

멀티미디어 QoS 보장을 위한 멀티 큐에서의 대역폭 조절 방법

김낙현^o 류현기

^o경북대학교 컴퓨터 공학과, 대구경북과학기술연구원
rackhyunkim^o@yahoo.co.kr, hkryu@dgist.ac.kr

The Multi-queue Bandwidth Regulation Method for Multimedia QoS

Rackhyun Kim^o Hyunki Ryu

^oDept. of Computer Engineering Kyungpook National University
DaeGu Gyeongbuk Institute of Science and Technology

요 약

최근 인터넷에서는 인터넷 방송, 화상 회의, VoIP 등 QoS 보장을 요구하는 새로운 응용 서비스들의 출현과 함께 IP QoS의 문제는 인터넷에서 가장 중요한 과제 중 하나로 등장하고 있다. 이러한 QoS의 적용은 가장 상용화 되어 있는 이더넷 망에서도 마찬가지로 적용되어야 하는 문제이다. 또한 사용자들은 SLA(Service Level Agreement)내에서 최선의 QoS 요구하게 되었다. QoS 보장을 위해 기존에는 통합 서비스(Integrated Service)나 차별화 서비스(Differentiated Service) 방식만을 사용하였으나 실제 적용하는데 구현상의 여러 문제점이 발견되었고, 대역폭 할당 방식에 있어서도 토큰 버킷(Token Bucket)과 같은 복잡한 방식을 사용하였다. 본 논문에서는 간단하면서도 적용하기 쉬운 QoS 보장을 위한 멀티큐에서의 속도 계산(Rate Calculation)방식을 제안하며 OPNET을 통해 제안된 방식을 검증하도록 한다.

1. 서 론

최근 인터넷의 사용 인구가 기하급수적으로 늘어나면서 네트워크 기반 응용 프로그램의 발전과 함께 인터넷 트래픽의 폭발적인 증가를 가져 왔다. 인터넷 트래픽의 증가 원인은 단순히 인터넷 사용자의 증가라는 측면만을 생각할 수도 있으나, 더 나아가 다양한 인터넷 어플리케이션의 개발 및 다량의 데이터 전송을 요하는 어플리케이션의 출현 등에서 그 원인을 찾아볼 수 있다. 이러한 트래픽의 증가를 막기 위해서는 단순히 망을 확충하는 빅 파이프(Big Pipe) 방법 및 다양한 유형의 트래픽들을 그 특성에 맞게 처리함으로써 서비스 모델을 통해 각 트래픽 특성을 보장해 줄 수 있는 기법을 적용하는 방법이 있다. 일반적으로 어플리케이션의 특성에 따라 트래픽을 차별화 해서 처리함으로써 차별화된 서비스 특성을 제공하는 것을 '인터넷 상에서 QoS (Quality of Service)[1]를 제공한다.' 라고 칭한다.

이러한 QoS를 보장하는 연구는 이미 다양한 형태로 계속 진행되어 왔었다. Diffserv[2]과 Intserv[3] 그리고 MPLS[4] 등과 같은 네트워크 구조들이 제안되었으며, 프로토콜에서는 RSVP[5], RTP[6] 등이 좋은 예이다. 또한, 이 외에도 여러 분야에서 QoS에 관한 연구가 활발하게 진행되고 있다. 이러한 연구들은 기본적으로 인터넷에 적용되어 왔다. 그렇지만 각 네트워크에 따라 그 특성에 맞게 개량되어 사용 가능하다. 본 논문에서는 인터넷에 이러한 방식을 적용시켜 활용하고자 한다. 또한 사용자와 서비스 공급업체와의 SLA에 따른 대역폭 조절을 위한 방안으로 속도 계산 방식을 사용하고자 하며 기존에 사용되던 큐 스케줄링 방식인 선입선출(FIFO Queuing), 공정 순위(Fair Queuing), 가중치 공정 순위(Weighted Fair Queuing) 등이 보장하지 못하는 인터넷에 맞는 멀티미디어 QoS를 보장하기 위해 DWRR 방식을 응용 하여 사용 한다.

본 논문에서는 EFM 구간에 활용되는 이더넷 장비에 쉽게 적용이 가능하며, 또한 멀티미디어 데이터의 QoS를 보장할 수 있으며, 토큰 버킷 방식의 유용성과 리키 버킷 방식의 단순성을 지닌 멀티 큐에서의 대역폭 조절 방법을 제안 한다. 논문의 구성은 2장에서 기존의 큐 스케줄링과 본 논문에서 제안하고 있는 멀티 큐에서의 대역폭 조절 방법을 그리고 3장에서는 OPNET에서 본 논문에서 제안하는 방식을 시뮬레이션을 통해 검증하고 결과를 도출한다. 그리고 4장에서 결론을 내린다.

2. 본 론

2.1. QoS 보장 기법

일반적인 QoS보장을 위한 기법은 순차적으로 진행이 되었다. 그림 1은 전반적인 QoS 보장 기법들을 보여 주고 있다. 분류기(Classifier)는 입력된 패킷을 어떤 기준에 따라 여러 개의 클래스로 구분하는 기능을 한다. 이것은 패킷 분류의 목적이 동일하거나 유사한 특성을 갖는 패킷들을 함께 처리함으로써 QoS 구조를 단순하게 하자는 것에 있다.

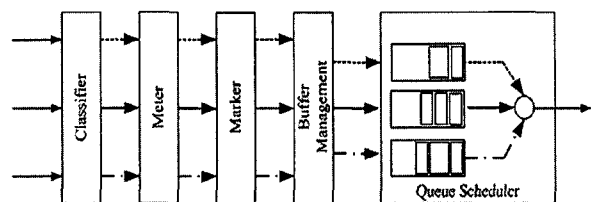


그림 1. 순차적 QoS 보장 기법

그리고 패킷 미터와 마커는 트랙픽을 측정하고 그 결과에 따라 순응 여부를 패킷에 표시하는 기능만을 포함하게 된다. 이렇게 진행된 패킷은 큐 관리 기법에 의해 패킷이 큐에 들어가는 데 있어서 어떤 방식으로 혼잡을 방지 하면서 큐에 패킷들을 넣어 주느냐를 결정 한다. 그 중 가장 대표적인 방식으로 RED와 BLUE, GREEN을 들 수 있다. 이렇게 유입된 패킷들은 멀티 큐에서 각 데이터의 특성이나 가중치들에 의해 큐에서 빠져 나오게 되는데 이를 큐 스케줄링이라 한다.

2.2. 멀티 큐에서의 대역폭 조절 방법

2.2.1 패킷 분류 방안

본 논문에서는 멀티미디어 데이터에서의 패킷 분류를 위하여 음성(Voice), 영상(Video), 웹(Web) 데이터로써 세 분류로 나누도록 하겠다. 실제로 세 분류를 넣어 서더라도 유사 패킷끼리 묶어 여러 개의 큐를 동일 정책으로 묶어서 사용하면 성능에 큰 문제는 없게 된다. 음성 데이터는 오디오 코덱 권고안[22]에서 전송 계층에 주기적으로 전송될 경우 전송 패킷 간격은 최대 5ms를 초과해서는 안 된다. 또한 최대 속도를 8Kbit/s를 유지하여야 한다. 이것은 음성의 지연을 최대한 억제함으로써 정확한 전송 및 지연에 따른 에러를 방지하기 위해서이다. 반면에 영상 데이터의 경우 프로토콜 자체는 UDP를 사용하고 어느 정도의 지연나 딜레이를 허용하지만 대역폭 자체를 보장해 주어야만 올바른 영상을 재생해 낼 수 있다. 세 번째 큐에서 적용될 일반 웹 데이터의 경우 상위 큐의 보장 서비스에 의해 많은 지연이 발생되거나 고착(Starvation)이 발생할 우려가 생긴다. 이러한 지연을 최대한 줄임으로써 대역폭을 보장해 준다.

2.2.2 큐 스케줄링과 대역폭 조절

패킷의 출력을 관리하기 위한 방안으로 기존에는 우선순위를 두거나 큐에 따른 순서 등 여러 가지 방안으로 패킷을 내 보내는 방식들이 있었다. 본 논문에서는 각 큐에서 보장하는 특성들에 맞추어 출력 순서를 정하는 방식으로 적용시키고자 한다. 음성 데이터 h.711의 경우 최대 보장 속도를 8 Kbit/s 를 유지하여야 하며 패킷 간격을 5ms 이내로 제한하고 있다. 또한 영상 데이터의 경우 많은 트랙픽을 발생시키면서도 최대 대역폭을 보장해 주어야 한다. 웹 데이터의 경우는 대역폭과 패킷 간격에 민감하진 않지만 특정 지역에서 고착되는 것을 방지해 주어야 한다. 또한 이더넷에서는 패킷의 크기가 고정되어 있지 않기 때문에 패킷의 개수에 의한 출력을 고려하게 되면 대역폭을 조절하기 힘들어 지게 되는 점까지 고려해 주어야 한다. 그림 2는 본 논문에서 제안하는 데이터 특성을 고려한 스케줄링 방식을 보여준다.

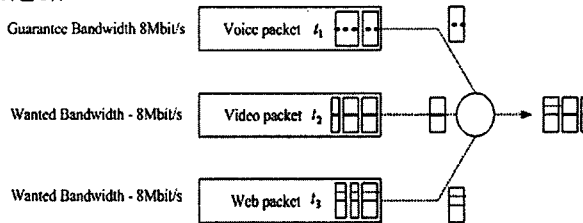
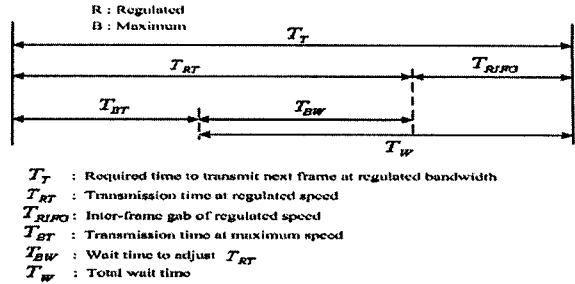


그림 2. 데이터 특성을 고려한 스케줄링 방식

대역폭 조절 방안으로는 음성 패킷 및 영상과 웹 패킷에 대해 대역폭을 조절하게 된다. 그림 2에서 보듯이 음성은 8Mbit/s를 보장한 후 영상과 웹에 대해 할당해 준다. 그림 3은 대역폭 조절을 위한 속도 계산 방식의 시간 흐름도를 나타낸다.



T_T : Required time to transmit next frame at regulated bandwidth
 T_{RT} : Transmission time at regulated speed
 $T_{RT/BC}$: Inter-frame gap of regulated speed
 T_{BT} : Transmission time at maximum speed
 T_{BW} : Wait time to adjust T_{RT}
 T_W : Total wait time

그림 3. 속도 계산 방식의 시간 흐름도

속도 계산 방식은 기다리다 보내는(Send And Wait) 방식으로 다음과 같은 순서로 대역폭의 제어를 수행한다. 먼저 하나의 이더넷 프레임의 전송하는 중간에 해당프레임이 제한된 대역폭 상에서 전송될 때 걸리는 시간을 계산한다. 그리고 해당 이더넷 프레임의 전송이 끝난 후에 실제 전송된 시간과 계산된 전송 시간 만큼 다음 프레임의 전송을 막음으로써 제한된 대역폭으로 전송된 것과 같은 효과가 나타나게 한다. 또한 패딩 존(padding zone)이 부가된 프레임 버퍼의 관리 방법을 사용하여 가변 크기의 이더넷 프레임의 효율적으로 관리할 수 있도록 한다.

그림 3에서 보듯 비트 사이저가 α 인 이더넷 프레임이 R Mbps의 속도로 전송될 때 필요한 시간은 전송시간(Transmission time) + 인터 프레임 갭(Inter frame gap)의 시간을 필요로 한다. 그러나 실제 전송은 B Mbps의 속도로 일어나므로 실제 패킷의 전송은 T_T 에 종료하게 된다. 그러므로 전송은 B Mbps로 했더라도 R Mbps의 속도로 전송한 것과 같은 효과를 나타내기 위해서는 패킷의 전송 후에 $T_W + T_{IFG}$ 만큼 대기하여야 한다. 이러한 시간 값들은 B MHz의 PHY Chip 기준으로 보면 그림 4를 참고 하여 아래와 같은 값을 얻을 수 있다.

Byte						
Preamble	SFD	DA	SA	Type	Data	CRC
7	1	6	6	2	46~1500	4

그림 4. 이더넷 프레임 구조

$$T_{BT} = \alpha \text{ clock } (\alpha : \text{preamble 부터 FCS 까지의 이더넷 프레임의 총 비트 길이}) \quad (1)$$

$$T_{BW} = \alpha \times (B / R - 1) \text{ clock} \quad (2)$$

$$T_{IFG} = B / R \times 96 \text{ clock} \quad (3)$$

식 (1)(2)(3)에 의해 실제 PHY Chip에 대한 입력인 B/4 MHz의 MII(Media Independent Interface) 기준으로 아래의 값들을 얻을 수 있다.

$$T_{BT} = \alpha / 4 \text{ clock} \quad (4)$$

$$T_{BW} = \alpha \times (B / (R \times 4) - 1 / 4) \text{ clock} \quad (5)$$

$$T_{IFG} = B / (R \times 4) \times 96 \text{ clock} \quad (6)$$

이제 (4)(5)(6)식으로부터 T_W 를 얻을 수 있다.

$$\begin{aligned}
 T_w &= T_{BW} + T_{IFG} \\
 &= \alpha \times (B/(4 \times R) - 1/4) + B/R \times 8 \times IFG_BYTE \\
 &= (\alpha + 8 \times IFG_BYTES) \times (B/(R \times 4) - 1/4) + 2 \\
 &\quad \times IFG_BYTES \\
 &= P_SIZE \times R_RATE + \beta \tag{7}
 \end{aligned}$$

여기서

$$P_SIZE = \alpha + \gamma, \quad \gamma: 8 \times IFG_BYTES \tag{8}$$

$$R_RATE = B/(4 \times R) - 1/4. \tag{9}$$

$$\beta = 2 \times IFG_BYTES \tag{10}$$

이다. 출력단에서 총 대기 시간(Total Waiting Time)의 계산은 특정 패킷의 전송이 종료되기 이전에 이루어져야 한다. 그런데 이더넷 프레임의 최소 크기는 Preamble과 SFD(Start of Frame Delimiter)를 빼고 64 Byte이므로, 이를 MII를 통해 전송하기 위해서는 $72 \times 2 = 144$ MII 기준 Clock 만큼의 시간이 필요하다. 수식의 계산은 R_RATE 의 값이 테이블(Table) 등을 통해 제공된다면 충분히 이 시간 내에서는 이루어질 수 있다.

3. 실험 결과

이 절에서는 제한한 멀티미디어 데이터 QoS 보장을 위한 멀티 큐에서의 트래픽 조절 방법의 성능을 분석하기 위해 네트워크 모델링 및 시뮬레이션 툴인 OPNET을 사용하여 구현된 시뮬레이션 모델들을 기술하고, 각각의 세부적인 모델들에 대해서 상세히 설명한다. 그림 5는 실험을 위한 전반적인 시나리오를 나타낸다. 4개의 소스노드에서 나오는 각 음성 8 Mbit/s, 영상 50 Mbit/s, 웹 30Mbit/s의 데이터에 대해 분류를 하여 원하는 30 Mbit/s의 원하는 대역폭으로 조절을 할 것이다. 그림6은 소스노드에서 발생하는 트래픽을 보여준다.

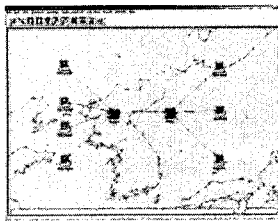


그림 5. 실험 시나리오

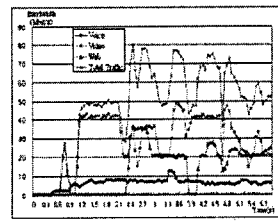


그림 6. 각 노드의 트래픽

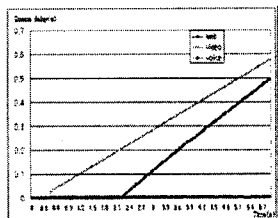


그림 7. 각 서브큐에서의 큐잉 딜레이

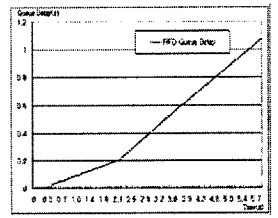


그림 8. FIFO 에서의 큐잉 딜레이

이제 입력 단에서의 대역폭을 출력 단에 맞추어 조절해 준 결과를 보도록 하겠다. 입력 단에서는 급격히 트래픽이 증가하는

부분과 일정하게 유지되는 것이 공존한다. 그러나 대역폭을 조절하여 출력 단에서 일정하게 대역폭이 조절되어 나오는 것을 그림 9에서 보여 준다.

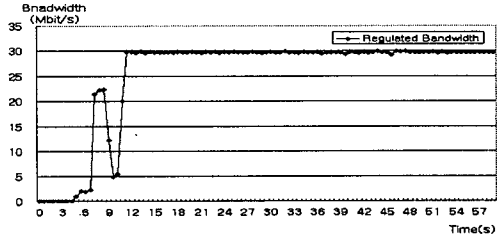


그림 9. 조절된 대역폭

그림 9에서 보듯이 조절되어 나온 대역폭은 일정하게 30 Mbit/s를 유지하고 있다. 유입되는 데이터가 비록 급격한 변화를 가진 트래픽이라 해도 시간 간격을 조절해 줌으로써 일정한 대역폭을 얻어낼 수 있다.

4. 결론

본 논문에서 제시하는 속도 계산 방식은 정확한 결과 값과 단순함 그리고 조절하고자 하는 대역폭의 동적인 변화에 있어서 장점을 지닌다. 또한 멀티미디어 데이터의 특성별로 차별화 시켜 서비스를 행해 줌으로써 한정된 대역폭만을 제공 받는 사용자의 입장에서도 우선 되어야 하는 데이터를 먼저 받음으로써 서비스의 질을 높일 수 있다. 무엇보다 이러한 방식은 현재 라우터나 스위치 등의 QoS 기기에 바로 적용 가능하다는 것이 가장 큰 장점이 되어 이러한 방식을 향후 QoS 기기에 적용시킨다면 네트워크 트래픽 관리 및 운영에 있어서 효율적으로 적용될 수 있을 것으로 기대된다. 본 논문에서는 데이터의 특성 별 차별화와 대역폭 조절 방안만을 제안 하였지만 차후에는 본 방식에 맞는 버퍼 앞단에서의 큐 매니저(queue manager) 방식까지 순차적으로 연구되어야 할 것이다.

참고 문헌

1. S. Shenker, et al., "Specification of the guaranteed quality of service," Internet RFC 22 12, September 1997.
2. Blake, S. et al., "An Architecture for Differentiated Services," Internet RFC 2475, December 1998.
3. R. Braden, et al., "Integrated Services in the Internet Architecture: an Overview," Internet RFC 1633, June 1994.
4. Le Faucheur et al., "MPLS Support of Differentiated Services," Internet RFC 3270, March 2000.
5. R. Braden, et al., "Resource Reservation Protocol (RSVP)Version 1 Functional Specification," Internet RFC 2205, Proposed Standard, September 1997.
6. S. Casner, et al., "RTP: A Transport Protocol for Real-Time Applications," Internet RFC 1889, January 1996.