
인터넷에서 멀티미디어 스트리밍을 위한 하이브리드형 TCP-friendly 혼잡제어기법에 관한 연구

조정현* · 나인호*

군산대학교 전자정보공학부

A Study on TCP-friendly Congestion Control Scheme using Hybrid Approach for Multimedia Streaming in the Internet

Jeong-hyun Cho* In-ho Ra*

*Kunsan National University

E-mail : cd20@kunsan.ac.kr

요 약

최근 인터넷의 발전으로 디지털 오디오 및 비디오와 같은 멀티미디어 스트리밍에 대한 요구가 증가하고 있다. 이러한 멀티미디어 스트리밍을 위해 UDP로 전송할 경우 TCP와 같은 혼잡제어를 수행하지 않기 때문에 동일한 경로에 TCP 트래픽 궁핍을 일으켜 혼잡붕괴를 초래한다. 이러한 역효과를 피하기 위해 멀티미디어 스트리밍을 위한 새로운 전송 프로토콜에 대한 연구가 수행되고 있다. TCP 친화적 혼잡제어기법은 크게 일반적인 혼잡윈도우 관리기능을 이용하는 윈도우기반 혼잡제어(window-based congestion control)와 TCP 모델링 방정식 등을 이용하여 전송율을 직접 조절하는 율기반 혼잡제어(rate-based congestion control)로 나눌 수 있다. 본 논문은 윈도우기반과 율기반을 복합적으로 다룬 하이브리드 TCP-friendly 혼잡제어기법에서 전송율 개선을 위한 알고리즘을 제안하였다.

ABSTRACT

Recently the multimedia streaming traffic such as digital audio and video in the Internet has increased tremendously. Unlike TCP, the UDP protocol, which has been used to transmit streaming traffic through the Internet, does not apply any congestion control mechanism to regulate the data flow through the shared network. If this trend is let go unchecked, these traffic will effect the performance of TCP, which is used to transport data traffic, and may lead to congestion collapse of the Internet. To avoid any adverse effect on the current Internet functionality, A study on a new protocol of modification or addition of some functionality to existing transport protocol for transmitting streaming traffic in the Internet is needed. TCP-frienly congestion control mechanism is classified with window-based congestion control scheme and rate-based congestion control scheme. In this paper, we propose an algorithm for improving the transmitting rate on a hybrid TCP-friendly congestion control scheme combined with widow-based and rate-based congestion control for multimedia streaming in the internet.

1. 서 론

최근 정보통신기술의 발전과 인터넷 보급의 확산으로 다양한 멀티미디어 서비스에 대한 수요가 증가하고 있다. 실시간 멀티미디어 스트리밍은 높은 전송 대역폭과 실시간 전송을 필요로 한다. 이러한 특성 때문에 실시간 멀티미디어 스트리밍의

경우 패킷 손실이나 지연, 대역폭과 같은 조건을 크게 고려하여 전송해야 한다. 기존의 UDP로 실시간 멀티미디어 데이터를 전송할 경우 TCP와 같은 혼잡제어를 사용하지 않기 때문에 동일한 경로의 TCP 트래픽 궁핍을 일으켜 혼잡붕괴(congestion collapse)를 초래한다[1]. 또한 실시간 멀티미디어 응용에 TCP 혼잡제어를 적용하면 멀티미디어 사

용자 인지품질이 크게 저하되는 문제점이 있다. 따라서 실시간 멀티미디어 스트리밍에 적합한 TCP 친화적 혼잡제어기법이 필요하게 되었다.

TCP 친화적 혼잡제어기법은 크게 두 종류로 분류할 수 있다. 일반적인 혼잡윈도우 관리기능을 이용하는 윈도우기반 혼잡제어(window-based congestion control)와 TCP 모델링 방정식 등을 이용하여 전송율을 직접 조절하는 율기반 혼잡 제어(model-based congestion control)로 나눌 수 있다.

본 논문은 윈도우 기반과 율기반을 복합적으로 다룬 하이브리드형 TCP 친화적 혼잡제어기법을 제안하였다. 본 논문의 구성은 다음과 같다. 먼저 II장에서는 TCP 친화적 혼잡제어 기법의 특징과 방법에서 대해 기술하고, III장에서는 본 논문에서 제안한 혼합형 TCP 친화적 혼잡제어 기법에 대해 설명한다. 마지막으로 IV장에서는 결론 및 향후 연구내용에 대해 기술한다.

II. TCP-friendly 혼잡제어기법

1. 윈도우기반 혼잡제어

윈도우기반 혼잡제어는 혼잡윈도우를 이용하여 전송율을 조절하는 방법이다. 송신측은 혼잡 윈도우의 최대크기로 패킷을 전송할 수 있으며, 수신측은 수신한 패킷마다 Ack를 보냄으로써 Ack가 송신호스트의 전송속도를 조절하게 한다. 즉 송신측은 Ack 간격에 의해 패킷을 전송한다. 이와 같은 윈도우 기반 혼잡 제어의 특성을 'Ack-clocking'라 한다[2]. 이러한 특성을 가진 대표적인 윈도우기반 혼잡 제어로는 AIMD(Additive increase multiplicative decrease), GAIMD(Geneal Additive increase/ multiplicative decrease), Binomial등이 있다.

1.1 AIMD 혼잡제어

TCP는 AIMD를 이용하여 혼잡제어를 수행한다. 송신측은 가용대역폭을 사용하기 위해 혼잡윈도우 크기를 선형적으로 증가하고, 혼잡지시자인 패킷 손실이 발생할 때는 혼잡윈도우크기가 감소한다.

$$I: w_{t+R} \leftarrow w_t + \alpha; \alpha > 0$$

$$D: w_{t+\delta t} \leftarrow (1 - \beta)w_t; 0 < \beta < 1 \quad (1)$$

일반적으로 TCP에서는 $\alpha=1$, $\beta=1/2$ 로 설정한다. 이러한 TCP AIMD방식은 효율적으로 가용대역폭을 사용하기 때문에 대부분의 응용에 사용되고 있다[3].

1.2 GAIMD 혼잡 제어

GAIMD는 AIMD와 마찬가지로 RTT동안 패킷 손실이 발생하지 않으면 α 만큼 증가하고, 패킷 손실이 발생하면 β 만큼 감소한다. 즉 AIMD방식을

확장하여 일반화한 것이다. GAIMD가 TCP 친화적이기 위해서는 α 와 β 가 다음과 같은 관계를 가져야 한다.

$$\alpha = \frac{4(1-\beta^2)}{3} \quad (2)$$

GAIMD는 멀티미디어 응용에 적합한 평활화한 전송을 위해 $\beta=7/8$ 로 설정한다. 따라서 식(2)에 의해 α 는 0.31의 값을 가지게 된다.

1.3 Binomial 혼잡제어

Binomial 혼잡제어기법은 선형 혼잡제어를 수행하는 AIMD와 달리 비선형 혼잡제어를 수행하여 가용대역폭을 보다 빨리 사용할 수 있는 기법이다.

$$I: w_{t+R} \leftarrow w_t + \alpha/w_t^k; \alpha > 0$$

$$D: w_{t+\delta t} \leftarrow w_t - \beta w_t^l; 0 < \beta < 1 \quad (3)$$

여기서 Binomial이 TCP 친화적이기 위해서는 적절한 α , β 값과 $k+l=1$, $l < 1$ 의 조건을 만족하면 된다. AIMD보다 급격한 감소가 덜하지만 여전히 멀티미디어 응용에 적합하지 않다[3][5].

2. 율기반 혼잡 제어

율기반 혼잡 제어는 혼잡을 지시하는 네트워크 피드백 기법을 이용하여 동적으로 전송율을 조절하는 기법이다. 대표적으로 simple AIMD방식을 이용하는 RAP(rate adaptation protocol)과 TCP 모델링방정식을 이용하는 TFRC 등이 있다.

2.1 TCP 모델링 방정식

TCP 처리율은 왕복시간 t_{RTT} 재전송타임아웃 t_{RTO} 패킷크기 s , 패킷손실율 p 를 근거로 하여 구할 수 있다.

$$T = \frac{s}{RTT \sqrt{\frac{2p}{3}} + T_r \min\left(1, 3\sqrt{\frac{3p}{8}}\right) / (1 + 32p^2)} \quad (4)$$

2.2 RAP(Rate adaptation protocol)

RAP는 유니캐스트 흐름을 위한 AIMD방식을 이용한 프로토콜이다. RAP 송신측은 전송패킷에 대한 Ack를 수신 측으로부터 받는다. 이러한 Ack는 패킷손실과 왕복시간을 측정하기 위해 사용된다. RAP는 패킷손실이 발생할 때 전송율을 절반으로 감소시키고, 패킷손실이 발생하지 않으면 RTT마다 패킷크기만큼 증가시킨다. 이러한 증가/감소는 RTT간격으로 결정된다. RAP는 단기간 RTT와 장기간 RTT비를 이용하여 연속적인 패킷간격을

조정하는 fine-grained delay-based congestion control를 사용함으로써 멀티미디어 응용에 적합한 평활화한 전송이 가능하다. 그러나 RAP는 타임아웃을 고려하지 않아 TCP 처리율이 타임아웃에 지배적인 경우 공격적인 행동을 한다는 문제점을 가지고 있다[2][7].

2.3 TFRC(TCP-friendly Rate control)

TFRC는 TCP 모델링방정식을 이용하여 전송율을 조절하는 방법으로 여기서는 식 (4)를 사용하는데 있어 정확한 패킷 손실율을 측정하는 것이 중요하다. 따라서 TFRC는 연속적인 패킷손실간의 패킷수를 측정하는 평균손실간격(Average Loss Interval)과 EWMA기법을 이용하여 패킷손실율을 측정한다. TFRC 수신측은 매 RTT마다 TCP방정식에 필요한 파라미터를 업데이트하여 송신측에 보내고 송신측은 업데이트된 정보를 통해 계산한 전송율로 전송한다. TFRC는 지연기반 혼잡제어(delay-based congestion control)을 사용하여 프로토콜의 성능을 향상시키고 있으며, 경쟁하는 트래픽에 대해 충분한 반응을 보임으로써 상대적으로 안정된 전송율을 유지할 수 있다.[6][7]

III. 하이브리드 TCP-friendly 혼잡제어기법

TEAR는 윈도우기반측면과 올기반측면을 결합한 하이브리드 TCP-friendly 혼잡 제어 기법 중 대표적인 프로토콜[7][8]이며, 이것을 기반으로 TEAR 전송율을 개선하기 위한 알고리즘을 제안하였다.

1. TEAR(TCP emulation at receivers)

TEAR 수신측은 혼잡윈도우크기를 관리하고 이를 토대로 전송율을 계산한다. 계산된 전송율을 피드백을 통해 송신측에 전송하고 송신측은 이를 토대로 전송율을 결정한다.

• Round

TEAR는 혼잡윈도우를 수신측에서 관리하므로 RTT대신 round개념으로 혼잡윈도우를 업데이트된다.

• TEAR 상태 전이

TEAR는 그림 1과 같이 7가지 상태로 구성되어 있다.

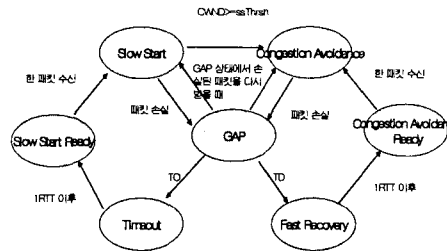


그림 1 TEAR 상태전이

- ① Slow-start-ready
- ② Slow-start
- ③ Congestion-avoidance-ready
- ④ Congestion-avoidance
- ⑤ Fast-recovery
- ⑥ Timeout
- ⑦ Gap

• 혼잡윈도우 증가 알고리즘

Slow-start, Congestion-avoidance 상태에서 하나의 패킷을 수신하는 경우 혼잡윈도우는 그림2와 같이 동작하게 된다.

여기서 lastcwnd는 매 round가 시작할 때마다 다음 round의 혼잡윈도우 증가를 계산하기 위해 현재 혼잡윈도우로 업데이트된다.

```

IncreaseWindow()
{
    Switch(state){
    case Slow_Start:
        cwnd = cwnd + 1;
        if(ssThresh <= cwnd)
            cwnd = ssThresh;

    case Congestion_Avoidance:
        cwnd += 1/lastcwnd;
    }
}
    
```

그림 2 혼잡윈도우 증가 알고리즘

• 혼잡 윈도우 감소 알고리즘

패킷 손실이 발생하면 Slow-start, Congestion-avoidance상태는 Gap상태로 전이된다. Gap 상태에서는 혼잡윈도우크기가 변하지 않는다. Gap 상태는 수신측이 패킷 손실이 타임아웃에 의한 것인지, 세 개의 중복 Ack에 의한 것인지를 결정하는 중계상태이다.

① 빠른 회복(Fast recovery)

손실된 패킷을 l 이라고 가정하자. 수신측이 패킷 손실 이후 l lastcwnd 패킷을 수신하기 전에 적어도 2개의 패킷을 수신하게 되면 Gap상태에서 Fast-recovery상태로 전이하게 된다. 이 상태에서 수신측은 혼잡 윈도우와 lastcwnd를 절반으로 감소한다.

② 타임아웃(Timeout)

수신측이 손실된 패킷 l 를 인지한 후 $T_{timeout}$ 동안 Gap상태에서 Fast recovery상태로 전이되지 못하면 Timeout 상태로 전이하게 된다. 이 상태에서 수신측은 혼잡윈도우와 lastcwnd를 1로, $T_{timeout}$ 는 두 배로 설정한다.

```

DecreaseWindow()
{
    Switch(state){
    case Fast_Recovery:
        cwnd = cwnd/2;
        ssThresh = cwnd;
    case Time_Out:
        cwnd = 1;
    }
}
    
```

그림 3 감소 알고리즘

2. TEAR 전송율 개선을 위한 알고리즘

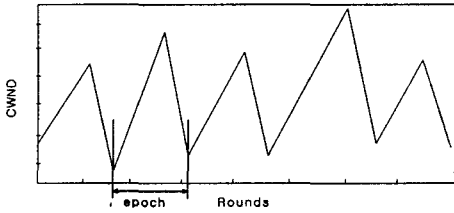


그림 4 epoch

제안한 알고리즘은 TCP의 톱니형태의 전송율 변화를 피하기 위해 매 RTT 대신 epoch단위로 전송율을 계산하게 한다. epoch는 연속적인 율 감소 사건간의 시간으로 정의된다. 그림 4는 epoch를 나타내고 있다. 노이즈에 의한 불필요한 율 변화를 방지하기 위해 지난 w개 epoch의 전송율에 대한 가중치 평균을 사용하여 평활화한 전송율을 계산하고, 계산된 전송율은 피드백을 통해 송신측에 보낸다. 송신측은 이를 토대로 전송율을 결정하기 앞서 인터넷과 같은 다양한 네트워크 상태에 적용하도록 전송율 T를 식(5)와 같이 조절하였다.

$$T = T_{estimate} \times \frac{\sqrt{SRTT}}{RTT_{current}} \quad (5)$$

여기서 $T_{estimate}$ 는 수신측에서 계산된 전송율이다. 제안한 기법은 현재의 네트워크 상태를 예측하는 가장 좋은 방법인 네트워크 지연을 토대로 혼잡 회피를 수행하면서 다양한 네트워크 환경에 적용 가능하도록 하였다. 그림 5는 전체적인 동작 메카니즘이다.

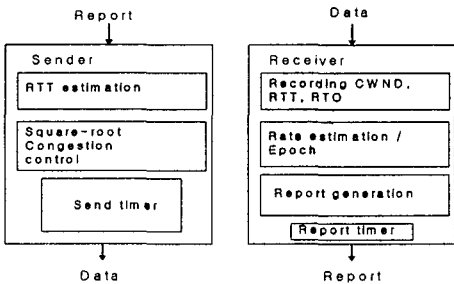


그림 5 동작 메카니즘

IV. 결론 및 향후과제

본 논문은 멀티미디어 스트리밍을 위한 TCP 친화적 혼잡 제어기법을 다루었다. TCP와 경쟁시 대역폭을 공정하게 나누어 사용할 수 있게 TCP 동작을 적용하였고, 갑작스런 율 감소에 의한 멀티미디어 응용의 사용자 인지 품질이 저하되는 것을 피하기 위해 평활화한 전송을 할 수 있도록 epoch단위로 전송율을 측정하였다. 또한 네트워크 지연으로 인한 혼잡을 피하기 위한 방법을 제시하였다. 또한, 향후 시뮬레이션을 통해 제안된 알고리즘의 성능을 검증하고 멀티캐스트상에서 동작할 수 있도록 확장하고자 한다.

참고 문헌

- [1] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. IEEE/ACM Transactions on Networking, Aug 1999.
- [2] R.Regai, M. Handley, and D. Estrin. RAP : An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the internet. In Proceedings of IEEE INFOCOM, 1999.
- [3] S. Jin, L. Guo, I. Matta, A. Bestavros. A Spectrum of TCP-friendly Window-based Congestion Control Algorithms. July 2002.
- [4] Y. Richard Yang and Simon S. Lam. General AIMD congestion control. In Proceedings of ICNP, November 2000.
- [5] D. Bansal and H. Balakrishnam. Binomial congestion control algorithms. In Proceedings of IEEE INFOCOM, April 2001.
- [6] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. Technical Report, ACIRI, Feb 2000.
- [7] J. Widmer, R. Denda, M. Mauve. A Survey on TCP-Friendly Congestion Control(extended version). Feb 2001.
- [8] I. Rhee, V. Ozdemir, and Y. YI. TEAR : TCP Emulation At Receivers - flow control for multimedia streaming, Technical Report, NCSU. April 2000.