XGPON의 여유 대역폭을 활용한 동적대역할당 알고리즘

정종훈 · 한만수*

목포대학교 광공학연구소

Dynamic Bandwidth Allocation Algorithm Utilizing Unused Bandwidth for XGPON

Jong Hoon Jeong · Man Soo Han*

Mokpo National University
*교신저자, E-mail : mshan@mokpo.ac.kr

요 약

본 논문에서는 XGPON (10-Gigabit-capable passive optical network)의 새로운 동적대역할당 알고리즘을 제안한다. 기존의 알고리즘들과 달리 제안된 알고리즘은 각 서비스 클래스별로 공통의 서비스 카운터를 갖는다. 그리고 하나의 서비스 클래스에서 사용되지 않고 남는 대역폭을 다른 서비스클래스에서 사용할 수 있게 하여 대역폭 활용도를 향상시킨다. 제안된 알고리즘이 기존 알고리즘들에 비해 비교우위에 있는 것을 보이기 위해 다양한 예를 제시한다.

ABSTRACT

This paper proposes a new dynamic bandwidth allocation algorithm for XGPON (10-Gigabit-capable passive optical network). The proposed algorithm uses a common service counter for each service class unlike to the existing algorithms. The unused bandwidth of a service class can be used for other service classes to increase bandwidth utilization. We illustrate examples to show the proposed algorithm is superior to existing algorithms.

키워드

XGPON, dynamic bandwidth allocation, bandwidth utilization

1. 서 론

가입자망에서 트래픽 증가를 해결하기 위해 저비용이면서 고속 대역을 제공할 수 있는 XGPON (10-Gigabit-capable passive optical network) 기술이 많이 주목을 받고 있다. XGPON은 1개의 OLT (optical line termination)과 다수의 ONU는 다수의 queue를 서비스 클래스별로 1개씩 가지고 있다. XGPON 기술에서 서비스 클래스는 T-CONT (transmission container)로 불린다. 각 ONU의 각 queue는 대기중인 패킷들의 크기를 OLT에게 request로 보내며 OLT는 동적대역할당을 수행하여 각각의 queue에게 전송시간과 전송

량을 결정하여 통보한다. XGPON시스템의 모든 동작은 단위시간인 125us에 동기되어 있다. 따라서 동적대역할당도 매 단위시간마다 이루어지며 동적대역할당시에 할당하는 대역폭도 125us로 제한된다 [1][2][3].

지금까지의 대부분의 동적대역할당 알고리즘들은 동적대역할당을 위해 각 queue별로 서비스랑을 저장하는 서비스 카운터와 그 서비스 카운터가 재충전되는 시간을 나타내는 감소 타이머를 사용하였다. 감소타이머는 매 단위시간마다 1씩 감소하여 타이머 값이 0이 되면 서비스 카운터가 재충전되고 감소 타이머 값도 재충전된다. 그러나각 queue가 개별적으로 서비스 카운터를 갖는 경우에는 request가 없는 queue에 할당된 서비스

양이 낭비되는 문제가 발생 한다. 이러한 문제를 해결하는 방법으로 각 queue별로 개별 서비스 카 운터를 두지 않고 T-CONT 별로 공용 서비스 카 운터를 사용하는 방법이 제시되었다 [4].

그런데 [4]의 방법에서도 어떤 T-CONT의 request가 적은 경우 이 T-CONT의 서비스 량이 낭비가 될 수 있다. 본 논문에서는 T-CONT별로 공용 서비스 카운터를 사용하는 경우에 서비스 량이 낭비되는 예를 제시하고 이를 해결하기 위한 새로운 동적대역할당 알고리즘을 제안한다.

Ⅱ. 본 론

여기서는 [4]의 방법에서 서비스 카운터의 서비스량이 낭비되는 문제점을 먼저 설명한다. [4]에서는 ONU에 3개의 queue가 T-CONT type 2, 3, 4별로 1개씩 있다고 가정했다. T-CONT type 1은 정적대역할당을 하므로 동적대역할당문제에서는 고려하지 않는다. 동적대역할당을 위해 OLT는 각 T-CONT별로 서비스 카운터, S2, S3, 그리고 S4를 갖는다. 또 감소 카운터도 각 T-CONT별로 T2, T3, T4를 갖는다. 각각의 서비스 카운터는 재충전 값으로 A1, A2, 그리고 A3를 갖는다. 각각의 감소 카운터도 재충전 값으로 V1, V2, 그리고 V3를 갖는다. 감소 카운터 Ti는 매 단위시간마다 1씩감소하고 Ti = 0이 되면 Ti = Vi로 재충전된다. 또 Ti = 0이 되면 Si=Ai로 재충전된다.

그림 1은 서비스 카운터의 서비스량이 낭비되는 예이다.

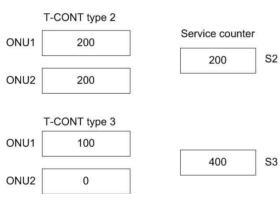


그림 1. 서비스량 낭비의 예

그림 1에서 S2 = 200, S3 = 400이다. 그리고 T-CONT type 2에 대해서 ONU1, 2의 전송요청이 각각 200 byte, 200 byte이다. T-CONT type 2에서 200 byte가 전송허가가 되고 S2 = 0으로 감소된다. T-CONT type 3에서는 ONU1만 100 byte의 전송요청이 있다. T-CONT type 3에서는 100 byte가 전송허가가 되고 S3 = 300으로 감소한다. T-CONT type 2에서는 전송허가가 되지 않은 전송요청이 200 byte가 남아 있고 S3의 서비

스 잔량이 300 byte이므로 S3의 200 byte가 낭비가 되고 있음을 알 수 있다.

위 예에서 우리가 T-CONT type 2에서 S3의 서비스 잔량을 사용할 수 있다면 서비스 낭비를 줄일 수 있음을 알 수 있다. 본 논문에서는 이를 해결하는 동적 대역할당 알고리즘을 제안한다. 서 비스 잔량의 낭비를 줄이기 위한 가장 간단한 방 법은 동적대역할당이 끝난 후 Si의 남은 서비스 량을 가지고 다시 동적대역할당을 반복하는 방법 이다. 두 번째 동적대역할당에서 사용할 서비스 카운터를 T-CONT type별로 U2, U3 그리고 U4 라고 하자. 동적대역할당을 먼저 수행한 후 Si의 서비스 잔량을 모두 합한 후에 이를 U2에 충전한 다. 즉, U2 = S2+S3+S4가 된다. S2 대신에 U2를 사용하여 T-CONT type 2에 대해서 동적대역할당 을 수행한다. 이때 전송허가만큼 U2를 감소시킨 다. U2의 잔량을 갖고 즉, U3 = U2로 하여 T-CONT type 3에 대해서 동적대역할당을 수행한 다. 비슷하게 T-CONT type 4에 대해서도 동적 대역할당을 수행한다.

위 예의 경우 두 번째 동적대역할당에서 U2 = 200이 되어 T-CONT type 2의 남은 전송요청인 200 byte에 대해 전송허가가 됨을 알 수 있다.

Ⅲ. 결 론

본 논문에서는 T-CONT별로 공용 서비스 카운터를 사용하는 경우에 서비스 량이 낭비되는 예를 제시하고 이를 해결하기 위한 새로운 동적대역할당 알고리즘을 제안하였다. 제안된 알고리즘에서는 동적 대역할당을 두 번 수행하고 두 번째수행에서는 첫 번째 수행에서 남는 서비스량을 사용하여 동적대역할당을 수행한다.

참고문헌

- [1] ITU-T Rec. G.987.1, "10 Gigabit-capable passive optical network (XG-PON): General requirements," 2010.
- [2] ITU-T Rec. G.984.3, "Gigabit-capable passive optical networks (G-PON): Transmission convergence layer specification," 2008.
- [3] ITU-T Rec. G.987.3 Rev.2, "10-Gigabit-capable passive optical networks (XG-PON): Transmission convergence (TC) specifications," 2010.
- [4] Man Soo Han, "Simple and feasible dynamic bandwidth allocation for XGPON," $15^{\rm th}$ ICACTM, pp. 341-344, Jan. 2013 .