

광 버스트 스위칭 네트워크에서 Look-Ahead 알고리즘을 적용한 그룹 스케줄링의 성능 분석

정희원 신 중 덕*, 학생회원 장 원 석*

Performance Analysis of Group Scheduling with Look-Ahead Algorithm for Optical Burst Switching Networks

Jong-Dug Shin* *Regular Member*, Won-Seok Jang *Student Member*

요 약

본 논문에서는 광 버스트 스위칭 네트워크의 코어 노드에 그룹 스케줄링을 적용한 경우 발생하는 데이터 버스트 처리 지연시간의 감소를 위한 알고리즘을 제안하였다. 이 알고리즘에서는 일차 스케줄링 세션 종료 직후에, 스케줄링 창 내의 모든 빈 공간(void) 정보를 나타내는 표를 생성하기 때문에, 직전 스케줄링에서 예약에 실패한 데이터 버스트들을 다른 채널의 빈 공간에 재할당할 때, 이 표를 활용하여 효과적으로 버스트 처리 시간 지연을 감소시킬 수가 있다. 이 알고리즘이 적용된 그룹 스케줄링 방식과 기존의 그룹 스케줄링 방식을 비교한 결과, 채널 이용률에 있어서는 서로 거의 동일한 성능을 보이나, 버스트 손실 확률과 파장 변환률은 알고리즘을 적용한 그룹 스케줄링 방식이 기존의 그룹 스케줄링 방식보다 약간 우수하였다. 버스트 처리 시간에 있어서는 부하가 약 0.6이하인 영역에서는 기존 그룹 스케줄링 방식이 약간 빠른 반면에, 부하가 0.8이상인 영역에서는 알고리즘을 적용한 방식의 처리 시간이 현저히 짧게 나타났다, 부하 0.9에서 약 1/2.1로 감소하였다.

Key Words : Optical Burst Switching, Optical Packet Switching, Group Scheduling, Void Time, Look-Ahead Algorithm

ABSTRACT

There has been proposed an algorithm to reduce data burst processing delay in group scheduling in core nodes of optical burst switching networks. Since, in this algorithm, look-up tables containing all the void time information in scheduling windows are generated as soon as the primary group scheduling session terminates, it becomes faster to reassign dropped data bursts to proper voids in different data channels by referring to the tables. The group scheduling with this algorithm showed almost the same channel utilization as the one without using the algorithm but performed a little better in both burst loss probability and wavelength conversion rate. On the other hand, per-burst processing time has been reduced dramatically in the load region of higher than 0.8, showing a factor of 2.1 reduction at 0.9.

I. 서 론

광 버스트 스위칭(Optical Burst Switching; OBS)

은 광 회선 스위칭(Optical Circuit Switching; OCS)보다 더 높은 채널 이용률을 얻을 수 있으므로 코어 네트워크의 진화과정에서 중장기적으로 연구되고

※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌습니다.

* 숭실대학교 정보통신전자공학부 (jdshin@ssu.ac.kr)

논문번호 : KICS2006-09-390, 접수일자 : 2006년 9월 22일, 최종논문접수일자 : 2006년 11월 10일

있는 기술이다^[1]. OBS의 핵심 개념 중의 하나는 제어 정보와 데이터 정보의 분리이다. 데이터 채널은 사용자 데이터 버스트(Data Burst; DB)만을 전달하는 데 사용되며, 라우팅 정보 및 자원 요청과 같은 정보는 버스트 헤더 패킷(Burst Header Packet; BHP)에 포함되어 제어 채널을 통해 전송된다. 그러므로 네트워크 운용을 위해서는 서로 다른 전송 채널을 통해 전달되는 DB와 해당 BHP간에 적절한 조율이 필요하게 된다. 결국, OBS 네트워크에서 해결해야 할 문제 중에 하나는 채널 이용률을 극대화 하면서 DB와 해당 BHP 간의 조율을 적절히 수행할 수 있는 스케줄링 방식을 고안하는 것이다.

현재까지 제안된 OBS 코어 노드를 위한 채널 스케줄링 방식들은 수평 스케줄링(Horizon scheduling) 방식과^[2] 지연 예약(Delayed reservation) 방식이 있다. 초기에 제안된 수평 스케줄링 방식은 코어 노드에 도착한 BHP의 제어 정보에 의해 스케줄러가 채널 예약을 시도하는 데, 만일 예약이 가능하다면 스케줄러가 예약 결정을 내린 시점부터 자원을 즉시 예약하여 해당 DB의 전송이 완료될 때까지 자원 예약 상태를 계속 유지한다. 이와 같은 시스템은 채널의 최근 사용 시간 정보만을 갖고 있기 때문에 간단하다. 하지만, 채널 예약 요청이 수락되는 경우에는 DB가 노드에 도착하기도 전에 자원을 예약하므로, 이미 예약되어 있는 전송 시간들 사이의 빈 공간(void)에 다른 DB를 삽입할 수가 없기 때문에 실제 서비스에 필요한 시간보다 더 많은 시간이 DB 전송에 할당되는 단점을 갖고 있다. 그 후에 제안된 지연 예약방식에는 Just-In-Time(JIT)과 Just-Enough-Time(JET) 프로토콜이 있다^[3,4]. 이 방식은 수평 스케줄링 방식과는 달리 DB가 코어 노드에 도착하는 시점부터 DB의 길이 동안만 채널 자원을 할당한다. 만약 이미 예약된 DB들 사이에 빈 공간이 존재하면, 이후에 도착하는 DB의 크기에 해당하는 시간이 이 빈 공간과 같거나 작으면 새로 도착하는 DB를 이미 예약된 DB들 사이에 삽입할 수도 있다. 따라서, 이 방식은 수평 스케줄링보다 자원을 효율적으로 사용할 수 있다.

최근에 새로운 OBS 스케줄링 방식으로서 그룹 스케줄링 방식이 제안되었다^[5]. 이 방식은 기존에 제안된 방식들처럼 입력 BHP의 정보를 이용하여 순서대로 해당 DB를 개별적으로 스케줄링하는 것이 아니라, 일정한 크기의 시간 창 안에 BHP들을 수집한 후, 이 BHP들을 동시에 처리하여 나중에 도착할 해당 DB들을 한꺼번에 지연 스케줄링하는

방식이다. 이 방식은 스케줄링 창 안에 수집된 BHP들을 그룹으로 처리하기 때문에 스케줄링할 DB 그룹의 정보를 모두 알 수 있어서 보다 효율적으로 네트워크 자원을 이용할 수 있다. 이와 같은 방식은 기존의 방식들보다 자원 이용률이 높고 버스트 손실 확률이 낮은 장점을 갖고 있다. 다채널 입/출력 광섬유 포트와 각 출력 포트에 파장 변환기가 장착된 OBS 코어 노드의 경우에^[6], 그룹 스케줄링이 즉시 스케줄링 보다 채널 이용률과 버스트 손실 확률이 약간 향상되는 것을 보였다. 반면에 파장 변환률은 급격히 감소하였다. 이 같은 결과는 그룹 스케줄링의 지연 스케줄링 속성에 기인한다. 그러나, 스케줄링을 위해 수집된 BHP들로부터 스케줄링 정보를 수집하는 시간으로 인해 노드에서 버스트 한 개를 처리하는 데 걸리는 평균 시간이 증가하였다. 또한, 초기 스케줄링에서 실패한 DB들을 다른 채널의 빈 공간에 재할당하는 방법의 비효율성도 버스트 처리 시간을 증가시키는 요인이 되었다. 예를 들어 포트 당 4채널을 사용하는 경우에, 기존의 알고리즘에서는 손실된 DB의 파장이 λ_1 일 경우에는 해당 포트의 λ_2 스케줄링 창에서 이 DB를 스케줄링 할 수 있는 시간 상의 빈 공간(void)이 있는지 먼저 찾아보고, 만약 없을 경우에는 순차적으로 λ_3 , λ_4 스케줄링 창에서 빈 공간을 찾게 된다. 마지막까지 찾지 못하면 이 DB는 손실된다. DB 파장이 λ_2 일 경우에는 λ_3 , λ_4 , λ_1 스케줄링 창의 순서로 빈 공간을 검색한다. 다른 채널의 DB들도 마찬가지로 방법으로 빈 공간을 검색한다. 이와 같은 알고리즘의 문제점은 동일 포트 내에서 파장 번호 순으로 스케줄링 창의 빈 공간을 검색하기 때문에, 만일 재할당할 DB의 파장이 λ_1 이고, 이 DB에 적합한 빈 공간이 λ_4 스케줄링 창에 있을 경우에도 λ_2 , λ_3 파장의 스케줄링 창을 검색하여야 하므로 스케줄링 시간이 증가한다.

본 논문에서는 그룹 스케줄링에서 평균 버스트 처리 시간을 줄이기 위해 새로운 DB 재할당 알고리즘을 제안하였다. 이 알고리즘에서는 초기 스케줄링 세션 종료 직후에 각 포트의 모든 빈 공간 정보를 포함하는 표를 생성한다. 이 표를 활용하면 직전 스케줄링에서 할당되지 못했던 DB들을 적절한 빈 공간에 기존의 그룹 스케줄링 방식보다 빠르게 재할당하는 것이 가능하다. 본 논문의 구성은 다음과 같다. 제 2장에서는 그룹 스케줄링을 위한 Look-Ahead 알고리즘에 관하여 간단히 설명하고, 제 3장에서는 입출력 포트가 각각 4개, 포트 당 데이터 채널 개

수와 제어 채널 개수가 각각 4개와 1개, 그리고 광 스위치 패브릭(Optical Switch Fabric)의 각 출력 포트에 파장 변환기가 장착된 코어 노드에서, 이 알고리즘이 적용된 그룹 스케줄러의 성능을 전산 모의실험으로 측정하여 기존의 그룹 스케줄링 방식 및 직접 스케줄링 방식과 비교하였다. 성능 파라미터로서 버스트 손실 확률과 채널 이용률, 파장 변환기 사용 빈도인 파장 변환률, 그리고 버스트 처리 시간을 측정하였다. 마지막으로 제4장에서는 본 논문의 연구결과를 정리하였다.

II. LOOK-AHEAD 알고리즘

2.1 그룹 스케줄링

OBS 그룹 스케줄링 방식에서는 데이터 채널을 작은 시간 구간 단위를 갖는 데이터 채널 스케줄링 창(scheduling window)들로 분할하여 사용한다. 동일한 데이터 채널 스케줄링 창을 사용하려는 DB(Data Burst)들의 BHP(Burst Header Packet)들은 해당 BHP 수집 창에 수집된다. BHP 수집 창이 종료되면, 이 창에 수집된 모든 BHP들은 동시에 처리되기 때문에, 해당 DB 그룹이 한꺼번에 스케줄링된다. 이와 같이 반 오프라인(semi-offline) 모드로 동작하기 때문에 스케줄러가 BHP의 도착 순서에 따라 개별적으로 스케줄링 결정을 내리는 기존의 스케줄링 방식과는 달리, 수집 창 내의 모든 DB들에 대한 정보를 알 수 있으므로 보다 효율적으로 DB들을 스케줄링할 수 있다. 또한, BHP들을 모두 다 수집할 때까지 기다리는 오프라인 스케줄링과는 달리 데이터 채널 시간을 작은 스케줄링 창으로 분할하여, 각 창의 마지막에 스케줄링 결정을 내리기 때문에 효율적인 스케줄링을 할 수 있다.

DB는 데이터 채널 시간 축에서 DB의 길이에 해당하는 시간 선분으로 표현할 수 있으므로, 가변 길이를 갖는 DB들을 스케줄링하는 문제는 가변 길이의 시간 선분들을 직선 상에 배열하는 문제로 해석할 수 있다. DB의 집합이 해당 시간 선분들로 변환되면, 스케줄러는 가능한 많은 수의 중첩되지 않는 시간 선분들을 채널 시간 창에 채우게 된다. 스케줄러는 주어진 DB들의 집합에 대해 구간 표현(interval representation) 프로필을 구성하고, 이 프로필로부터 구간 그래프(interval graph)를 생성하여 주어진 시간 창에 중첩하지 않는 최대 개수의 구간들을 구한다. 이들 중첩이 되지 않는 구간들은 데이터 채널에서 서로 충돌하지 않고 전송될 수 있는

DB들을 의미한다.

2.2 LOOK-AHEAD 알고리즘 기본 개념

본 논문의 DB 재할당 알고리즘인 LAA(Look Ahead Algorithm)의 동작은 다음과 같다. 일차 스케줄링이 종료되면, 각 포트에서는 파장, 즉 채널별 스케줄링 창에서 DB가 할당되지 않은 빈 공간들에 대한 정보 표들이 생성된다. 이 표들에는 출력 광섬유 포트 번호, 출력 파장 번호, 빈 공간의 길이, 빈 공간의 시작 시간과 종료 시간에 관한 정보가 저장된다. 일차 스케줄링 세션에서 할당되지 못한 DB들은 채널별, 즉 파장별로 한 개씩 할당되어 있는 버퍼에 각 DB의 길이와 시작 시간 및 종료 시간 정보가 저장된다. 버퍼에 저장되어 있는 DB 정보들은 다음 스케줄링 세션에서 빈 공간의 정보 표와 비교되고, 이 때 적절한 빈 공간이 검색되면 해당 DB들이 공간에 스케줄링한다. 만약 하나의 빈 공간에 다수개의 DB가 스케줄링이 되고자 할 경우에는 가장 긴 길이의 DB가 그 빈 공간을 예약하며, 나머지 DB들은 다른 채널에 적절한 빈 공간이 존재한다면, 그 공간들을 예약한다. 따라서, 종래의 알고리즘처럼 순차적으로 파장별 스케줄링 창의 빈 공간 정보를 참조하지 않고, 정보 표를 이용하면 동일 포트 내의 모든 빈 공간에 대한 정보를 한 번에 알 수 있어서 스케줄링 처리 시간을 감소시킬 수 있다.

2.2.1 빈 공간 정보 표 생성

그림 1은 일차 그룹 스케줄링이 끝난 후에 포트 1, 채널 1의 200μs 스케줄링 창에 예약된 DB들을 보여주고 있다. 빈 공간 정보 표 중에서 첫 번째 표 a에는 빈 공간의 번호, 빈 공간의 길이, 포트 번

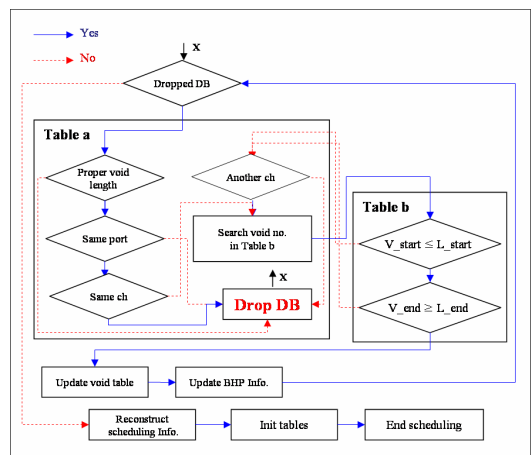
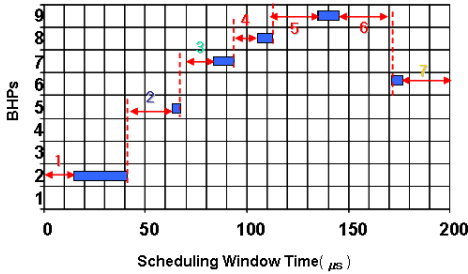


그림 1. 탈락한 DB 재할당 흐름도



Void No.	Length	Port No.	Ch. No.
1	15	1	1
2	25	1	1
3	17	1	1
4	10	1	1
5	20	1	1
6	25	1	1
7	25	1	1

Table a
그림 2. 빈 공간 정보 표 구성

Void No.	S_Time	E_Time
1	0	15
2	40	65
3	68	85
4	95	105
5	115	135
6	145	170
7	175	200

Table b

호 그리고 채널 번호 정보가 기록되는데, 이 표는 노드에 한 개만 생성된다. 표 b는 빈 공간 번호, 빈 공간의 시작시각, 빈 공간의 끝 시각에 대한 정보를 포함하며, 채널마다 한 개씩 생성된다. 여기서 빈 공간 번호는 고유번호이며, 표 a와 표 b를 연결시켜 준다. 빈 공간 번호는 중복되지 않는다. 스케줄링 창에 예약된 DB들의 정보를 이용하여 생성한 표 a와 b는 그림 2와 같다.

2.2.2 빈 공간 정보 표 검색

빈 공간 정보 표 검색 과정은 그림 1과 같다. 일차 스케줄링에서 탈락한 DB들에 대한 정보가 저장되어 있는 각 큐로부터 DB정보를 하나씩 읽어 들인 다음 이 정보를 이용하여 빈 공간 검색을 하게 된다. 만약 큐에 탈락한 DB가 존재할 경우에는 표 a를 참조하여 이 DB가 어느 빈 공간에 할당될 수 있는지를 결정한다. 빈 공간의 길이를 우선 검색하고 그 다음에 포트 번호, 채널 번호 순서로 검색한다. 이 조건들을 만족하는 빈 공간 번호에 대해서만 표 b를 참조하여 DB의 재할당이 가능한지 판단한다. 표 b에 저장된 빈 공간의 시작 시각과 종료 시각에 대한 정보와 DB의 시작 시간과 끝 시간과 비교하여 DB를 여러 개의 빈 공간에 재할당 가능하다면 그 중에서 가장 작은 빈 공간을 선택하여 예약하고, 만약 가능한 빈 공간이 없으면 이 DB는 탈락 처리한다. 이와 같이 한 DB가 재할당되면 표 a와 표 b의 빈 공간 정보를 갱신한다. 다음 DB는 갱신된 표를 참조하여 큐에 저장되어 있는 모든 DB에 대하여 위와 같은 과정을 반복한다.

이상의 스케줄링이 끝나면 스케줄링 창에 새로 추가된 DB들의 정보를 갱신하고, 표 a와 b는 초기화된다. 큐에 처리할 DB들이 저장되어 있지 않을 경우에는, 재할당할 DB가 없는 것이므로 표들을 초기화하고, 스케줄링을 종료한다.

2.2.3 빈 공간 정보 표 다중 검색 문제

빈 공간 정보 표를 생성한 것은 빠른 검색과 정확한 빈 공간의 위치를 찾기 위해서이다. LAA에서는 두 개의 표를 사용하여 검색하므로 문제가 발생할 수도 있다. 표의 개수를 증가시키면 알고리즘이 복잡하게 되어 관리와 구현이 어렵게 되지만, 표의 개수를 줄여 하나로 통합할 경우에는 다중 검색 문제가 발생할 수도 있다. 왜냐하면, 동일한 표를 각기 다른 프로세서에서 참조하여 각자의 조건에 맞는 빈 공간을 찾을 경우에 어떤 빈 공간 한 개가 다른 프로세서들의 조건을 모두 만족시킨다면 하나의 프로세서에만 할당해야 한다. 본 알고리즘에서는 이와 같은 문제를 다음과 같이 해결한다.

그림 3에서 DB(A)는 포트 1, 채널 1, 길이가 15 μs, 시작 시각이 20 μs, 종료 시각이 35 μs이고, DB(B)는 포트 1, 채널 3, 길이는 20 μs, 시작 시각이 20 μs, 종료 시각이 40 μs이다. 이때 각각의 DB는 표 a에서 검색된 빈 공간 번호 DB(A)의 경우 3, 4, 5, 6, 7, 8 이고, DB(B)의 경우 1, 4, 5, 6, 9를 검색했다고 가정하자.

DB들은 검색된 빈 공간의 크기를 오름차순으로 정렬하여 표 a에서 검색한다. 표 a에서 조건에 맞는 빈 공간 번호가 각각 4, 5, 6번이라고 하면, DB(A)와 DB(B)는 중복 검색 결과를 갖는다. 이 경우에는 DB의 크기를 비교하여 가장 큰 DB를 먼저 가장 작은 빈 공간에 할당하고, 그 다음으로 큰 DB는 두 번째로 작은 빈 공간에 할당한다. 따라서, 본 예에서는 빈 공간 번호 4와 5번이 선택된 후 표 b에서

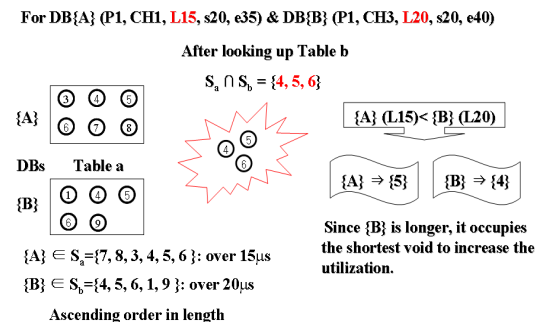


그림 3. 빈 공간 정보 표 다중 검색 문제 해결 방안

4, 5번의 시작 및 종료 시각을 확인하여 DB(B)를 빈 공간 4번에, DB(A)를 빈 공간 5번에 할당함으로써 다중 검색 문제를 해결하였다.

III. LOOK-AHEAD 알고리즘이 적용된 그룹 스케줄링의 성능

그림 4는 본 연구에서 사용한 OBS 코어 노드의 구조이다. 입·출력 포트가 각각 4개씩 있고, 각 포트에 연결된 광섬유에 1개의 제어 채널과 4개의 데이터 채널이 있으며, DB를 위한 광 버퍼는 없다고 가정하였다. 버스트의 길이와 도착시간은 지수 분포를 가정하였다. 본 논문에서 제안한 Look Ahead Algorithm이 적용된 그룹 스케줄링(Group Scheduling-Look Ahead Algorithm; GS-LAA)의 성능 비교 대상으로는 즉시 스케줄링(Immediate Scheduling; IS)과 기존의 그룹 스케줄링(Group Scheduling; GS)을 선택하였다. 전산모의 실험에 사용한 입력 파라미터는 표 1과 같다.

일차 스케줄링 세션 종료 후에 탈락되어 각 채널의 전기적 버퍼에 저장되어 있는 DB들은 본 논문에서 고려된 스케줄링 방식을 사용하여 재할당하였다. 전산모의 실험을 위해서 2.8 GHz Xeon CPU 두 개와 2GB 메모리가 장착된 Linux 서버를 사용하였다.

그림 5는 버스트 손실 확률 그래프이다. GS와 GS-LAA 모두 전 부하 범위에 걸쳐 IS방식보다 우수함을 나타내고 있다.

부하 범위 0.25~0.4의 경우에는 GS가 IS와 비교하여 약 20%이상 성능이 향상되었으며, GS-LAA의 경우에는 IS보다 약 30%이상 개선된 결과를 보이고 있다. 반면에, 부하 범위 0.5~1.0의 범위에서는 GS와 GS-LAA 모두 IS 방식보다 평균적으로 약 10% 정도 개선되었다.

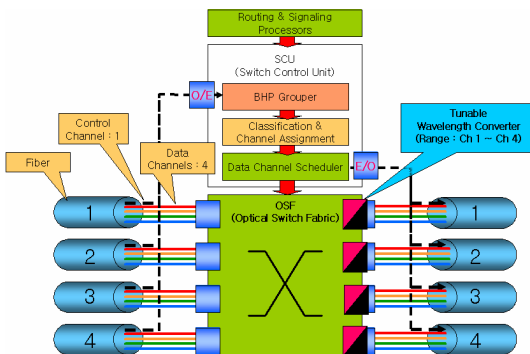


그림 4. OBS 코어 노드 구조

표 1. 전산모의 실험 입력 파라미터

No. of Control Channels/port	1
No. of Input/Output Ports	4
No. of Data Channels/Port	4
Burst Generator	On-Off Source
Burst Length and Burst Arrival	Exponential Distribution
Channel Bandwidth	10 Gb/s
Mean Burst Size	20 kB
Maximum Burst Length	30 μ s
Scheduling Time Window Size	200 μ s
No. of Windows	600
Range of Offered Load	0.1~1.0

이 결과는 그룹 스케줄링 방식이 스케줄링 정보 즉 BHP들을 모아서 DB들을 한꺼번에 스케줄링하기 때문에 버스트 손실 확률을 감소시킬 수 있음을 보이는 것이다. 특히, 부하가 낮을 경우에 GS-LAA가 현저한 성능 개선을 나타내는 것은 재할당 알고리즘이 이 영역에서 매우 효과적이기도 하며, 부분적으로는 버스트 길이의 제한, 본 논문에서는 버스트의 최대 길이를 30 μ s로 제한한 때문이기도 하다.

그림 6은 채널 이용률을 나타내고 있다. 비교한 세가지 스케줄링 방식들은 모든 부하 영역에서 모

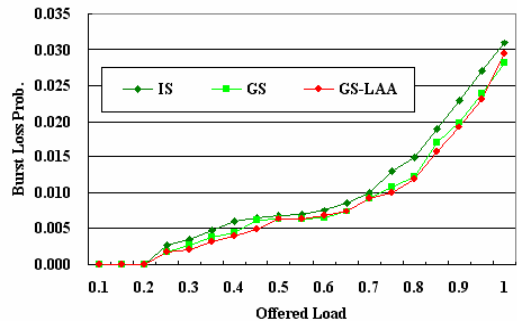


그림 5. 버스트 손실 확률

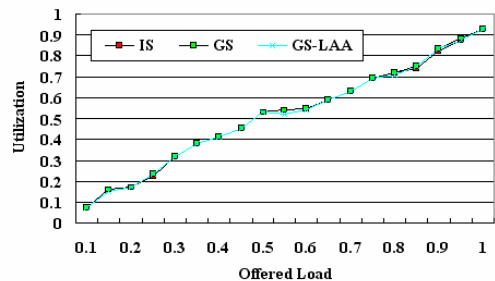


그림 6. 채널 이용률

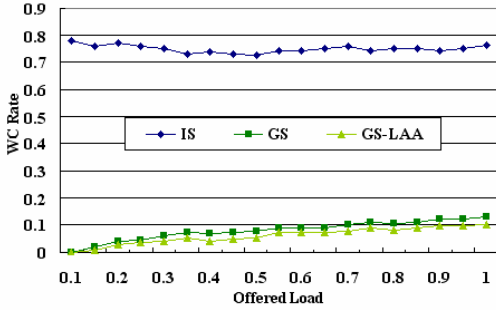


그림 7. 파장 변환률

두 거의 같은 채널 이용률을 보이고 있다. 단일 출력 채널의 경우에는 GS가 IS 보다 약 5% 개선된 결과를 보였었다⁷⁾. 본 논문과 같은 노드 구조를 사용하고, 버스트 길이에 대한 제한이 없는 경우에는, GS의 채널 이용률이 IS 보다 약 1% 정도 개선되었다. 이와 같은 채널 이용률의 변화는 노드에 사용된 파장 변환기에 의한 것으로서, 파장 변환기를 사용하면 직전 스케줄링 세션에서 탈락된 DB들을 재할당할 수 있는 가능성이 높아졌기 때문이다. IS의 경우에는 초기 스케줄링 세션에서 버스트 손실 확률이 보다 높기 때문에 다음 스케줄링 세션들에서 적절한 빈 공간을 찾을 확률이 당연히 높게 된다. 반면 GS의 경우에는 초기 스케줄링 세션에서 이미 거의 최적 스케줄링이 이루어졌기 때문에 GS는 IS의 경우와 마찬가지로 채널 이용률을 향상시킬 수 없다. 결과적으로 스케줄링 후에는 IS와 GS 모두 거의 동일한 채널 이용률을 보이게 되는 결과를 초래한다. GS-LAA의 경우에는 채널 이용률이 GS의 경우와 거의 동일하다. 이는 GS나 GS-LAA 모두 기본적으로는 동일한 그룹 스케줄링 방식이기 때문이다.

그림 7은 한 개의 DB가 수락되기 위해서 겪어야 하는 평균 파장 변환 횟수인 파장 변환률을 나타내고 있다. 두 그룹 스케줄링 모두 모든 부하 영역에서 IS보다 월등히 우수한 것을 볼 수 있는데, IS의 경우에 파장 변환률이 0.73~0.78사이로 나타난다. 그러나 GS와 GS-LAA의 경우에는 파장 변환률이 0~0.13과 0~0.1로 각각 나타났다. GS와 GS-LAA 간의 차이는 약 3%이하인 데 그 이유는 다음과 같다. IS 스케줄러는 버스트들을 순차적으로 하나씩 스케줄링한다. 이와 같은 스케줄링은 최적의 스케줄링 방식이 아니므로 초기 스케줄링 세션에서 보다 많은 빈 공간을 남긴다. 다음 스케줄링 세션에서는 초기 스케줄링 세션에서 탈락한 DB들을 파장 변환

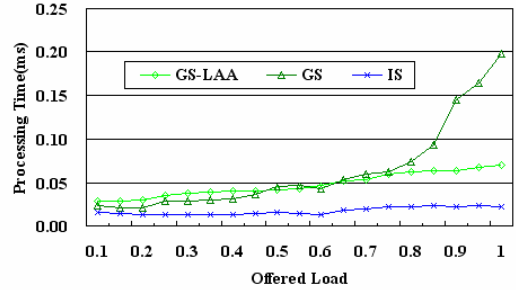


그림 8. 버스트 처리 시간

을 통해 빈 공간에 재할당하여야 하므로 그룹 스케줄링 방식보다 파장 변환 횟수가 증가하게 된다. GS-LAA의 경우에는 빈 공간 정보 표를 사용하여 초기 스케줄링 세션에서 탈락한 DB들을 스케줄링한다. 우선, DB의 길이 정보가 판독된 후 동일 출력 광섬유 포트의 스케줄링 창들에 있는 각 빈 공간과 비교하여 DB가 재할당될 수 있는 빈 공간이 존재하는지를 판단한다. 일단, DB의 길이보다 긴 빈 공간들이 발견되면, 빈 공간의 파장들이 DB의 파장과 비교된다. 만약 파장이 서로 같으면, 이 DB에 대한 재할당 절차가 종료된다. 그렇지 않으면, DB의 시작과 종료 시간이 각 빈 공간의 시작과 종료 시간과 비교된다. 만일 어떤 빈 공간이 이 DB에 맞는 경우에는, 이 빈 공간이 이 DB를 위해 예약된다. 각 채널별로 스케줄링 창이 존재하므로 각 포트에는 네 개의 프로세서가 병렬로 돌아간다. 하나의 빈 공간이 서로 다른 채널의 여러 DB들에 의해 요구되는 경우가 있을 수 있다. 이 경우, GS-LAA는 포트의 빈 공간에 관한 모든 정보를 제공한다. 이것은 적절한 빈 공간을 찾기 위하여 하나의 파장에서 다음 파장으로 차례대로 스케줄링 창을 들여다 볼 필요가 없다는 것을 의미하므로 GS-LAA에서는 파장 변환기의 사용이 감소된다.

마지막으로 그림 8은 코어 노드에서 DB 한 개를 채널에 할당하는 데 드는 평균 시간을 나타내고 있다. IS의 경우에 처리시간은 0.013~0.024 ms이고, GS-LAA의 경우에는 0.028~0.070 ms, GS의 경우에는 0.021~0.198 ms로 측정되었다. IS에 대한 GS의 비율은 1.5~9.0정도이고, IS에 대한 GS-LAA의 비율은 1.9~3.5로 측정되었다. GS가 DB를 처리하는 시간이 부하가 높은 영역에서는 IS보다 특히 길다. 그러나, GS-LAA의 경우에는 GS에 비해 처리 시간이 현저히 짧은 결과를 보이고 있다. 그 이유는 GS-LAA에서는 GS와 같이 전체적인 검색 을 하지

않고 빈 공간 정보 표를 이용하여 필요한 부분만을 검색하기 때문에 보다 빨리 빈 공간을 찾을 수 있기 때문이다.

V. 결론

본 논문에서는 OBS 네트워크의 코어 노드에 그룹 스케줄링을 적용하였을 때 발생하는 버스트 처리 시간 지연을 감소시키기 위한 Look-Ahead 알고리즘을 제안하였다. 이 알고리즘에서는 같은 포트의 모든 빈 공간 정보를 포함하는 표들이 생성되기 때문에, 기존의 그룹 스케줄링 방식보다 빠르게 탈락된 DB들을 재할당할 수가 있다. 그룹 스케줄링, 즉시 스케줄링, 그리고 Look-Ahead 알고리즘이 적용된 그룹 스케줄링 방식들의 성능을 다채널 입출력 포트와 각 출력 포트에 한 개씩의 파장 변환기가 장착된 코어 노드에 대하여 전산 모의실험을 통해 비교 평가하였다. 실험 결과 GS-LAA 그룹 스케줄링은 채널 이용률과 파장 변환률의 경우에는 GS와 거의 같은 성능을 보였다. 반면에, 버스트 당 처리 시간은 0.8이상의 부하 범위에서는 현저히 감소하였으며, 버스트 손실 확률은 보다 낮은 트래픽 환경에서 개선되었다.

참 고 문 헌

- [1] T. S. El-Bawab and Jong-Dug Shin, "Optical Packet Switching in Core Networks: Between Vision and Reality", IEEE Communications Magazine, Vol. 40, No. 9, pp. 60-65, September 2002.
- [2] J. Turner, "Terabit Burst Switching," Journal of High Speed Networks, Vol. 8, No. 1, pp. 3-16, 1999.
- [3] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) - a New Paradigm for an Optical Internet", Journal of High Speed Networks, Vol. 8, No. 1, pp. 69-84, 1999.
- [4] J. Y. Wei and R.I. McFarland, "Just-In-Time Signaling for WDM Optical Burst Switching Networks", IEEE Journal of Lightwave Technology, Vol. 18, No. 12, pp. 2019-2037, December 2000.

- [5] Saravut Charcranoon, Tarek S. El-Bawab, Hakki C. Cankaya, and Jong-Dug Shin, "Group-Scheduling for Optical Burst Switched (OBS) Networks," Globecom 2003, Vol. 5, pp. 2745-2749, San Francisco, A, Dec.1-5, 2003.
- [6] Jong-Dug Shin, Jae-Myung Lee, and Hyung-Suk Kim, "Performance of Group Scheduling in Optical Burst Switching Networks," OECC/COIN 2004, pp. 120-121, Yokohama, Japan, July 12-16, 2004.
- [7] 신종덕, "OBS 네트워크의 효율적 채널 이용을 위한 구간 스케줄링 방식," 전자공학회논문지, 제 40권, TC편, 제 10호, pp. 51-58, 2003년 10월.

신 종 덕 (Jong-Dug Shin)

정회원



1976년~1981년 연세대학교 전자공학과 (공학사)

1984년~1988년 University of Texas at Austin, 전기 및 컴퓨터공학과(공학석사)

1988년~1991년 Texas A&M University, 전기공학과

(공학박사)

1980년~1983년 대영전자공업 (주) 개발부, 계장

1991년~1995년 한국전자통신연구소, 선임연구원

2001년~2002년 Alcatel USA 방문교수

1995년~현재 숭실대학교, 정보통신전자공학부 부교수
<관심분야> OBS, OPS, OLS, 광 실시간 지연선로 빔
성형망, 광 센서 신호처리

장 원 석 (Won-Seok Jang)

학생회원



1997년~2004년 백석대학교
정보통신학부 (공학사)

2004년~2006년 숭실대학교
대학원 정보통신공학과
(공학석사)

2006년 2월~7월 (주)애니원 21
<관심분야> OBS, System

Integration