

광 인터넷에서 버스트 크기 제어 알고리즘 기반 서비스 차등화 기법

이용규*

Service Differentiation Scheme Based on Burst Size Controlling Algorithm in Optical Internet

Yonggyu Lee*

*Professor, Department of Electronic Engineering, Honam University, Gwangju, 62399 Korea

요 약

5G 서비스와 개인용 스마트 기기의 보급 확대는 트래픽의 폭증과 다양한 서비스에 대한 수요로 이어졌다. 이런 사실은 다시 네트워크 대역폭에 대한 거대한 수요로 이어졌다. 그러나, 전기 신호를 사용하는 기존의 네트워크 기술들은 이러한 수요를 감당하는 데 한계에 다다랐다. 이에 이러한 요구를 수용하기 위해, 광 인터넷이 활발하게 연구되고 있다. 그러나, 광 인터넷도 여전히 해결해야 할 많은 문제점을 가지고 있고, 이러한 문제점 중에서 가장 시급한 문제는 QoS 기술을 개발하는 것이다. 그러므로 광 인터넷-특히 OBS 네트워크-에서 클래스간 서비스 차별화를 달성하기 위해, 데이터 버스트의 크기를 동적으로 제어하는 새로운 QoS 기법이 본 논문에서 제안된다. 특히 본 논문에서는 광 지연 라인을 기반으로 하는 알고리즘을 제안한다.

ABSTRACT

The supply expansion of 5G services and personal smart devices has caused the sharp increase of data traffic and the demand of various services. Again, these facts have resulted in the huge demand of network bandwidth. However, existing network technologies using electronic signal have reached the limit to accommodate the demand. Therefore, in order to accept this request, optical internet has been studied actively. However, optical internet still has a lot of problems to solve, and among these barriers a very urgent issue is to develop QoS technologies. Hence, in order to achieve service differentiation between classes in optical internet, especially in OBS network, a new QoS method automatically tuning the size of data bursts is proposed in this article. Especially, the algorithm suggested in this article is based on fiber delay line.

키워드 : FDL, OBS, 광 인터넷, 동적 버스트 크기, 서비스 품질

Keywords : Dynamic Burst Size, FDL, OBS, Optical Internet, QoS

Received 8 March 2022, Revised 18 March 2022, Accepted 23 March 2022

* Corresponding Author Yonggyu Lee(E-mail:lyonggyu@honam.ac.kr, Tel:+82-62-940-5935)

Lecturer, Department of Electronic Engineering, Honam University, Gwangju, 62399 Korea

Open Access <http://doi.org/10.6109/jkiice.2022.26.4.562>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

인터넷 프로토콜을 기반으로 한 실시간 멀티미디어 서비스와 매우 다양한 서비스와 응용들의 등장으로 전송 대역폭에 대한 수요가 폭증하고 있다. 이에 따라 광 서킷 스위칭(Optical Circuit Switching), 광 레이블 스위칭(Optical Label Switching), 광 패킷 스위칭(Optical Packet Switching)과 같은 WDM 광 통신 기술이 발전하고 있다. 기존의 광 네트워크에 WDM 기술의 접목으로 네트워크에서 활용할 수 있는 대역폭이 비약적으로 증가하고 있다.

한편 이러한 WDM 네트워크가 가지고 있는 광학 메모리 기술, 광 파장 스위칭, 광 변환 기술, 광 버퍼링 기술 등과 같은 제약점을 극복하기 위한 기술이 바로 광 버스트 스위칭(Optical Burst Switching, OBS) 기술이다. OBS 기술은 데이터는 광학 도메인에서 처리하지만, 데이터를 제어하는 신호는 전기 도메인에서 처리하는 하이브리드 기술이다[1, 2]. 이렇게 제어 신호 처리와 데이터 전달 도메인을 광-전-광(Optical-Electrical-Optical, O-E-O) 변환을 통해서 분리함으로써 광 기술의 문제점을 극복할 수 있다.

OBS 기술은 O-E-O 변환 없이 중간 노드에서 전기적인 병목 현상을 제거하고 버퍼링 없이 QoS(Quality of Service)를 보장하기 때문에 광 인터넷 백본 기술로 연구되고 있다 [1, 2]. OBS 네트워크에서는 제어와 데이터 평면 사이에 강한 분리가 존재하므로 네트워크 관리 용이성과 유연성이 뛰어나다. 이러한 동적 특성은 높은 네트워크 적응 능력과 확장성으로 이어지며, 이는 인터넷 처럼 동적인 데이터 트래픽 전송에 매우 적합하다.

그러나 이러한 장점에도 불구하고, 실용적인 OBS 네트워크를 구현하기 위해서는 여전히 해결해야 할 많은 도전적인 문제들이 남아 있다. 에지 노드에서 주된 문제는 버스트 오프셋 시간 결정 체계와 버스트 어셈블리 메커니즘이다. 반면에 코어 노드에서는 데이터 버스트 및 제어 패킷 스케줄링 메커니즘, 보호 및 복구 체계, 충돌 해결 방법이 필요하다. 그러나 가장 시급하고 중요한 문제 중 하나는 다양한 애플리케이션에 대한 QoS 제공 기술 개발이다 [3 - 7].

지금까지 다양한 QoS 메커니즘이 다음과 같이 제안되었다. [8]에서는 OBS 네트워크의 신호 전달 방식을 개선하기 위해 GMPLS를 이용해 QoS를 보장하는 기법

을 제안하고 있다. [9]에서 저자들은 데이터센터처럼 지역적으로 밀집하고 데이터 흐름이 많은 곳에서 QoS를 보장할 수 있는 기법을 제안했다. [10]에서는 각 클래스 별로 구분되는 파장을 할당하는 기법이 제안되었다. [11]에서 데이터 버스트 분할 방식을 기반으로 우선순위가 높은 버스트가 낮은 우선순위 버스트의 자원을 빼앗아 활용함으로써 QoS를 보장하고 클래스 간 분리를 지원하는 방식을 제안했다. [12]에서는 OBS에서 QoS를 제공하기 위해 블록킹과 관련된 여러 매개변수를 통합적으로 분석해 효율적으로 QoS를 제공하기 위한 방안을 제안했다. [13]에서는 OBS 네트워크에서 QoS를 제공하기 위해 파장을 이용한 기법을 제안했고, 이를 선형 계획법을 이용해서 분석했다. [14]에서는 서비스 차등화를 달성하기 위해서 광 지연 라인의 QoS 알고리즘이 제안되었다.

위에서 언급한 이전 연구를 포함한 대부분의 방식들은 QoS를 달성하기 위해 고정된 크기의 데이터 버스트를 사용했다. 그러므로 입력 데이터 트래픽의 조건이 변한다면, 최적의 결과를 유지하기가 어렵다. 이에 따라 본 논문에서는 입력 데이터 트래픽의 변화에 따라 OBS 네트워크에서 전송되는 데이터 버스트의 크기를 동적으로 제어함으로써 클래스 간 차등화를 달성하는 알고리즘을 제안하고자 한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 OBS 노드를 위한 다양한 버퍼링 구조를 설명하고, 3장에서는 동적으로 버스트 크기를 제어하기 위한 알고리즘을 설명한다. 이어서 4장에서는 3장에서 제안한 알고리즘의 성능을 분석하고, 마지막으로 5장에서는 본 논문의 결론을 제시한다.

II. 버퍼 구조

이 장에서는 OBS 네트워크를 위한 다양한 광 버퍼링 구조가 제시된다. 일반적으로, 광 버퍼는 그림 1과 같이 구성된다. 그림 1(a)는 두 개의 2x2 스위치와 하나의 광 섬유로 구성된 단일 단계 지연 라인 버퍼 구조를 보여준다. 그림에서 아래쪽 부분은 0의 지연 값을 가지는 FDL(Fiber Delay Line)을 의미하고 위쪽 부분은 일정한 크기의 임의 지연 값(D)이 있는 경우이다. 그러므로 위쪽으로 통과하는 데이터는 일정한 지연 값을 가지게 된다.

한편, 여러 개의 단일 단계 지연 라인 버퍼를 직렬방식으로 이어 붙여 가변적인 지연 값을 가지도록 구성한 광 지연 버퍼는 그림 1(b)에서 보여주고 있다. 실제로 가변적인 지연 값은 데이터 버스트가 통과하는 각 단계의 총합에 의해서 결정된다[15].

2.1. 버퍼 구조

그림 2는 이러한 지연 라인을 이용한 OBS 노드의 구성 예이다. 그림에서 FDL Buffers는 그림 1에서 보여준 지연 라인들을 여러 개 묶어서 구성한 장비이다. 그림처럼 각각의 파장은 많은 FDL로 구성된 전용 FDL Buffers를 사용하는 구조이다. 이 그림은 각 파장별 전용 FDL 버퍼가 데이터 버스트 출력단에 존재하는 전용방식 피드 포워드 출력 버퍼링 구조이다. 이 구조는 상당히 큰 버퍼를 사용할 수 있어 성능 향상에 매우 쉬운 구조이다. 그러나, 버퍼의 크기를 늘리면 FDL의 비용이 증가하기 때문에 실제로는 무한히 늘릴 수 없다. 또한 FDL을 구성할 때, D 값은 입력 트래픽, 대역폭, 평균 버스트 크기 등과 같은 요소에서 측정과 분석에 의한 최적값을 고려해서 결정한다. 이것은 이러한 요소 중에서 하나라도 변한다면 최적의 성능을 보장할 수 없음을 의미한다[16].

다음으로, 그림 3은 피드백 버퍼링 구조를 보여준다. 출력 포트에서의 출력이 버퍼가 있는 순환 루프를 통해 다시 입력 포트에 공급된다. 그림 3처럼 각 파장의 순환 루프에 포함되어있는 전용 FDL 버퍼를 가지고 있는 경우를 전용방식 피드백 출력 버퍼링 구조라고 한다. 하지만, FDL 버퍼가 일부 혹은 전체 파장에 공유되는 경우를 공유방식 피드백 출력 버퍼링 구조라고 한다. 피드백 방식의 경우 작은 FDL 버퍼를 가지고 충분한 지연 값을 제공할 수 있다는 장점도 있지만, 주된 단점은 데이터 버스트가 여러 번 재순환될 때 재순환이 신호 대 잡음비(SNR)를 감쇠시킬 수 있다는 것이다 [16]. 또한 충분한 지연 값을 얻기 위해서 입력 파장과 출력 파장이 달라지는 경우도 존재하고, 서로 다른 출력 포트에 향하는 데이터 버스트들이 시차를 두고 동일한 FDL 버퍼를 공유할 수 있기 때문에, 공유 방식 FDL 버퍼의 성능 분석은 물론 실제로 구현하는 것은 현실적으로 불가능하다[16].

공유방식 피드 포워드 출력 버퍼링 구조는 그림 4에서 보여주고 있다. 그림처럼 FDL 버퍼는 첫 번째 스위치와 두 번째 스위치 사이에서 위치하며 모든 파장에 의해 공유되고 있다[15]. 그림 4에서처럼 이 구조는 두 개

의 스위치를 사용한다. 그러므로 공유방식 피드백 버퍼링 구조와 마찬가지로 FDL 버퍼의 수는 현저하게 감소할 수 있지만, 공유방식 버퍼링 아키텍처와 동일하게 버퍼의 성능 분석은 현실적으로 불가능하다[16].

2.2. 스케줄링 방식

OBS 네트워크에서 버퍼 구조와 크기에 더해 OBS 네트워크의 성능에 영향을 주는 두 가지 요소는 다음과 같다[16]. 첫 번째 요소는 FDL 버퍼 내에 들어있는 각각의 지연 라인 값의 분포이다. 각각의 지연 라인이 일정한 주기를 가지고 선형적으로 증가하는 지연 값을 가지는 방식과 일정한 주기를 가지고 있는 것이 아닌 임의의 지연 값을 가지는 방식으로 FDL 버퍼를 구성할 수 있다. 일정한 주기를 가지고 선형적으로 증가하는 방식의 경우 임의의 지연 값(D)를 바탕으로 연속적인 배수의 형태로 증가한다.

두 번째 요소는 데이터 버스트 전달을 위한 스케줄링 정책이다. OBS 네트워크에서는 특정 출력 포트에서 한 버스트의 마지막 비트와 다음 버스트의 첫 번째 비트 사이의 공백이 발생할 수 있다. 이것은 FDL이 제한된 시간 동안의 저장(혹은 지연)만을 제공하기 때문에 발생한다. OBS에서는 공백 제거 방식과 공백을 제거하지 않고 전송하는 두 가지 스케줄링 방식을 활용한다.

공백 제거 방식의 경우 하나의 출력 포트에서 충돌이 발생하지 않는다면 입력 데이터 버스트가 임의의 지연 라인을 이용해 전송될 수 있다. 이렇게 함으로써 빈 공간을 제거할 수 있다. 하지만, 이것은 출력 포트에서의 선입선출(First-In-First-Out, FIFO) 규칙을 위반하는 문제를 야기할 수 있다. 일부 이전 연구들[17, 18]은 특정 조건에서 공백 제거 방식이 공백을 제거하지 않는 방식보다 효율적일 수 있음을 보여주었으나, 공백 제거 방식의 경우 데이터 버스트 순서 문제를 초래해 더 복잡한 오버헤드를 요구함을 보여주고 있다.

본 논문에서는 그림 2에서 제시된 전용방식 피드 포워드 출력 버퍼링 구조를 고려하고, 지연 라인이 일정한 D 의 값을 기본으로 증가하는 방식을 고려한다. 또한 공백을 제거하지 않는 스케줄링 방식을 사용해 새로운 QoS 차등화 알고리즘을 제안하고 성능을 분석한다.

III. 알고리즘 제안

본 논문에서 고려하는 FDL 버퍼는 각각의 지연 라인이 선형적으로 증가하는 서로 다른 지연 값을 가지도록 구성된다. 이때, 각각의 지연 라인이 가지는 지연 값은 D 를 중심으로 선형적으로 증가하며, 기본이 되는 지연 단위 D 는 평균적인 데이터 버스트 길이의 상대 비율로 표현한다. FDL 버퍼는 하드웨어 시스템이기 때문에, 각 지연 라인의 지연 시간 값은 특정 값으로 계산된다. 즉 i 번째 지연 라인 값은 $i \times D$ 가 된다. 그러나 [14]에서처럼 이렇게 구성된 FDL 버퍼는 입력 IP 패킷이 변하면 최적의 성능을 보장할 수 없는 것이다. [14]의 방식은 입력 IP 패킷이 증가함에 따라 대다수를 차지하는 가장 낮은 클래스인 best-effort 트래픽을 거의 전송할 수 없었고, 우선순위가 낮을수록 해당 클래스 전송에 필요한 지연 라인 수의 변동 폭이 상당히 큼을 보여주고 있다.

따라서 본 논문에서는 입력 IP 패킷의 변화에 따라 동적으로 데이터 버스트의 길이를 제어함으로써 이러한 문제를 해결하기 위한 새로운 서비스 차등화 알고리즘을 제안한다.

3.1. 알고리즘

[14]의 알고리즘은 하위 FDL 그룹이 데이터 버스트 전송에 성공하고 그룹에 충분한 수의 지연 라인을 가지고 있는 경우 해당 그룹에 속해있는 지연 라인의 개수를 줄이지만, 버스트 전송에 실패하고 충분한 수의 지연 라인을 가지고 있지 않은 경우에는 지연 라인의 개수를 증가시킨다. 실제로 클래스 k 에서 충돌이 발생하면, 충돌이 발생한 코어 노드는 클래스 k 의 현재 블로킹 확률 BC_c^k 를 계산하고, BC_c^k 는 클래스 k 의 목표 블로킹 확률 BC_t^k 와 비교된다. 만약 BC_c^k 가 BC_t^k 보다 크다면, 클래스 k 를 위한 목표 블로킹 확률을 넘어서는 횟수를 기록하는 OC_k 를 1 증가시킨다. 또한 OC_k 가 클래스 k 를 위한 임계값을 나타내는 OT_k 의 값을 넘어서면, 해당 클래스를 위한 지연 라인의 개수를 나타내는 FC_k 의 값과 해당 그룹의 최대 지연 라인 개수를 나타내는 FC_k^{\max} 의 값을 비교한다. 만약 FC_k 의 값이 FC_k^{\max} 의 값보다 작으면 FC_k 의 값을 1 증가시키고 best-effort 클래스의 지연 라인 개수를 나타내는 FC_n 의 값을 1 감소시킨다. 그리

고 나서 OC_k 의 값을 0으로 초기화한다. 그러나 FC_k 의 값이 FC_k^{\max} 보다 크거나 같으면, 더 이상 가용 자원이 없기 때문에 이 경우에는 데이터 버스트를 폐기한다. 한편 best-effort 클래스의 현재 블로킹 확률 BC_c^m 이 타겟 블로킹 확률 BC_c^m 보다 크다면, 이러한 상황이 발생한 코어 노드는 그림 5에서 보여주는 것처럼 충돌이 발생하는 데이터 버스트를 전송한 에지 노드로 이러한 상황을 알리기 위한 메시지를 전송한다.

에지 노드가 위의 그림 5에서 보여주는 알고리즘으로부터 조정 메시지를 받으면 그림 6에 표시된 데이터 버스트 크기 결정 알고리즘을 실행하여 데이터 버스트의 길이를 조정한다. 실제로 에지 노드는 그림 5의 알고리즘으로부터 메시지를 받을 때마다 바로 데이터 버스트의 길이를 조정하면, OBS 네트워크의 변동성이 너무 커져 전체적인 네트워크의 성능이 저하된다. 그래서 에지 노드가 메시지를 수신할 때마다 상한 수신 카운터 OI 의 값을 1 증가시킨다. 이어서 OI 의 값이 수신 횟수 상한 임계값인 OT 보다 크다면, 그림 6의 알고리즘은 3.2장에서 보여질 식들을 활용하여 최적의 데이터 버스트 크기를 결정하고 OI 의 값을 초기화한다.

즉, 트래픽 상황이 변하면 코어 노드는 데이터 버스트의 크기를 조절하고자 그림 5의 알고리즘을 이용해 피드백 메시지를 에지 노드로 전송하고, 에지 노드는 그림 6의 알고리즘을 활용해 최적의 데이터 버스트 크기를 결정한다.

반면에, 위의 두 알고리즘에 추가하여, 데이터 버스트의 크기를 증가시키기 위해 그림 7과 같은 다른 알고리즘이 필요하다. 만약 계속해서 데이터 버스트가 성공적으로 전송이 된다면, OBS 네트워크의 효율성을 높이기 위해 그림 7과 같은 알고리즘을 이용해 데이터 버스트의 크기를 늘리는 것이 좋다. 그림에서처럼 BC_c^k 가 BC_t^k 보다 작다면, BC_c^k 가 BC_t^k 보다 작은 횟수를 기록하는 UC_k 의 값을 1 증가시킨다. 또한 UC_k 의 값이 임계값인 UT_k 의 값보다 크고, FC_k 의 값이 클래스 k 의 최소 지연 라인 개수인 FC_k^{\min} 보다 크다면, FC_k 의 값을 1 감소시키고 FC_n 의 값을 1 증가시킨다. 그러나 만약 FC_k 의 값이 FC_k^{\min} 보다 작거나 같다면, 코어 노드는 에지 노드로 데이터 버스트의 크기를 증가시키기 위한 메시지를 전송한다.

그림 7의 알고리즘으로부터 조정 메시지를 받을 때 마다 에지 노드는 하한 수신 카운터 UI 의 값을 1 증가시킨다. 이어서 UI 의 값이 하한 임계값인 UT 보다 크다면, 메시지를 수신한 에지 노드는 3.2장에서 설명한 수식들을 이용해 최적의 데이터 버스트 크기를 결정하고, UI 의 값을 초기화한다.

3.2. 알고리즘 성능 분석

본 논문에서 제안된 알고리즘의 성능을 분석하기 위해 다음과 같은 몇 가지 가정을 만든다.

- IP 패킷의 도착 프로세스: λ_p [패킷/초]를 가진 포아송 프로세스 가정,
- IP 패킷의 평균 길이: $1/\mu_p$ 평균을 가지는 지수 분포(μ_p [패킷/초]는 IP 패킷의 서비스 속도),
- 코어 노드의 FDL 개수: m ,
- 클래스 개수: n (클래스 1이 가장 우선순위가 높고, 클래스 n 은 가장 우선순위가 낮은 클래스로 best-effort 트래픽을 의미),
- 서브 FDL 그룹 개수: n (서브 그룹 1은 클래스 1에, 서브 그룹 2는 클래스 2에, 그리고 서브 그룹 n 은 클래스 n 에 미리 할당),
- 두 트래픽 클래스 i 와 $j(i < j)$ 관계: i 는 j 보다 높은 우선순위 클래스를 의미,
- 파장 변환기는 전혀 고려하지 않음.

두 개의 트래픽 클래스 i 와 $j(i < j)$ 가 있을 때, 높은 클래스 i 는 서브 FDL 그룹 i 부터 n 까지 활용하여 데이터 버스트를 전송할 수 있으나, 낮은 클래스 j 는 i 에 할당된 서브 FDL 그룹 i 를 제외한 j 부터 n 까지의 서브 FDL 그룹을 활용해 전송한다. 한편 클래스 j 는 서브 FDL 그룹 1부터 $j-1$ 까지의 서브 FDL 그룹을 활용할 수 없다.

먼저 코어 노드에서 각 클래스에 대한 블로킹 확률을 분석해보자. FIFO 규칙을 준수하기 위해서 두 개의 서브 그룹 l 과 $m(l < m)$ 이 있을 때, 서브 그룹 l 의 가장 긴 지연 값은 서브 그룹 m 의 가장 짧은 지연 값보다도 작아야 한다. 먼저 $BC_c^i(i = 1, 2, \dots, n)$ 를 유도하기 위해서 각 트래픽 클래스의 블로킹 확률 $BG_i(i = 1, 2, \dots, n)$ 를 고려하자. BG_i 는 각 서브 그룹 i 에 속하는 각 지연

라인의 상태와 각 서브 그룹에 도착하는 데이터 버스트 비율에 영향을 받는다[14]. 그러므로 BG_i 는 다음과 같이 구할 수 있다.

$$BG_i = f(FG_i, \lambda_i^i), \quad (1)$$

여기에서 FG_i 는 t 개의 지연 라인들로 구성된 서브 FDL 그룹 i 를 위한 상태값 $\{f_1^i, f_2^i, \dots, f_t^i\}$ 로 이루어진다. 하나의 데이터 버스트가 서브 FDL 그룹 i 내에 있는 지연 라인 k 로 스위치 될 때, 활용 가능 상태를 f_k^i ($1 \leq k \leq t$)로 나타낸다. 그리고 λ_i^i 는 서브 그룹 i 에 도착하는 전체 데이터 버스트 도착 비율로 다음과 같이 구할 수 있다.

$$\lambda_i^i = \lambda_i + \sum_{k=1}^{i-1} \lambda_k \cdot \prod_{j=k}^{i-1} BG_j, \quad (2)$$

여기에서 첫 번째 부분은 서브 FDL 그룹 i 로 들어오는 클래스 i 의 순수 데이터 버스트의 도착 비율이고, 두 번째 부분은 i 보다 높은 1부터 $i-1$ 까지의 클래스 트래픽이 서브 그룹 i 까지 내려오는 도착 비율이다. 즉 두 번째 부분은 상위 서브 그룹에 사용 가능한 지연 라인이 없어 서브 그룹 i 까지 내려온 버스트 트래픽을 의미한다.

이제는 각 서브 FDL 그룹을 위한 블로킹 확률을 알아보자. 클래스 $n-1$ 에 속한 데이터 버스트는 서브 FDL 그룹 $n-1$ 에서 전송되지 못한 경우 하위 서브 그룹인 n 에 속한 지연 라인을 이용해 전송될 수 있기 때문에, 서브 그룹 n 과 $n-1$ 에서 사용 가능한 지연 라인을 찾지 못한 경우 블로킹된다. 또한 클래스 $n-2$ 에 속한 데이터 버스트는 서브 그룹 $n-2$ 부터 n 까지에 속한 활용 가능한 지연 라인이 없을 때 버려진다. 비슷한 방식으로 클래스 1에 속한 데이터 버스트는 모든 서브 그룹에서 사용 가능한 지연 라인이 없는 경우에만 전송될 수 없다. 그러므로 BC_c^i 는 다음처럼 계산할 수 있다.

$$BC_c^i = \prod_{j=i}^n BG_j. \quad (3)$$

BC_c^i 는 코어 노드에서 계산된다. 만약 BC_c^i 가 BC_i^i 보다 작다면, 3.2 절에서 제안한 알고리즘은 서브 FDL 그룹에 포함되어있는 지연 라인의 개수를 조절할 것이

다. 그러나 BC_c^i 가 BC_t^i 보다 큰 경우, 코어 노드는 데이터 버스트의 크기를 조절하기 위해 데이터 버스트를 전송하는 에지 노드로 크기를 조절하기 위한 요청을 전송한다.

이제는 데이터 버스트 크기를 조절해 보자. 최적의 데이터 버스트 크기를 계산하려면 코어 노드에서 전송된 트래픽 정보를 사용해 에지 노드에서 블록킹 확률을 계산해야한다. [19]처럼 블록킹 확률은 평균적인 데이터 버스트 도착 비율 λ_b [데이터 버스트/초], 평균적인 데이터 버스트 서비스 비율 μ_b [데이터 버스트/초], 지연 라인의 개수 m , 그리고 기본 지연 라인 값 D 에 의해서 영향을 받는다. 그러므로 에지 노드에서 블록킹 확률 BP_c 는 다음처럼 표현할 수 있다.

$$BP_c = f(\lambda_b, \mu_b, m, D). \quad (4)$$

식 4에서 μ_b 는 쉽게 구할 수 있다. 데이터 버스트를 전송하기 위해서, 집계된 IP 패킷들의 전체 크기는 데이터 버스트를 위한 임계값보다 커야만 한다. 그러므로 μ_b 는 다음처럼 구할 수 있다.

$$\mu_b = \frac{\mu_p \cdot W_c}{S_b \cdot \mu_p + W_c} \quad (5)$$

여기에서 S_b [비트]는 데이터 버스트의 임계값이고, W_c [비트/초]는 각 채널의 대역폭이고, 그리고 $1/\mu_p$ [초]는 지수 분포를 가지는 평균적인 IP 패킷의 크기이다. 또한 정상 상태에 있기 위해서는, 입력 채널의 속도는 출력 채널의 속도와 동일해야 하므로 $\lambda_p/\mu_p = \lambda_b/\mu_b$ 처럼 구할 수 있다. 그러므로 λ_b 는 다음처럼 구할 수 있다.

$$\lambda_b = \frac{\lambda_p \cdot W_c}{S_b \cdot \mu_p + W_c} \quad (6)$$

λ_b 와 μ_b 같은 트래픽 조건, 지연 라인 개수 m , 그리고 기본 지연 라인 D 가 주어지면, 최적의 블록킹 확률은 식 4를 이용해서 구할 수 있다. 그러나, 코어 노드는 에지 노드와는 다른 D 값을 가지고 있고, 에지 노드는 코어 노드의 D 값을 알 수 없다. 따라서 트래픽 조건이 변경되면 다른 D 값으로 구성된 새로운 FDL 버퍼를 사용해야 하지만 FDL은 하드웨어 구성요소이기 때문에 쉽게 대체할 수 없다. 그러므로, 트래픽 조건이 변하면, λ_b 와

μ_b 와 같은 트래픽 조건은 변한 트래픽 조건하에서 계산된 새로운 S_b 에 영향을 받기 때문에 최적의 임계값 S_b 는 다시 계산되어야만 한다.

식 5로부터 평균적인 데이터 버스트의 크기는 다음처럼 구할 수 있다.

$$E[L_b] = \frac{1}{\mu_b} = \frac{S_b \cdot \mu_p + W_c}{\mu_p \cdot W_c} = \frac{S_b}{W_c} + \frac{1}{\mu_p} = L, \quad (7)$$

여기에서 $E[X]$ 는 랜덤 변수 X 에 대한 기댓값이고 L_b [초]는 데이터 버스트의 크기를 나타내는 랜덤 변수이다. IP 패킷을 데이터 버스트로 수집하는 과정은 임계값을 이용하기 때문에 $E[L_b]$ 는 다음처럼 두 부분으로 구성되어 있다. 첫 번째는 고정된 부분으로 식 7에서 첫 번째 부분이고, 두 번째는 식 7의 두 번째 부분으로 가변적인 부분이다.

일반적으로, 코어 노드를 설계할 때, 특정한 트래픽 조건을 이용해 구한 D 값을 이용해 코어 노드를 위한 지연 라인들을 구성한다. 또한 데이터 버스트 길이에 관한 임계값도 이러한 특정한 조건을 이용해서 결정한다. 시스템을 설계할 때 사용하는 기본 지연 라인 값, 데이터 버스트 임계값, 그리고 평균적인 데이터 버스트 길이를 각각 D_0 , S_0 , 그리고 L_0 라고 할 때, 일반적으로 D 값은 L 값에 비례해서 결정되기 때문에 트래픽 조건이 변하면 최적의 성능을 보장하기 위해서는 D 값이 변해야 한다. 그러므로 이러한 관계는 비례식 $D_0 : L_0 = D_n : L_n$ 으로 표현할 수 있고, 여기에서 D_n 과 L_n 은 새로운 기본 지연 라인 값과 이에 맞추어 수정된 평균적인 데이터 버스트 크기이다. 이전에 구한 식들을 활용하면, 새로운 데이터 버스트의 임계값 S_b^n 은 다음처럼 구할 수 있다.

$$S_b^n = W_c(L_0 \cdot \frac{D_n}{D_0} - \mu_p^{-1}). \quad (8)$$

3.3. 동작 조건 분석

이 절에서는 이전에 구한 식들을 이용해 제안한 알고리즘의 최적 동작을 조건을 분석하고자 한다. 식 3으로부터 클래스 k 의 블록킹 확률을 구해보면 다음과 같다.

$$BC_C^k = \prod_{i=k}^n BG_i. \quad (9)$$

위의 식을 정리하면 클래스 j 를 위한 블록킹 확률은 다음과 같이 구할 수 있다.

$$BC_c^j = BG_j \prod_{i=j+1}^n BG_i. \quad (10)$$

식 10을 이용해서 서버 FDL 그룹 j 를 최소 블록킹 확률을 구해보자. FC_j 는 3.2 절에서 제안한 알고리즘을 이용해 구할 수 있다. 또한 FC_j 가 적을수록 높은 블록킹 확률이 나오고, 반대로 FC_j 가 많을수록 낮은 블록킹 확률을 구할 수 있다. 이 사실과 식 10으로부터 클래스 j 가 서버 그룹 j 를 이용해 목표 블록킹 확률을 만족하는 최소 FC_j 를 구할 수 있다. 그러므로 이렇게 구한 FC_j 를 이용한 최소 블록킹 확률은 다음 식을 이용해 구할 수 있다.

$$BG_j \leq \frac{BC_c^j}{BC_c^{j+1}}. \quad (11)$$

실제로 클래스 j 에 속한 데이터 버스트를 전송하기 위해서는 서버 그룹 j 에 속한 지연 라인이 최소한 하나는 있어야 한다. 그러므로 서버 그룹 j 에 속한 최대 지연 라인 개수는 다음 식처럼 구할 수 있다.

$$FC_j = \sum_{k=1(k \neq j)}^n FC_k^{\min}. \quad (12)$$

IV. 성능 분석

이 장에서는 OPNET(현재 리버베드 모델러)을 이용한 시뮬레이션과 수치 분석을 통한 성능 분석이 묘사된다. 이러한 분석을 하기 위해서 다음과 같이 몇 가지 매개변수를 가정한다.

- 채널 대역폭 W_c : 1 Gbps,
- 초기 데이터 버스트 임계값: 200 kbps,
- IP 패킷 도착 프로세스: λ_p 의 포아송 프로세스,
- 평균 IP 패킷 길이: $1/\mu_p$ 의 지수 분포,
- 데이터 버스트 생성: 임계값 기반 정책 사용.

3.2 절에서 제안된 데이터 버스트의 길이를 동적으로

제어하는 알고리즘의 유효성을 보여주기 위해서 세 개의 클래스를 고려한다. 클래스 1은 가장 높은 우선순위 클래스이고, 클래스 3은 가장 낮은 우선순위 클래스이다. 또한 세 개의 클래스를 고려하기 때문에, FDL 버퍼도 세 개의 서버 그룹으로 고려한다. 서버 그룹 1은 클래스 1에, 서버 그룹 2는 클래스 2에, 그리고 서버 그룹 3은 클래스 3에 할당된다. 그리고 서버 그룹 1은 클래스 1에 의해서, 서버 그룹 2는 클래스 1과 2에 의해서, 그리고 서버 그룹 3은 모든 클래스에 의해서 활용된다. [20]에서 보았던 것처럼 자원의 효율적인 활용과 안정적인 서비스를 달성하기 위해서는 높은 우선순위의 트래픽이 너무 많아서는 곤란하다. 이에 따라 클래스 1, 2, 그리고 3의 분포 비율은 각각 10%, 20%, 그리고 70%로 가정하며, 클래스 1과 2의 목표 블록킹 확률을 각각 10^{-4} 과 10^{-3} 으로 가정한다.

그림 8은 D 값에 따른 블록킹 확률의 변화를 보여준다. 그림에서 D 가 증가함에 따라 블록킹 확률이 감소하지만, D 가 일정 지점을 통과하면 확률이 증가함을 알 수 있다. 예를 들어 그림은 제공 부하가 0.6일 경우 D 값이 약 0.3일 때까지 블록킹 확률이 감소하지만, 이후 D 값의 증가에 따라 블록킹 확률이 증가함을 보여주고 있다. 또한 그림에서 점선은 제공 부하와 같은 트래픽 조건이 변함에 따라 서로 다른 D 값에서 최적의 블록킹 확률을 나타내고 있음을 보여주고 있다. 즉 이 말은 트래픽 조건이 변경되면 이에 따라 D 값이 변해야만 한다는 것을 의미한다. 다시 말해 새로운 FDL 버퍼를 고려해야 한다는 것을 의미하나, FDL 버퍼는 하드웨어 장비이므로 대체할 수가 없어, 동적인 데이터 버스트 길이를 통한 최적의 블록킹 확률을 구할 수 있고 이를 이용해 클래스별 서비스 차등화를 달성하고자 한다.

이제는 새로운 알고리즘이 이전 알고리즘에 비해 개선되었음을 보이기 위해 이전에 제안된 알고리즘과 비교해보자. 그림 9는 두 알고리즘을 비교한 결과이다. 그림 9(a)에서 제공 부하가 높을 때 70%로 대다수를 차지하는 클래스 3 best-effort 트래픽의 경우 거의 전송이 이루어지지 않음을 알 수 있다. 실제로 제공 부하가 0.7을 넘어서면 10% 이상의 트래픽이 블록킹되고 있음을 알 수 있다. 또한 클래스 2의 경우 제공 부하가 0.8을 넘어 서면서부터 목표 블록킹 확률보다 높은 확률값을 보여주고 있다.

한편 그림 9(b)는 본 논문에서 제안된 알고리즘을 이

용한 결과이다. 이 그림에서 보여지는 것처럼, 동적으로 데이터 버스트의 크기를 제어함으로써 그림 9(a)에서 보여주는 결과와는 상당히 다른 결과를 보여주고 있다. 이전 논문에서 문제가 되었던 best-effort 트래픽조차도 상당량의 트래픽이 전송되고 있음을 보여주고 있다. 또한 클래스 2조차도 제공 로드 0.8 이상에서 목표 블로킹 확률을 달성하고 있음을 보여주고 있다.

실제로 새로운 알고리즘의 가장 큰 장점을 그림 10에서 보여주고 있다. 그림 10은 각 클래스의 목표 블로킹 확률을 지키면서 트래픽을 전송하기 위해 각 서브 그룹이 필요로 하는 지연 라인 개수의 변화를 보여주는 그림이다. 새로운 알고리즘의 경우 이전 알고리즘과 비교해 변동폭이 상당히 작음을 알 수 있다. 이것은 전반적인 네트워크 안정성의 증가를 의미한다.

한편 표 1은 64개의 지연 라인으로 구성된 FDL 버퍼에서 두 알고리즘의 통계치를 비교한 것이다. 표는 새로운 알고리즘의 유효성을 다시 보여주고 있다. 그룹 1과 2 모두 표준편차의 값이 상당히 작아진 것을 알 수 있다.

그림 11은 새로운 알고리즘의 또 다른 장점을 보여준다. 그림에서 보여지는 것처럼 클래스 1과 2의 경우에는 두 알고리즘이 큰 차이가 없으나, 클래스 3 best-effort 트래픽의 경우 전송되는 트래픽 양에 있어서 많은 차이가 있다. 실제로 best-effort 트래픽은 70%정도를 차지하기 때문에, 비록 우선순위는 낮더라도 안정적인 서비스를 유지하기 위해서는 best-effort 트래픽에 대한 고려가 필요하다. 표 1을 보면 이전 알고리즘의 경우 클래스 1과 2의 트래픽을 전송하기 위해서 64개의 지연 라인 중에서 49개의 지연 라인을 사용하는 데 반해서, 본 논문에서 제안한 알고리즘의 경우 35개만을 사용하는 것을 알 수 있다. 그러므로 표 1과 그림 11로부터, 본 논문에서 언급한 새로운 알고리즘이 best-effort 트래픽에 더 많은 지연 라인을 제공해주고 있고, 이로 인해 새로운 알고리즘이 더 많은 데이터 버스트를 전송할 수 있는 것이다.

그림 12는 제공 부하에 따른 임계값의 변화를 보여준다. 그림 8로부터 제공 부하의 변화에 따라 D 의 값을 변경해야 함을 알 수 있었다. 특히 그림 5와 식 8은 제공 부하의 증가에 따라 D 와 S_i 가 감소되어야 함을 보여주고 있다. 그림 12는 이러한 결과를 보여주고 있다.

한편 그림 13은 64개의 FDL과 트래픽 분포 비율이 클래스 1은 10%, 클래스 2는 20%, 그리고 클래스 3은 70%일 때, 이전 알고리즘과 새로운 알고리즘의 블로킹

확률을 비교한 그림이다. 그림에서 보면 클래스 1과 2는 큰 차이 없이 잘 서비스가 이루어지고 있으나, 클래스 3의 경우 상당한 차이가 있음을 알 수 있다. 실제로 대부분의 트래픽은 best-effort 트래픽이기 때문에, 클래스 3의 전송률은 전반적인 네트워크의 안정성과 관련이 있다. 새로운 알고리즘이 적은 수의 FDL에 불구하고 더 원활한 서비스가 이루어짐을 알 수 있다.

V. 결론

본 논문은 서비스 차등화를 달성하기 위해 많은 수의 광 지연 라인을 요구하고 그림에도 불구하고 대다수를 차지하는 가장 낮은 등급인 best-effort 트래픽 클래스의 희생을 요구하는 기존 알고리즘의 문제점을 극복하고자 새로운 차등화 알고리즘을 제안했다. 본 논문의 알고리즘은 데이터 버스트의 크기를 동적으로 제어함으로써 상위 클래스에는 원하는 서비스 품질을 제공하면서도 가장 낮은 등급의 클래스에도 상당한 서비스 품질을 제공함을 보여주었다. 또한 각 그룹이 사용하는 지연 라인의 변동폭을 줄임으로써 전반적인 네트워크 안정성을 높일 수 있었고 이를 통해 광 인터넷의 적용성을 높일 수 있었다.

한편 본 논문에서는 서비스 차등화를 달성하기 위해 피드 포워드 방식의 버퍼 구조를 활용했으나, 시스템 활용의 다양성을 고려하기 위해 피드백 방식의 버퍼 구조를 활용한 방식에 대한 연구도 필요하다.

REFERENCES

- [1] V. K. A. Kumar, K. S. Reddy, and M. N. G. Prasad, "Review of Contemporary Literature on Burst Assembling and Routing Strategies in OBS Networks," *Journal of Optics*, vol. 47, pp. 324-331, Mar. 2018.
- [2] E. N. Lallas, "A Survey on All Optical Label Swapping Techniques: Comparison and Trends," *Optical Switching and Networking*, vol. 31, pp. 22-38, Jan. 2019.
- [3] M. Gomba and B. Nleya, "Enhanced Congestion Management in OBS Networks," in *Proceeding of 2017 IEEE AFRICON*, Cape Town: South Africa, pp. 320-324, 2017.

- [4] B. Nleya, P. Khumalo, and A. Mutsvangwa, "A Restricted Intermediate Node Buffering-Based Contention Control Scheme for OBS Networks," in *Proceeding of 2019 icABCD*, Winterton: South Africa, pp. 1-6, 2019.
- [5] L. -S. Cui and G. Srivastava, "QoS Routing Algorithm for OBS Networks Based on a Multi-Objective Genetic Algorithm," *IEEE Access*, vol. 10, pp. 12047-12056, Dec. 2021.
- [6] V. K. A. Kumar, K. S. Reddy, and M. G. Prasad, "Optical Burst Routing by Balanced Wavelength Allocation under Multi-Objective Quality Metrics," *Wireless Personal Communications*, vol. 107, pp. 1093-1114, Jul. 2019.
- [7] B. Nleya, "A RWA Framework for Improved Throughput in OBS Networks," *Journal of Optics*, vol. 51, no. 1, pp. 89-105, Aug. 2021.
- [8] M. Kaidan, V. Andrushchak, N. Kryvinska, M. Klymash, and M. Seliuchenko, "Configuration of Network Management for Energy Efficiency in Optical Transport Networks Using GMPLS and OBS Techniques," *Simulation Modelling Practice and Theory*, vol. 74, pp. 17-27, May. 2017.
- [9] P. Bhattacharya, A. Singh, A. K. Tiwari, V. K. Pathak, and R. Srivastava, "DbOBS: Dual Buffered Switch for Variable Optical Bursts in Future Datacenters," *Optical and Quantum Electronics*, vol. 53, pp. 235, Apr. 2021.
- [10] P. Khumalo, B. Nleya, and A. Mutsvangwa, "A Controllable Deflection Routing and Wavelength Assignment Algorithm in OBS Networks," *Journal of Optics*, vol. 48, no. 4, pp. 539-548, Nov. 2019.
- [11] H. Sibanda, and B. Nleya, "Model Analysis of Dynamic Priority Segmented Burstification and Deflection Routing," in *Proceeding of 2017 IEEE AFRICON*, Cape Town: South Africa, pp. 325-331, 2017.
- [12] D. H. Hailu, G. G. Lema, E. A. Yekun, and S. H. Kebede, "Unified Study of Quality of Service (QoS) in OPS/OBS Networks," *Optical Fiber Technology*, vol. 36, pp. 394-402, Jul. 2017.
- [13] R. S. Barpanda, A. K. Turuk, and B. Sahoo, "QoS Aware Routing and Wavelength Allocation in Optical Burst Switching Networks Using Differential Evolution Optimization," *Digital Communications and Networks*, vol. 4, no. 1, pp. 3-12, Feb. 2018.
- [14] Y. Lee, N. Kim, and M. Kang, "Service Differentiation Scheme in OBS Networks," *Photonic Network Communications*, vol. 15, no. 1, pp. 67-75, Feb. 2008.
- [15] I. Chlamtac, A. Fumagalli, and S. Chang-Jin, "Multibuffer Delay Line Architectures for Efficient Contention Resolution in Optical Switching Nodes," *IEEE Transactions on Communications*, vol. 48, no. 12, pp. 2089-2098, Dec. 2000.
- [16] T. Zhang, K. Lu, and J. R. Jue, "Shared Fiber Delay Line Buffers in Asynchronous Optical Packet Switches," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 4, pp. 118-127, Apr. 2006.
- [17] L. Tancevski, S. Yegnanarayanan, G. Castanon, L. Tamil, F. Masetti, T. McDermott, "Optical Routing of Asynchronous, Variable Length Packets," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 2084-2093, Oct. 2000.
- [18] L. Tancevski, G. Castanon, F. Callega, and L. Tamil, "Performance of an Optical IP Router Using Non-degenerate Buffers," in *Proceeding of IEEE Globecom 1999*, Rio de Janeiro: Brazil, vol. 2, pp. 1454-1459, 1999.
- [19] T. Zhang, K. Lu, J. P. Jue, "An Analytical Model for Shared Fiber-delay Line Buffers in Asynchronous Optical Packet and Burst Switches," in *Proceeding of ICC 2005*, Seoul: Korea, vol. 3, pp. 1636-1640, 2005.
- [20] N. Kim, H. Yun, Y. Lee, and M. Kang, "A QoS Based Service Reservation: Supporting Multiple Service in Optical Access Networks," *Lecture Notes in Computer Science*, vol. 3262, pp. 81-90, Sep. 2004.



이용규(Yonggyu Lee)

전기전자 공학박사

※관심분야: 광 인터넷, 서비스 품질 보장형 네트워크, 초고속 인터넷