인 경우의 지연시간 성능을 보여주고 있는데, λ 가 0에서 0.8까지 변화할 때까지는 대칭적 분포일 때의 성능과 큰 차이가 없다는 것을 보여주고 있다.

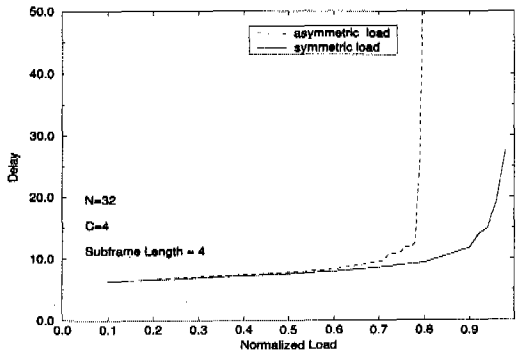


그림 16. 비대칭 트래픽일 때의 성능
Fig 16. Delay vs. load under asymmetric load condition

그림 17은 각 노드가 패킷을 발생시키는 확률은 같으나, 그들의 목적지가 편중되는 경우의 성능을 보여주고 있다. 편중된 트래픽 모델의 예로서 [10]에서는 노드를 두 개의 그룹으로 나누어 각 노드에서 발생시킨 트래픽의 양은 같지만, 자기와 같은 그룹에 속한 노드에게 전송할 확률이 다른 그룹에 속한 노드에게 전송할 확률보다 b 배 높은 경우를 들고 있는데, 여기서도 [10]의 트래픽 모델을 적용한다. 이

모델은 트래픽이 일반적으로 국지성(traffic locality) 성격을 갖는 것을 고려하면 타당한 모델이다. [10]에서는 일반적으로 N 개의 노드를 $N/2$ 개씩 A 그룹과 B 그룹으로 나누어 각 그룹간의 통신시의 목적지 분포가 다음과 같은 경우를 고려한다. (아래에서 Q_{ij}^{XY} 는 그룹 X 에 속하는 노드 i 에서 발생한 패킷의 목적지가 그룹 Y 에 속하는 노드 j 일 확률을 의미한다.)

$$Q_{ij}^{AA} = \frac{2b}{(b+1)N-2b}$$

$$Q_{ij}^{AB} = \frac{2}{(b+1)N-2b}$$

$$Q_{ij}^{BA} = \frac{2b}{(b+1)N-2}$$

$$Q_{ij}^{BB} = \frac{2}{(b+1)N-2}$$

$N=8$, $b=2$ 인 경우에 $\{0,1,\dots,N/2-1\}$ 와 $\{N/2,N/2+1,\dots,N-1\}$ 의 두 개의 그룹으로 나누어 위의 모델을 적용시키면 목적지 분포가 다음과 같이 된다.

-	2	2	2	1	1	1	1
2	10	10	10	10	10	10	10
10	-	2	2	1	1	1	1
2	10	10	10	10	10	10	10
10	2	2	2	1	1	1	1
2	10	10	10	10	10	10	10
10	2	2	2	-	1	1	1
1	11	11	11	11	-	2	2
11	1	1	1	1	2	-	2
1	11	11	11	11	11	11	11
11	1	1	1	1	2	2	-
1	11	11	11	11	11	11	11
11	1	1	1	1	2	2	-
1	11	11	11	11	11	11	11
11	1	1	1	1	2	2	-

$N=8$ 인 경우에 $\{0,2,\dots,N-2\}$ 와 $\{1,3,\dots,N-1\}$ 의 두 개의 그룹으로 나누어 [10]의 트래픽 발생 모델을 본 논문의 스케줄링 알고리즘에 적용시켰을 때의 성능이 그림 17에 나타나 있다.

그림 17에서 알 수 있듯이 성능은 b 의 변화에 큰 영향을 받지 않음을 알 수 있는데, 부하가 높을 때 대칭 분포($b=1$)에 비해 비대칭 분포일 때가 약간 성능이 떨어짐을 알 수 있다. 이것은 비대칭 분포일 지라도 채널 부하 행렬 L 의 행의 길이가 길어서 1인 항이 넓게 분포하기 때문에 스케줄링이 잘 되고 있음을 알 수 있다. 그림 18은 $N=32$ 일 때 목적지 분포가 비대칭일 때의 성능을 보여주고 있는데, 이와

같이 채널 부하 행렬 L 의 행의 길이가 긴 경우에는 비대칭 분포의 영향을 거의 받지 않음을 알 수 있다.

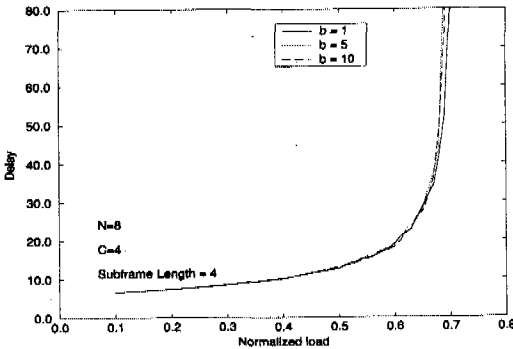


그림 17. 목적지 분포가 다를 때의 성능 ($N=8$)
Fig 17. Delay vs. load under asymmetric destination distribution ($N=8$)

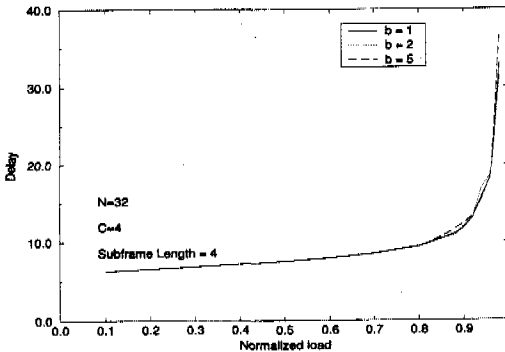


그림 18. 목적지 분포가 다를 때의 성능 ($N=32$)
Fig 18. Delay vs. load under asymmetric destination distribution ($N=32$)

VI. 결론 및 향후 연구 방향

본 논문에서는 star coupler 토폴로지의 WDM LAN에서 조율 속도가 느린 가변 수신기를 사용하면서도 WDM의 대역폭을 효과적으로 사용할 수 있는 매체 접근 제어 프로토콜에 대해 제안하고 그 성능을 평가하였다. 각 노드는 2개의 가변 수신기를 갖고 하나의 수신기를 통해서 데이터를 수신하고 있는 동안 나머지 하나는 또 다른 채널로 조율하게 하는 파이프라인 방식으로 소자의 느린 동작을 해결하였다. 또한, 수신 소자가 하나의 채널에서 다른 채널로 조율하는데 걸리는 시간을 하나의 서브프레임으로 잡고 하나의 서브프레임은 여러 개의 타임 슬롯으로 구성되도록 하여, 하나의 서브프레임 내에서 같은 채널로

여러 개의 패킷 전송이 이루어지도록 하였다. 이러한 서브프레임에서 목적지 충돌과 송신지 충돌이 없도록 트래픽 행렬을 스케줄링하는 알고리즘을 제안하였다. 노드의 수를 N , 채널의 수를 C 라 할 때, 제안된 알고리즘의 시간 복잡도는 $O(N^2C)$ 로 주어지는데, $C \ll N$ 인 경우에는 시간 복잡도가 최적 알고리즘의 시간 복잡도인 $O(N^2)$ 에 근사함을 확인하였다.

또한, 제안된 스케줄링 알고리즘의 지연 시간 성능을 시뮬레이션을 이용하여 분석하였다. 분석 결과 채널의 수와 서브프레임의 길이가 고정되어 있을 때는 노드의 수가 많을수록 스케줄링이 잘되어 지연 시간 성능도 좋아지며, 노드의 수와 채널의 수가 고정되어 있는 구성에서는 서브프레임의 길이가 길수록 스케줄링이 불완전하게 되어 지연 시간 성능도 나빠진다. 또한, 패킷의 목적지를 같은 확률로 선택하는 대칭적 목적지 분포에 비해 목적지 선택 확률이 다른 비대칭적 목적지 분포에서도 지연 시간 성능은 크게 저하되지 않는다. 또한, 클라이언트-서버 모델과 같이 특정 노드나 그룹이 트래픽을 많이 발생시키는 비대칭 구성에서 지연 시간 성능이 상대적으로 저하되기는 하지만 크게 저하되지 않는다는 특성을 보여준다.

제안된 알고리즘은 현재의 소자 기술 수준에서 고속으로 동작하는 고가의 가변 소자를 사용하지 않고 저속의 가변 소자만으로도 효과적인 초고속 LAN을 구성하는 데에 사용될 수 있으리라 기대된다. 현재 하나의 star coupler가 제공하는 파장(wavelength)의 수를 한계를 극복하기 위하여 star coupler들을 여러 개 연결하여 계층 구조로 구성하는 방안이 대해 연구되고 있는데, 본 논문에서 제안된 방법을 이러한 다계층 star coupler 구조에 확대 적용하는 방안이 대해 앞으로 연구할 계획이다.

참고 문헌

- [1] B. Mukerjee, "WDM-based local lightwave networks, part I: single hop systems," *IEEE Network Mag.*, pp.12-27, May 1992.
- [2] B. Mukerjee, "WDM-based local lightwave networks, part II: multihop systems," *IEEE Network Mag.*, pp.20-32, July 1992.
- [3] R. Ramaswami, "Multiwavelength lightwave networks for computer communication," *IEEE Communications Mag.*, pp. 77-88, Feb. 1993.
- [4] K. M. Sivaligam, J. Wang, "Performance of a MAC protocol for WDM networks with on-line

scheduling,” INFOCOM'96, pp. 1234- 1241, 1996.

- [5] Nen-Fu Huang and Huey-Ing Liu, “On the isochronous and asynchronous traffic scheduling problem for single-star WDM networks,” ICC'96, pp. 1766-1770, 1996.
- [6] C. A. Bracket, “Dense wavelength division multiplexing networks: principle and applications,” IEEE J. Select. Areas Commun., vol. SAC-8, no. 6, pp.948-964, Aug. 1990.
- [7] Aura Ganz and Yao Gao, “A time-wavelength assignment algorithm for a WDM star network,” INFOCOM'92, pp. 2144-2150, 1992.
- [8] _____, “Traffic scheduling in multiple WDM star systems,” ICC'92, 1992.
- [9] G. N. Roukas and V. Sivaraman, “On the design of optimal TDM schedules for broadcast WDM networks with arbitrary transceiver tuning latencies,” INFOCOM'96, pp. 1217-1224, 1996.
- [10] Ming Chen and Tak-Shing Yum, “A Conflict-free protocol for optical WDMA networks,” GLOBECOM'91, pp.1276-1281, 1 991.
- [11] Milan Kovacevic and Mario Gerla, “Analysis of a T/WDMA scheme with subframe tuning,” ICC'93, pp. 1239-1244, 1993.
- [12] M. H. MacDougall, Simulating Computer Systems: Techniques and Tools, The MIT Press, 1987.
- [13] R. Chipalkatti, Z. Zhang and A. S. Acampora, “High Speed Communication Protocols for Optical Star Coupler Using WDM,” INFOCOM'92, pp. 2124-2133, 1992.

백 선 욱(Seonuck Paek)

정회원



1981년 3월~1986년 2월 : 서울대학교 공과대학 컴퓨터공학과 (학사)

1987년 3월~1989년 2월 : 서울대학교 공과대학 컴퓨터공학과 (석사)

1989년 3월~1994년 8월 : 서울대학교 공과대학 컴퓨터공학과(박사)

1996년 4월~1996년 12월 : 한국 전자통신 연구소

1995년 3월~현재 : 상명대학교 전자계산학과 조교수
<주관심분야> WDM 통신망, 초고속 통신망, 네트워크 보안