

# ECN과 문턱값을 겸용한 효과적인 큐 관리기법

정회원 천 상 훈\*

## An Efficient Queue Management Scheme Using Both ECN And Threshold

Sang Hun Chun\* *Regular member*

요 약

최근 인터넷에서 인터넷 프로토콜 기반의 멀티미디어 어플리케이션의 사용이 증가하고 있으나, 네트워크 폭주는 심각한 패킷 손실과 품질저하를 유발하고 있다. 본 논문에서는 어플리케이션 계층에서 전송률의 조정을 돕도록 하기 위해 네트워크의 폭주정보를 활용하는 효과적인 큐 관리기법을 제안하였다. 네트워크 폭주하에서 전송률 조정의 멀티미디어 어플리케이션을 사용하는 제안된 큐관리기법을 평가하였다. NS-2.31을 사용한 실험결과는 패킷들이 수신기에 일정하게 도착함으로써 멀티미디어의 품질을 개선할 수 있음을 보였다.

**Key Words** : ECN, Drop Tail, RED, UDP, Multimedia Application

ABSTRACT

Recently Internet Protocol (IP)-based multimedia applications are being used increasingly in the Internet. However, network congestion brings out severe packet loss and degraded quality. In this paper, we proposed an efficient queue control scheme which utilizes a congestion information obtained from the network to help rate control at the application layer. We estimated the proposed queue control scheme using the rate-controlled multimedia application under network congestion. Simulation results using the NS-2.31 showd that the multimedia quality can be improved from the fact that the packets was uniformly arrived at the receiver.

### I. 서 론

비디오, 이미지, 오디오 등의 양방향 및 네트워크에 기반한 멀티미디어 어플리케이션이 인터넷에서 점점적으로 사용되고 있다<sup>[1]</sup>. 디지털 네트워크의 성공으로 장래의 인터넷 서비스는 인터넷 프로토콜(IP)에 기반한 멀티미디어 어플리케이션을 지원할 것이 확실시되고 있다<sup>[2]</sup>. 그러한 어플리케이션으로는 스티리밍 비디오/오디오, 원격교육, e-commerce, 엔터테인먼트 등이 있다. 대부분의 인터넷 기반 실시간 멀티미디어 서비스는 전송 프로토콜로 UDP (user datagram protocol)를 채택하고 있다<sup>[3]</sup>. TCP

(transmission control protocol)<sup>[4]</sup>에 비해서 UDP는 전송지연을 발생시키지 않으며 이는 지연에 민감한 어플리케이션에 장점으로 작용한다. 인터넷에서 UDP 패킷 전송은 경쟁방식으로 진행되기 때문에 네트워크 폭주는 패킷 손실을 발생시키고, 수신측의 호스트에는 패킷이 도달 아니면 손실이 된다.

UDP는 네트워크 폭주로 인하여 패킷이 손실되어도 이에 대처할 수 없으며 UDP를 이용하는 어플리케이션의 서비스 품질은 이러한 패킷손실로 인하여 저하된다. 최근에 폭증하고 있는 인터넷을 통한 멀티미디어 서비스에 대한 요구는 네트워크 폭주를 큰 폭으로 증가시키고 있으며, 이러한 네트워크 폭

\* 재능대학 정보통신과 (altari@mail.jnc.ac.kr)

논문번호 : 08004-0115, 접수일자 : 2008년 1월 15일

주에 적절히 대처하지 못하는 경우 서비스 품질의 심각한 저하를 유발할 수 있다.

본 논문에서는 네트워크 폭주의 상황을 수신단에 경유하여 송신단에 알려서 멀티미디어 어플리케이션의 전송률을 조절할 수 있도록 하기위한 큐 관리기법을 제안하였다. 제안된 큐 관리기법은 라우터의 큐 길이가 특정 문턱 값을 넘어서는 경우에 폭주정보를 패킷에 표시하여 종단에 전달하는 방법을 사용하였다. 본 논문의 구성은 다음과 같다. 2장에서는 제안한 큐 관리기법의 동작에 대해서 자세히 설명하고, 3장에서는 시뮬레이션 결과를 다루고, 4장에는 결론을 제시하였다.

## II. 제안한 큐 관리기법

기존의 라우터의 큐 관리기법은 버퍼가 오버플로우할 때만 패킷을 폐기함으로써 라우터의 패킷 큐의 길이를 관리한다. 라우터는 설정된 라우터의 패킷 큐의 최대값이 초과될때까지 패킷을 받아들이지만, 그 이후에 들어오는 패킷을 폐기하게 된다. 이것이 드롭테일(Drop Tail)기술이다. 이 기술은 다음의 단점이 있다. 즉 모든 플로우로부터의 패킷이 폐기되기 때문에, 폭주에 참가하지 않은 플로우는 의도하지 않은 손해를 보게된다.

RED(random early discard)는 드롭테일의 단점을 보완하는 폭주회피기법이다. 큐가 버퍼를 넘기전에 확률적으로 패킷을 폐기한다. 드롭 확률은 평균 큐 크기에 따라 결정된다. RED 라우터는 랜덤하게 패킷을 폐기한다. RED는 두개의 문턱값  $minth$ 와  $maxth$ 를 사용하여 동작한다. 평균 큐 크기가  $minth$ 와  $maxth$  사이인 경우 확률  $Pa$ 로 패킷이 폐기되고, 여기서  $Pa$ 는 평균 큐 크기의 함수이며 선형적으로 설정된 최대드롭확률  $maxp$ 로 증가,  $maxth$ 를 넘어서면 확률은 100%가 된다.

ECN(explicit congestion notification)은 RED의 확장을 위해 제안된 기법중의 하나로, 평균 큐 크기가  $minth$ 와  $maxth$  사이에 있을때 패킷을 폐기하지 않고 표시한다. ECN은 종단 호스트 뿐만아니라 라우터의 도움도 필요로한다. ECN이 가능하지 않은 플로우의 패킷은 이전처럼 계속해서 RED에 의해 폐기된다<sup>[5]</sup>.

그림 1은 본 논문에서 멀티미디어 어플리케이션의 데이터를 전송하는데 사용한 프로토콜 스택과 데이터 유닛을 나타낸다.

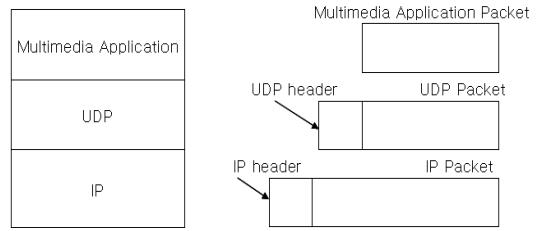


그림 1. 프로토콜 스택과 패킷 구조

멀티미디어 어플리케이션 패킷에 UDP와 IP에 관련된 헤더를 붙인다. ECN기능을 부여하기 위해 IP 헤더의 ECT 비트를 1로 설정하고 CE 비트를 0으로 초기화하여 전송한다.

그림 2는 ECN 기능과 문턱값을 겸용하는 본 논문에서 제안한 큐 관리기법의 흐름도를 나타낸다. IP 패킷이 도착하면, 라우터는 IP 헤더의 ECT(ECN capable transport) 비트가 1로 설정되어 있는 패킷과 그렇지 않은 패킷을 구분한다. ECT 비트가 1로 설정되어 있는 패킷에 대해서는, 큐 길이가 문턱값보다 크면 IP 헤더의 CE(congestion experienced) 비트를 1로 설정하고, 큐 길이가 문턱값보다 작으면 CE 비트를 1로 설정하지 않고 수신단으로 전송한다. ECT 비트가 1이 아닌 패킷에 대해서는 큐 길이가 큐의 크기보다 크면 폐기하고 그렇지 않으면 수신단으로 전송하는 일반적인 큐 관리 방식을 따른다.

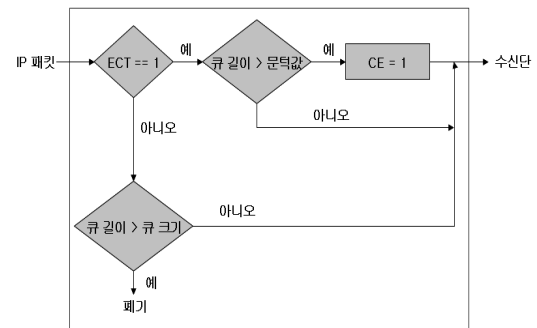


그림 2. 제안한 큐 관리기법의 흐름도

수신단에서는 ECT 비트가 1로 설정된 패킷과 그렇지 않은 패킷을 구분하여 처리한다. ECT 비트가 1인 패킷을 수신한 경우, 폭주정보를 추출하여 송신단에 알려서 전송률을 조정토록할 수 있다.

## III. 모의 실험 결과

모의실험에는 그림 3의 망 모델을 이용하였다.

송신원 S1은, 전송률의 조정이 가능한 멀티미디어 어플리케이션이 UDP 전송프로토콜을 통해서, 라우터 R1에 5Mbps, 3ms의 지연을 갖는 링크를 통해 연결된다. 송신원 S2는, On/Off 트래픽 소스가 UDP 전송프로토콜을 통해서, 라우터 R1에 5Mbps, 3ms의 지연을 갖는 링크를 통해 연결된다.

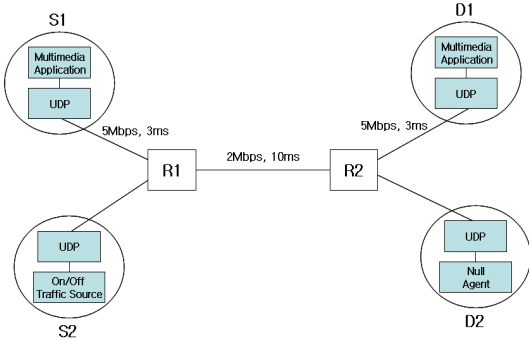


그림 3. 실험에 사용한 덤벨 망 모델

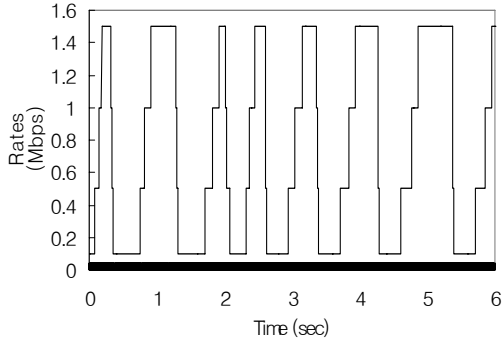
라우터 R1은 2Mbps, 10ms의 지연을 갖는 링크를 통해 라우터 R2에 연결되며, 라우터 R2는 제한한 큐 관리기법을 장착하고 5Mbps, 3ms의 전송지연을 갖는 링크를 통해 수신원 D1, D2에 각각 연결된다.

수신원 D1은 송신원 S1으로부터의 패킷을 수신하고 폭주정보를 송신원에 ACK 패킷에 실어서 보낸다. 수신원 D2는 송신원 S2로부터의 패킷을 수신한다. 라우터 R1과 R2 간의 링크를 송신원 S1과 송신원 S2가 공통으로 사용하게 함으로써 트래픽의 병목현상을 발생하게 하였다.

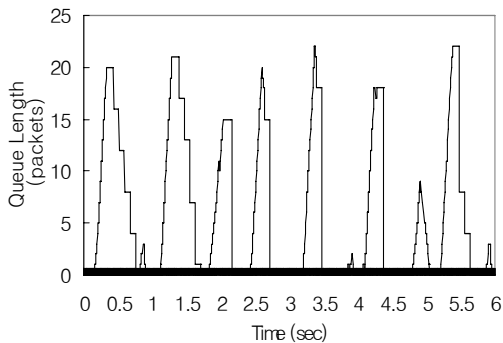
라우터 R2에서 폭주에 의한 패킷의 폐기를 방지하기 위해 큐 크기를 100 패킷으로 충분히 높게 잡았고, 폭주에 신속히 반응하게 하기위해 문턱값을 큐 크기의 10%인 10 패킷으로 설정하였다. 송신원 S1에는 폭주에 따라 전송률을 0.1Mbps, 0.5Mbps, 1.0Mbps, 1.5Mbps로 가변시키면서 1000 바이트 크기의 데이터 패킷을 UDP 패킷에 실어서 수신원 D1에 전송하는 멀티미디어 어플리케이션을 실행시켰다. 송신원 S2에는 평균 버스트 구간과 평균 아이들 구간을 각각 100ms, 버스트 전송률을 1.5Mbps, 패킷크기를 1000바이트로 설정한 On/Off 트래픽을 실행시켰다. 송신원 S1과 S2를 0초에 동시적으로 데이터의 전송을 시작하였고 6초 동안 수행하였다. 다음은 ns2에서 작성한 시뮬레이션 스크립트의 일부를 나타낸다.

```
#Setup a MM UDP connection
set udp_s [new Agent/UDP/UDPmm]
set udp_r [new Agent/UDP/UDPmm]
$ns attach-agent $node_(s1) $udp_s
$ns attach-agent $node_(s3) $udp_r
$ns connect $udp_s $udp_r
$udp_s set packetSize_ 1000
$udp_r set packetSize_ 1000
$udp_s set fid_ 1
$udp_r set fid_ 1
#Setup a MM Application
set mmapp_s [new Application/MmApp]
set mmapp_r [new Application/MmApp]
$mmapp_s attach-agent $udp_s
$mmapp_r attach-agent $udp_r
$mmapp_s set pktsize_ 1000
$mmapp_s set random_ false
#Setup a OnOff Application's UDP connection
set udp_onoff [new Agent/UDP]
set null0 [new Agent/Null]
$ns attach-agent $node_(s2) $udp_onoff
$ns attach-agent $node_(s4) $null0
$ns connect $udp_onoff $null0
$udp_onoff set packetSize_ 1000
$udp_onoff set fid_ 2
#Setup an Exponential OnOff Application
set e [new Application/Traffic/Exponential]
$e set packetSize_ 1000
$e set burst_time_ 100ms
$e set idle_time_ 100ms
$e set rate_ 1.5mb
$e attach-agent $udp_onoff
record 0.005
#Simulation Scenario
$ns at 0.0 "$mmapp_s start"
$ns at 0.0 "$e start"
$ns at 6.0 "finish"
$ns run
```

평가를 위해 송신원 S1의 전송률과 라우터 R2의 큐 길이, 수신원 D1에 도달하는 패킷, 송신원 S1에 도달하는 ACK 패킷을 0.005초의 시간간격으로 측정하였다.



(a) 송신원 S1의 전송률



(b) 라우터 R2의 큐 길이

그림 4. 송신원 S1의 전송률과 라우터 R2의 큐 길이

그림 4는 송신원 S1의 전송률과 라우터 R2의 큐 길이를 나타낸다. 그림4로부터 약 0초 이전에는 큐의 길이가 거의 0에 가깝고 이 기간 동안에는 네트워크 폭주가 없음을 알 수 있다. 같은 기간동안 송신원 S1은 전송률을 초기값 0.1Mbps에서 시작하여 최고값 1.5Mbps까지 증가함을 알 수 있다. 약 0.2초에서 0.6초 사이에서는 큐의 길이가 10 이상이기 때문에 네트워크에 폭주 있음을 알 수 있다. 이 기간동안 송신원 S1은 전송률을 감소시킴을 알 수 있다. 이러한 패턴이 6초 동안의 실험기간에 걸쳐 반복됨을 그림4로부터 확인할 수 있다.

그림 5는 수신원 S2에 도달한 데이터 패킷과 송신원 S1에 도달한 ACK 패킷을 나타낸다. 90개 단위로 출력하기 위해 시퀀스 번호를 90으로 나눈후 나머지 값에 1을 더하여 표시하였다. 기호  $\bullet$ 는 데이터 패킷을, 기호  $\circ$ 는 ACK 패킷을 나타낸다. 데이터 패킷의 경우 꺾이는 부분은 네트워크 폭주로 인하여 전송률이 0.1Mbps로 조정되어 패킷 간의 지연이 상대적으로 다른 부분에 비해서 길기 때문에 나타나는 현상이다.

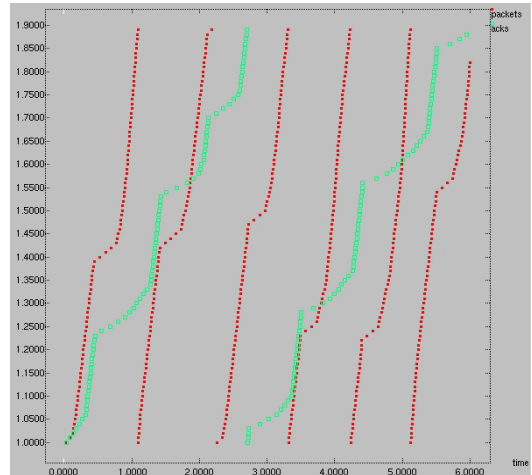


그림 5. 수신원 S2에 도달한 데이터 패킷과 송신원 S1에 도달한 ACK 패킷

ACK 패킷의 경우 기울기가 급한 부분은 네트워크 폭주로 인하여 폭주시마다 ACK 패킷이 발생되기 때문이며, 기울기가 완만한 부분은 네트워크에 폭주가 없기 때문에 왕복지연시간 단위로 ACK 패킷이 발생되기 때문이다. 위 그림으로부터 네트워크에 폭주가 심하게 발생한 짧은 구간을 제외하고는 멀티미디어 데이터 패킷이 일정한 간격으로 수신원에 도달함을 알 수 있다. 따라서 네트워크 폭주에 신속히 대응하여 멀티미디어 어플리케이션의 서비스 품질을 크게 저하시키지 않음을 알 수 있다.

#### IV. 결 론

본 논문에서는 멀티미디어 어플리케이션 계층에서 전송률의 조절을 돕도록 하기 위해 네트워크 계층의 폭주정보를 활용하는 효과적인 큐 관리기법을 제안한다. 제안한 기법은 ECN 기법과 문턱값을 겸용하며 폭주정보를 패킷에 표시하여 종단에 전달하는 방법을 사용하였다. 어플리케이션은 폭주정보에 따라서 전송률을 4단계로 가변시킬 수 있도록 하였다. 성능의 평가를 위해서 제안한 멀티미디어 어플리케이션의 전송률과 병목링크의 라우터의 큐 길이, 그리고 수신원에 도착하는 데이터 패킷의 형태를 살펴보았다. 성능의 측정을 위해서 NS-2.31 시뮬레이터를 사용하였다<sup>[6]</sup>. 실험 결과 네트워크 폭주에 신속히 대응하여 패킷들이 수신기에 일정한 간격으로 도착하게 함으로써 멀티미디어 어플리케이션의 서비스 품질이 크게 저하되지 않도록 함을 보였다. 추후의 연구과제로는 ECN 기법을 무선 인터넷 환

경에 적용하는 경우의 성능개선 방법에 대한 연구가 필요하다.

### 참 고 문 헌

- [1] Haitao Zheng and Jill Boyce, "An Improved UDP Protocol for Video Transmission Over Internet-to-Wireless Networks," IEEE Transactions on Multimedia, vol. 3, no. 3, pp. 356-365, September 2001.
- [2] R. pandya, D. Grillo, E. Lycksell, P. Meiybegue, H. Okinaka, and M. Yabusaki, "IMT-2000 standard: Network aspects," IEEE personal communication, Aug. 1997.
- [3] J. Postel, "User Datagram protocol," Request for Comments, RFC 768, ISI, Aug. 1980.
- [4] W. Stevens, *TCP/IP Illustrated*. Reading, MA: Addison-Wesley, 1994, vol. 1.
- [5] Seong-jun Bae, Song Chong, "TCP-friendly flow control of wireless multimedia using ECN marking," *Signal Processing:Image Communication* 19, 2004.
- [6] "The Network Simulator - ns-2," URL <http://www.isi.edu/nsnam/ns>.

천 상 훈 (Sang Hun Chun)

정회원



1990년 2월 인하대학교 전자공학과 졸업

1992년 2월 한국과학기술원 전기전자공학과 석사

2000년 2월 인하대학교 전자공학과 공학박사

2000년 3월~현재 재능대학 정보

통신과 부교수

<관심분야> 폭주제어, 트래픽 제어, 컴퓨터 네트워크, 유비쿼터스