제36회 한국정보처리학회 추계학술발표대회 논문집 제18권 제2호 (2011. 11)

WLAN에서 최소전송률을 보장하는 TFRC 제어방법

이강섭*, 최승식**
*인천대학교 컴퓨터공학과
**인천대학교 컴퓨터공학과
e-mail : lks5030@incheon.ac.kr

TFRC Control Method Guaranteed Minimumrate in Wireless-LAN

Kang-Seob Lee*, Seung-Sick Choi**
*Dept of Computer Science, Incheon University
**Dept of Computer Science, Incheon University

요 약

본 논문에서는 유 무선 이종망의 네트워크에서 혼잡손실과 무선손실의 구별을 하지 못하고 전송률을 감소하는 TFRC의 성능을 개선하기 위한 최소 전송률을 보장하는 TFRC 제어방법을 제안한다. 제안하는 TFRC기법은 기존의 TFRC기법의 패킷손실률에 따른 loss event rate를 제한하고 피드백 타임아웃에 따른 전송률 감소를 제한하여 비디오의 최소전송률을 보장하는 방법이다. TCP와 제안하는 TFRC를 같은 네트워크망에서 경쟁하는 환경으로 실험 했을 때 기존의 TFRC와 비교해서 제안한 TFRC의 전송률이 보장되어 더 좋은 성능을 보였다. 무선구간에서의 손실까지도 혼잡손실로 판단하고 전송률을 감소시켜 다중 네트워크에서 대역폭을 보장받지 못하는 TFRC기법을 최소전송률 보장을 통해 비디오 스트리밍 서비스를 제공함으로써 서비스의 품질을 보장한다.

1. 서론

IEEE 802.11 무선 랜과 같은 광대역 무선망의 급속한 확대와 3G에 이은 4G 이동통신망이 상용화되기 시작하면서 무선 네트워크의 사용이 크게 증가하고 있다. 특히 태블릿 PC와 스마트폰의 보급 확대가 이에 큰 기여를 하였다. 이에 따라 무선 모바일 상에서 인터넷 서비스를 이용하는 사용자가 많아졌다. 인터넷의 큰 비중을 차지하는 멀티미디어 응용 프로그램 서비스, 즉 비디오 스트리밍 서비스 역시 마찬가지다. 특히 스마트폰이나 태블릿PC 등의 유튜브 서비스처럼 last-hop을 무선구간으로하는 이종망에서 멀티미디어 전송시스템 연구가 활발히 연구되고 있다[1],[2].

대부분의 멀티미디어 스트리밍은 혼잡제어를 수행하지 않는 UDP 프로토콜을 사용한다. 그러나 다중 네트워크에서 TCP와 UDP를 사용하는 어플리케이션들이 같은 혼잡상태의 체널을 이용하는 경우 문제가 생긴다. TCP프로토콜을 이용하는 어플리케이션은 혼잡에 의해서 전송률을 제어하게 되는 반면 UDP는 전송률을 유지하게 될 것이다. 따라서 TCP가 UDP에 비해 대역폭을 더 적게 사용하는 불공평성이 발생한다. 이러한 문제점을 보안하기 위해서 TCP Reno의 혼잡제어를 모델링한 TFRC(TCP Friendly Rate Control)제어방법 을 사용하게 되었다. TCP와 유사하게 패킷 손실에 따라 흐름제어를 하여 TCP와 공평하게 대역폭을 유지 할 수 있는 것이다.

하지만 서두에서 언급한 스마트폰 같은 무선모바일을 last-hop으로 하는 환경에서 TFRC기법은 무선구간에서의 패킷손실을 유선구간의 손실과 더불어 같은 패킷 손실로 파악한다. 무선구간에서의 손실 역시 흐름제어를 하게 되고 이에 따른 전송률 감소로 무선 모바일에서 원활한 영상 스트리밍 지원에 문제가 생기게 된다.

2. 관련연구

네트워크상에서 TCP와 균등하게 전송률을 유지하기 위해서 TCP 친화적인 프로토콜들이 제안되고 있다. TFRC 기법 또한 TCP 친화적인 프로토콜들 가운데 하나로서 Floyd, S가 제안했다[3],[4]. TFRC(Tcp Friendly Rate Control) 프토로콜은 TCP 프로토콜과 유사하게 혼잡손실에 따라서 전송률을 높이고 줄임으로써 TCP 친화적으로 동작하는 방법으로 TCP의 전송률 방정식을 모델링하였다. 수신측에서 패킷들을 전송받는 중에 손실이 발생하면 이를 기억하고 있다가 현재의 RTT 만큼의 시간이 지난 뒤부터 패킷의 손실이 또 발생하면 두 패킷 사이의 패킷 수를 통해 패킷손실률을 계산한다. 가장 최근 패킷이 전송되었을 때 까지 8개의 RTT동안 패킷 손실률을 수신버퍼에 유지하고 이들의 평균손실률을 계산한다. 이 계산

된 값에 따라서 손실에 따른 전송률을 송신측으로 피드백하게 되고 피드백 받은 전송률로 다시 패킷을 전송하게 된다[2],[4],[5].

이와 더불어 피드백 되는 패킷들이 네트워크의 혼잡으로 손실되는 경우도 발생한다. 이 경우에 피드백 된 정보가 없어 전송률을 조절할 수 없게 된다. TFRC 프로토콜에서 트래픽 전송을 시작할 때 RTT Timer와 함께 No Feedback Timer를 시작한다. 패킷을 전송하고 피드백을 받는 동작 중에 혼잡으로 인해 피드백에 대한 손실이 발생하게 되고 No Feedback Timer의 시간 동안 피드백이 수신되지 않으면 TFRC 기법은 전송률을 절반으로 감소시키는 동작을 수행한다. 이는 TFRC 프로토콜이 TCP Reno의 동작을 모델링 한 것으로 TCP Reno 프로토콜에서 세 개의 중복 ack가 발생할 경우 혼잡윈도를 절반으로 줄이는 동작을 모델링 한 것이다[6].

SLD 기법은 도착한 패킷들의 상태를 구분하여 혼잡의 정도를 파악하는 방법이다. 정상적으로 도착한 패킷과 손실된 패킷, 혼잡상태에서 마킹된 패킷의 비율로 혼잡의 정도를 결정하여 전송률을 제어하는 방법이다[1].

WM-TFRC는 무선적응레이어(WAL)를 추가하여 이 무선구간에서 전송되는 패킷으로부터 패킷손실정보를 피드백 받아 평균적인 무선채널의 손실을 구하여 혼잡손실과 구분하는 방법이다[7].

3. 최소전송률을 보장하는 TFRC기법

본 논문에서 제안하고자 하는 방법은 순수한 TFRC프로토콜에 기반으로 하여 수정한 TFRC기법이다. 수정한 방법은 전송률을 임의의 수치 이하로는 내려가지 않게 하여 최소 전송률을 보장하는 TFRC 기법이다. 이는 손실의 원인을 구별하여 전송률을 계산하는 방법이 아닌 직접적으로 전송률의 수치를 조정하여 이를 보장하는 방법이다.

전송률을 제어하는 알고리즘은 다음과 같다. 네트워크 채널에서 손실이 발견되고 이에 의해 TFRC 기법으로 패킷손실률을 구하고 loss event rate를 계산한다. 그 이후에 계산된 loss event rate에 따라 변경할 전송률을 피드백하게 되는데 그 값을 최소 전송률로 설정한 Rmin값과 비교한다. 만약 변경할 전송률이 제한하는 최소전송률 보다 크다면 다음 패킷을 보내기 위한 전송률을 loss event rate에 의해 계산된 전송률로 유지하여 피드백 한다. 이와 다르게 변경할 전송률이 제한하는 최소전송률보다 작은 경우에는 피드백할 전송률을 제한하고자 하는 전송률로 변경한다. 그 이후에 패킷을 변경된 전송률로 보낸다.

```
Most recent received packet have losses
set the loss event rate();
decrease rate();
if(rate < Rmin)
        rate = Rmin;
change state to Rate Decrease
send next packet();
```

(그림 1) 피드백에서 손실에 따른 전송률 조정

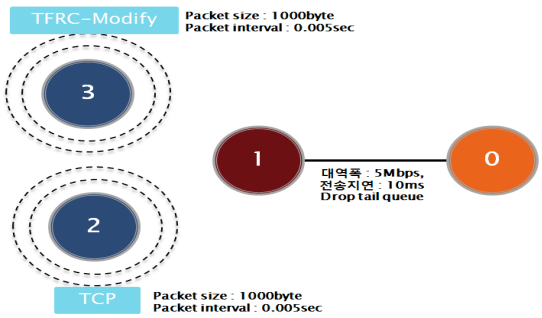
더불어 패킷이 송신된 뒤에 no feed back timer가 동작하고 피드백이 돌아오기를 기다린다. 만약 채널에서 피드백이 손실되어 돌아오지 못할 경우 전송률을 재 조절하고 패킷을 전송해야 하기 때문이다. 이렇게 피드백이 손실되어 돌아오지 않고 no feed back timer가 타임아웃 되었을 때 rate를 절반으로 줄이는 과정에서 절반으로 줄인 rate의 값을 제한하고자 하는 최소전송률(Rmin)과 비교한다. 절반으로 줄인 rate의 값이 최소 전송률보다 크다면 제한없이 전송률을 변경하여 다시 패킷을 보내게 된다. 만약 전반으로 준인 전송률이 제한한 최소전송률 보다 작다면 다음 패킷을 보내기 위한 rate를 Rmin으로 조정 한 뒤에 다시 전송한다.

```
TFRC No feedback Timer is expire

Event -> reduce rate

rate * = 0.5;

if(rate < Rmin)
        rate = Rmin;

reset the no feed back timer
Change state to Congestion Avoid

send next packet();
```

(그림 2) TFRC no feed back timer에 대한 알고리즘

4. 실험 성능 및 평가

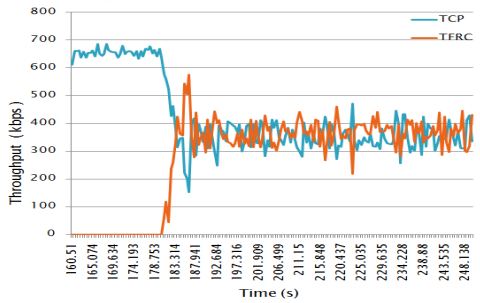
본 논문에서 제안하는 TFRC기법의 성능을 알아보기 위해서 네트워크 시뮬레이터 NS2를 사용하였다[9],[10]. 우선 시뮬레이션을 위한 시나리오는 아래 (그림 3) 과 같이 환경을 설정 하였다.



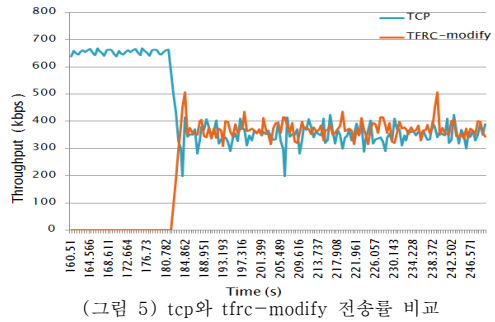
(그림 3) 시뮬레이션 환경

각각의 무선 노드 3, 2에서 TFRC-Modify 프로토콜과 TCP프로토콜에 의해 1000byte의 패킷을 노드 1에 걸쳐 노드 0까지 전송한다. 실험을 시작한 뒤 160초 이후부터 TCP패킷의 전송을 시작하고 180초 이후에 TFRC-modify의 전송을 시작한다. 패킷은 CBR 트래픽을 5ms의 지연으

로 전송한다. 시뮬레이션은 시작 후 250초에 종료 된다.



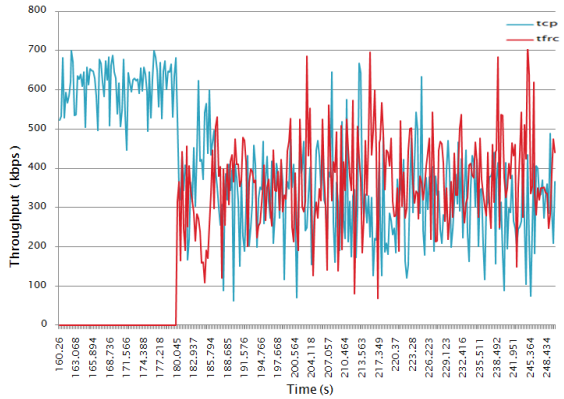
(그림 4] tcp와 tfrc 전송률 비교



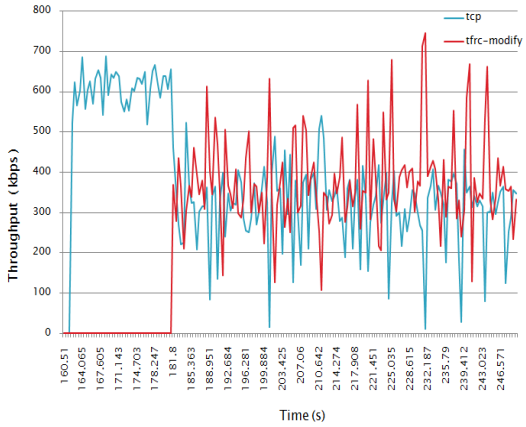
(그림 5) tcp와 tfrc-modify 전송률 비교

(그림 4)은 TCP와 TFRC-modify의 전송률을 비교한 그래프이다. 수정된 TFRC의 전송률이 TCP에 비해 평균적으로 높은 수치를 보인다. TFRC-modify 패킷의 최소 전송률 값을 350kbps 으로 설정해 주었기 때문에 전송률이 그 이하로 내려가지 않는다. 같은 조건에서 TCP와 수정하지 않은 TFRC의 성능을 비교한 (그림 5)의 그래프를 보면 TFRC의 전송률도 TCP와 동등하게 유지되는 모습을 보인다. 이 둘을 비교했을 때 수정된 TFRC-modify의 전송률이 TFRC보다 높게 유지됨을 알 수 있다.

다음으로 시뮬레이션의 환경에서 무선구간에 에러가 있는 경우를 실험했다. TCP와 TFRC 패킷을 발생시키는 노드 2, 3과 노드 1사이에 1%의 에러가 발생하도록 하고 최소 전송률을 설정하지 않은 순수한 TFRC 일때와 최소 전송률을 350kbps 으로 설정한 상태에서 실험해 보았다.



(그림 6) tcp와 tfrc 전송률 비교



(그림 7) tcp와 tfrc-modify 전송률 비교

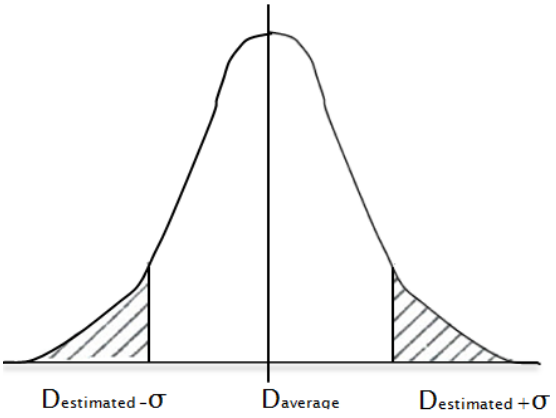
(그림 6)과 (그림 7)은 각각 tcp와 tfrc, tcp와 tfrc-modify의 시뮬레이션에 따른 결과 그래프이다. 그림 6을 보면 에러를 추가하지 않았던 그림 4의 tcp와 tfrc의 전송률 그래프와 비교해 보았을 때 전송률의 변동폭이 급격히 증가함을 보인다. 이는 무선구간에 에러를 추가한 경우 이를 혼잡손실과 구별하지 못하고 전송률을 감소시킨 결과로 알 수 있다.

무선구간에 에러발생률을 1% 추가하고 최소 전송률을 설정한 시뮬레이션의 결과인 그림 7을 보면 이 역시 무선구간의 에러를 혼잡손실과 구별하지 못하고 전송률을 감소시켜 그 변동 폭이 그림 5의 경우보다 크게 나타나지만 최소전송률을 보장해 준 결과, 그림 6과 비교해 보았을 때 tfrc-modify 트래픽의 전송률이 tfrc보다 높은 수치에서 유지됨을 보인다. 하지만 최소전송률을 설정해준 그림 7의 시뮬레이션에서 최소전송률보다 제한한 결과의 전송률이 낮게 나오는 경우가 나타난다. 이는 No Feed Back Timer가 타임아웃 되었을 때 전송률을 절반으로 줄이는 과정에서 혼잡상황이 매우 높음에 따라 제한 전송률이 낮게 나오는 상황이라고 판단한다. 이렇게 혼잡상황이 높은 경우는 응용계층에서 보았을 때 성능이 낮게 나올 수 있다고 예상되었다.

이에 대한 추가적인 향후 방향을 지연시간에 따라 전송률을 제어하는 방법으로 연구하고자 한다. 현재의 알고리즘에 패킷들이 송신되고 수신되었을 때의 지연시간들의 수치로 평균값을 계산한다. 모든 피드백되는 패킷들의 전송지연시간 $D_{estimated}$를 측정하고 $D_{estimated}$값의 평균 $D_{average}$를 유지하고 지연시간을 획득할 때 마다 아래의 공식에 의해 $D_{average}$를 갱신한다. [9]

$$D_{average} = (1 - \alpha) \cdot D_{average} + \alpha \cdot D_{estimated} \quad (1)$$

이 공식에 의해 갱신되는 $D_{average}$값에 따라 갱신 직후 수신 되는 피드백의 지연시간이 그 표준편차에서 상위로 벗어나는 경우와 하위로 벗어나는 경우를 혼잡상황에 따른 전송지연의 변화라고 판단한다.



(그림 8) 피드백 되는 전송지연 시간의 정규분포

그림 8에서 빗금 친 부분이 혼잡이 높은 상황에서 수신된 피드백의 지연시간이라고 할 수 있다. 기존의 전송률 제한 방법에 더불어 빗금 친 부분의 지연시간을 갖는 패킷이 수신되면 혼잡상황으로 판단한다. 이때 수신되는 피드백들에 대한 전송률만을 혼잡손실로 판단하고 무선손실과 구별하게 된다. 이에 대한 최소 전송률만을 보장해 준다면 높은 혼잡상황에서 전송률이 감소하는 폭을 줄여줄 수 있을 것이다.

5. 결론 및 향후 과제

본 논문에서 네트워크에서 패킷의 손실에 따른 전송률 감소를 직접적으로 제한하는 TFRC-modify기법을 제안했다. 이는 직접적으로 최소전송률을 제한하여 일정수치 이하로 내려가지 않게 전송률을 유지해 줌으로써 영상스트리밍 서비스를 지원하는데 그 품질을 향상시킬 수 있다.

향후 과제로는 혼잡상황이 높은 경우에 최소전송률보다 제한한 전송률이 낮게 나오는 상황을 지연시간에 따른 손실 구별 방법을 적용하여 성능을 높이는 방법에 대한 연구가 필요하다.

참고문헌

[1] Kyumin Jeong, Jahon Koo, Kwangsue Chung "Loss Discrimination Mechanism for Improving the Performance of TFRC in Last-hop Wireless Networks" 정보과학회 2010

[2] Hyungho Lee, Chong-ho Choi, "A Loss Discrimination scheme for TFRC in Last Hop wireless Networks" WCNC 2007

[3] S. Floyd, M. Handley, J. Padhye, J. Widmer "TCP Friendly Rate Control (TFRC): Protocol Specification" RFC 5348, 2003

[4] Hyun-Tae Kim, Suk-hun Ji, In-ho Ra "A TCP-Friendly Congestion Control Scheme using Hybrid Approach for Enhancing Fairness of Real-Time Video Stream" Proceedings of KFIS Spring Conference 2004

[5] Jungmin Lee, Sunhun Lee, Kwangsue Chung, "Efficient Video Streaming Based on the TCP-Friendly Rate Control Scheme" 방송공학회논문지, 2005

[6] Dongchil Kim, Jahon Koo, Seungjun Oh, Seungho Yeon, Kwangsue Chung "A Cross - Layer Approach to Improving the Performance of TFRC for Efficient Streaming Service in the Wireless Networks" 한국컴퓨터종합학술대회 2009

[7] Jae young Pyun, "Wireless Measurement based TFRC for QoS Provisioning over IEEE 802.11" 한국통신학회 논문지 2005

[8] 배성수, 한종수, "네트워크 시뮬레이터 NS2 기초와 활용" 세화 2009

[9] Jams F. Kurose, Keith W.Ross "Computer Networking A Top-Down Approch" Addison Wesley 1999

[10] 네트워크 시뮬레이터 NS2
www.isi.edu/nsnam/ns