

무선 환경에서 스트리밍 서비스를 위한 멀티패스 TCP 혼잡 제어

(Multi-path TCP Congestion Control for Streaming Services in Wireless Network)

강 형 규 [†] 레 투 안 [†] 홍 충 선 ^{††}
(Hyeong Kyu Kang) (Le Tuan Anh) (Choong Seon Hong)

요 약 MPTCP(Multipath TCP)는 최근 IETF에 의해 표준화가 완료된 기술이다. MPTCP는 기존 TCP 기반으로 설계되었고, 그 성능은 단일 TCP 중에서 가장 좋은 패스만큼 성능이 나오도록 설계되었다. 하지만 MPTCP는 기존 TCP를 기반으로 확장한 프로토콜이기 때문에 TCP가 갖는 문제를 그대로 가지고 있다. 특히 TCP는 무선환경과 같은 신뢰성이 부족한 네트워크에서 빈번한 혼잡 제어로 그 성능이 떨어진다고 증명된 바 있다. TCP Westwood와 SQRT TCP는 무선 환경에서 스트리밍 전송을 개선하기 위해 제안된 기법이다. 그러나 이들 혼잡제어 기법은 단일패스에서 동작하는 프로토콜이므로 멀티패스 전송 시 다른 프로토콜과의 공평성 문제나 부하를 적절히 분산할 수 없다는 문제점이 있다. 따라서 본 논문에서는 TCP Westwood와 SQRT TCP의 혼잡 제어 기법을 개선하여 멀티패스 전송이 가능한 기법을 제안하였다. 이는 위에서 언급한 공평성 문제나 부하 분산 문제를 해결할 수 있는 방안이며 성능측면에서도 단일 TCP 중에서 가장 좋은 패스만큼 성능이 나오도록 한다. 시뮬레이션 결과는 제안된 기법이 TCP Westwood, SQRT TCP와 비교하여 그 성능이 뛰어난 것을 보였으며, 또한 MPTCP에서 요구하는 세 가지(향상된 처리율, 공평성, 부하 분산) 목표를 달성하고 있음을 보여주었다.

키워드 : 멀티패스 TCP, TCPW, SQRT TCP, 혼잡 제어, 부하 분산, 공평성, 스트리밍 전송

Abstract Multipath TCP (MPTCP) has just been completed the standardization by IETF. MPTCP is designed to be backward compatible with regular TCP, and its performance is equivalent to that of a regular TCP flow on the best path for it. However, MPTCP has performance degradation similar to that the regular TCP does, because MPTCP is extended based on the regular TCP. Especially, the performance of TCP decreases in unreliable wireless network environment. TCP Westwood and SQRT TCP have been proposed for streaming transmission in wireless network. In this paper, we use the characteristics similar to those schemes. However, there are some problems that the TCP Westwood and SQRT TCP cannot fairly use the resources and properly distribute the load to multipath, because these schemes are designed for single-path transmission. Therefore, in this paper, we propose a multipath transmission scheme which improves TCP Westwood and SQRT TCP congestion control algorithms. The proposed scheme demonstrates that its performance is equivalent to that of a regular TCP flow on the best path for it, and is a well-suited method to solve the above mentioned problems. The results from extensive simulations show that the proposed SMPTCP outperforms the existing TCP Westwood and SQRT TCP, and also achieves the goals of MPTCP (i.e., improved throughput, fairness, load balancing).

Key words : Multipath TCP, TCP Westwood, SQRT TCP, Congestion control, Load-balancing, Fairness, Streaming transmission

· 본 연구는 한국방송통신전파진흥원의 차세대통신네트워크 원천기술개발사업 (10913-05004: 미래인터넷에서의 이동환경 및 네트워크 다양성 지원구조 연구)의 일환으로 수행되었음

† 학생회원 : 경희대학교 컴퓨터공학과
hkkang@networking.khu.ac.kr
letuanh@networking.khu.ac.kr
경희대학교 컴퓨터공학과 교수

†† 종신회원 : cshong@khu.ac.kr
(Corresponding author임)

논문접수 : 2011년 9월 6일
심사완료 : 2011년 12월 23일

Copyright©2012 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 정보통신 제39권 제2호(2012.4)

1. 서론

TCP가 나온 이래 인터넷에 존재하는 트래픽 대부분은 TCP 프로토콜이 차지해왔다. TCP는 전송 제어 프로토콜로써 흐름제어와 오류제어를 통해 서비스의 신뢰성과 네트워크 적응성을 제공해 준다. 이런 특징적 이유로 TCP는 다양한 연결형 서비스에 사용되어 왔다. 특히 최근에는 TCP가 고속 망에서 적은 지연(delay)과 높은 처리율(throughput)을 보일 뿐만 아니라 혼잡한 네트워크에서도 데이터 흐름을 조절해 끊김없는 서비스가 가능하다는 이유로 유튜브나 스카이프와 같은 스트리밍 서비스에서도 TCP를 사용하고 있다.

이렇듯 다양한 서비스에 TCP가 쓰이고 있으나 여전히 TCP에는 개선의 여지가 있다. 특히 TCP는 무선환경과 같이 신뢰성이 부족한 네트워크에서는 스트리밍 전송이 어렵다고 알려져 있다. 이는 TCP의 혼잡 회피 [1] 동작으로 전송률을 필요 이상으로 급격히 줄이거나 스트리밍 전송에 적합하지 않게 Cwnd(Congestion window)를 급변하게 조절하기 때문이다. 그리고 가장 큰 이유는 단일 TCP 패스가 혼잡 네트워크를 이용하고 있을 때 그 혼잡 네트워크를 피하지 못하기 때문이다.

따라서 본 논문에서는 이런 배경으로부터 무선 환경에서 TCP기반의 스트리밍 전송에 적합한 멀티패스 혼잡 제어 기법 SMPTCP(Streaming Multi-path TCP)을 제안하였다. SMPTCP기법에서는 TCPW(TCP Westwood)[2], SQRT TCP[3] 혼잡 제어 기법을 활용하여 스트리밍 전송에 적합하도록 Cwnd를 적절히 조절하며, 또한 TCP 멀티패스에 적용 가능하도록 이를 확장함으로써 혼잡 경로에 따른 부하 분산이 가능하도록 하였다.

2. 관련연구

2.1 MPTCP

MPTCP[4]는 최근에 IETF에 의해 표준화가 완료된 멀티패스 기술이다. 이는 단말간에 다중 네트워크 주소를 사용하여 여러 개의 sub-flow를 동시에 구성함으로써 멀티패스 TCP 세션을 구성할 수 있도록 하는 기술이다. 사용자는 이를 이용하여 다중 TCP연결을 구성할 수 있으며 이를 통해 TCP의 성능 향상을 기대할 수 있다.

MPTCP의 특징으로는 기존 TCP를 기반으로 구성되었기 때문에 기본적으로 TCP를 하위 지원하며, 그 성능은 단일 TCP중 가장 좋은 TCP 만큼의 성능이 발휘될 수 있도록 설계되었다. 또한 coordinated 혼잡 제어를 사용하여 다른 단일 링크와 자원을 공평히 사용할 수 있도록 하는 Resource pooling 원리[5]를 사용했다는 것이 특징이라 할 수 있다.

2.2 TCP Westwood

TCPW는 네트워크의 대역폭을 측정하고 측정된 결과

값을 이용하여 혼잡 제어를 수행하는 기법이다. TCPW의 특징은 무선 네트워크의 특성중의 하나인 사용자의 이동성 지원에 영향을 받지 않을 뿐만 아니라 네트워크 계층의 정보 없이도 네트워크 대역폭을 측정할 수 있다는 것이다. 실제로 현재의 네트워크는 사용자에게 아무런 정보를 주지 않는다. 따라서 TCPW는 TCP의 동작으로부터 대역폭을 측정하게 되는데, 이를 위해 TCPW에서는 ACK 패킷간의 도착시간 간격으로부터 대역폭을 측정하는 기법을 이용한다. 이는 네트워크에 추가적인 부하를 주지 않음으로써 평가할 수 있다는 점에서 큰 장점이라 할 수 있다.

$$b_k = d_k / (t_k - t_{k-1})$$

$$\hat{b}_k = \frac{2\tau - \Delta k}{2\tau + \Delta k} \hat{b}_{k-1} + \frac{b_k + b_{k-1}}{2\tau + \Delta k} \Delta k$$

TCPW는 가용 대역폭 측정을 위하여 ACK를 매번 수신할 때마다 \hat{b}_k 를 통해 가용 대역폭을 측정한다. 이를 구하기 위해서는 우선적으로 샘플 대역폭(b_k)을 구해야하는데 b_k 는 k번째 수신된 패킷 d_k 을 이전에 도착한 패킷과의 시간차($t_k - t_{k-1}$)로 나눔으로써 샘플 대역폭을 뽑아내게 된다. 그리고 이 샘플 대역폭을 \hat{b}_k 에 적용시킴으로써 측정된 가용 대역폭을 구하게 된다. 여기서 Δk 는 ACK 패킷간의 inter-arrival 시간을 나타낸다. 그리고 이 평가된 대역폭은 중복 ACK 수신이나 타임아웃 이벤트발생시 Cwnd를 조절(¶)에 이용되게 된다. $w^{TCPW}(t)$ 는 시간 t에서의 Cwnd의 크기를 나타내며 RTTmin는 TCP 연결동안 최소 RTT(Round Trip Time)를 나타낸다.

$$w^{TCPW}(t) \leftarrow \hat{b} \times RTT_{min} / MSS \quad (\text{¶})$$

2.3 SQRT TCP

TCPW는 혼잡 발생시 Cwnd를 맹목적으로 반으로 줄이지 않고 측정된 대역폭에 따라 조절한다는 점에서 기존 TCP 혼잡 제어 기법보다 성능이 좋다고 할 수 있다. 하지만 TCPW는 TCP에서와 마찬가지로 증가단계에서는 AI(Additive Increase)기법을 똑같이 사용하기 때문에 계속적으로 Cwnd를 증가하게 되면 혼잡 회피 동작이 자주 발생하게 되는 문제가 있다. 이런 문제는 빈번하게 전송률을 변경하게 되고 스트리밍 전송에서 이런 문제는 전송 품질을 떨어뜨리는 요소가 될 수 있다. SQRT TCP는 이렇게 빈번하게 변하는 Cwnd를(전송률=Cwnd/RTT) 보완하기 위해 Cwnd를 완만하게 조절하는 기법을 소개한다.

SQRT TCP는 멀티미디어 스트림의 전송률 요구를 Cwnd의 증가의 경우 반비례하게 비율적으로 증가시키고, 감소의 경우 Cwnd의 제곱근에 따라 비율적으로 감소시킴으로써 Cwnd를 완만하게 조절한다. 즉 이를 통해 스트리밍 전송에서 혼잡 회피 동작을 줄임으로써 서비스의

품질을 향상시킬 수 있게 된다. SQRT TCP의 증가와 감소는 ACK의 수신에 따라 이루어지는데 다음과 같이 설명할 수 있다: 정상적 ACK 수신에 경우 Cwnd는 $\alpha/(w^{Bi})^{(k+1)}$ 에 따라 증가하며, 중복 ACK 수신에 따라 혼잡 회피 이벤트 발생시 $\beta(w^{Bi})^l$ 에 의해 값이 설정된다. 증가:

$$w^{Bi}(t) \leftarrow w^{Bi}(t-1) + \frac{1}{(w^{Bi}(t-1))^{(k+1)}} \quad (\S)$$

감소:

$$w^{Bi}(t) \leftarrow w^{Bi}(t-1) + (w^{Bi}(t-1))^l$$

위 증가/감소 식에서 Bi는 SQRT의 binomial 혼잡 프로토콜을 나타내며, $w^{Bi}(t)$ 는 시간 t에서 Cwnd의 크기를 나타낸다. 그리고 k와 l은 혼잡 제어에 따른 고정 상수 값이다. K=0, l=1일 경우 TCP Reno의 혼잡 제어를 나타내며, k=1/2이고 l=1/2일 경우 SQRT TCP를 나타낸다.

3. TCP Westwood와 SQRT TCP를 이용한 혼잡 제어

본 장에서는 SMPTCP를 소개하기에 앞서 TCP Westwood와 SQRT TCP의 혼잡 제어에 대해 보인다. SMPTCP는 TCPW와 SQRT TCP를 이용하여 Cwnd를 조절한다. 여기서 SMPTCP는 Cwnd의 증가의 경우 SQRT TCP에 따라 Cwnd를 완만하게 증가시키며, 감소의 경우 TCPW에 따라 가용 대역폭을 최대로 사용할 수 있도록 한다. 다음은 이를 표현한 식을 나타내며 각각 (§)와 (¶)으로부터 유도된다.

증가:

$$w^{SMPTCP}(t) \leftarrow w^{SMPTCP}(t-1) + \frac{1}{(w^{SMPTCP}(t-1))^{3/2}}$$

감소:

$$w^{SMPTCP}(t) \leftarrow \hat{B} \times RTT_{min,r} \left(\hat{B} = \frac{\hat{b}}{MSS} \right)$$

그리고 이로부터 SMPTCP의 지속적인 시간 t에 대한 증가와 감소의 유체모델(fluid model)을 다음과 같이 나타낼 수 있다(\hat{B} 를 대략 w^{SMPTCP}/RTT 로 가정[6]).

$$\frac{d}{dt} w^{SMPTCP}(t) = \frac{w^{SMPTCP}(t)}{RTT} \left(\frac{1-p}{(w^{SMPTCP}(t))^{3/2}} - \frac{q}{RTT} w^{SMPTCP}(t)p \right) \quad (1)$$

q는 $RTT - RTT_{min}$ 로써 큐잉 딜레이를 나타내며, p는 패킷 손실 확률을 나타낸다. 그리고 본 논문에서 패킷 손실 확률 p는 매우 작다고 가정한다($1-p \approx 1$). 다음은 위로부터 정의되는 고정 소수점 식이다.

$$\frac{1}{(w^{SMPTCP})^{3/2}} = \frac{q}{RTT} \hat{w}^{SMPTCP} \quad \text{이며}$$

$$\text{그리고 } p = \frac{RTT}{q(\hat{w}^{SMPTCP})^{5/2}} \quad (2)$$

(2)에서 \hat{w}^{SMPTCP} 값은 $w^{SMPTCP}(t)$ 값의 평행 값을 나타낸다(증가율 $\frac{d}{dt} w^{SMPTCP}(t) = 0$ 인 경우).

4. SMPTCP의 멀티패스 지원을 위한 혼잡 제어

본 장에서는 무선 환경에서 스트리밍 서비스를 효과적으로 지원하기 위해 위 유체모델로부터 멀티패스에 적용될 수 있도록 확장된 SMPTCP(Streaming Multi-path TCP) 혼잡 제어 기법에 대해 보인다. SMPTCP의 설계 목적은 TCPW와 SQRT TCP를 이용하여 하나의 멀티패스 세션으로부터 스트리밍 데이터를 다중 링크를 통해 효과적으로 분산할 수 있게 하는 것이다 [7]. 그리고 추가적으로 MPTCP의 설계 목적인 처리율 향상, 공평성[4,8]을 함께 고려하여 설계하는 것이다.

지금부터 경로(Path) r 상에서 증가 함수(Additive Increase)에 대한 SMPTCP의 Cwnd 증가 방법에 대해 보인다. 정상적인 ACK를 수신하여 Cwnd를 증가할 경우, Cwnd(w_r)는 $\min\left(\frac{\delta}{w_r(t)^{3/2}}, \frac{1}{w_r(t)^{3/2}}\right)$ 에 의해 증가한다. 그리고 중복 ACK로 인해 감소할 경우, w_s 는 $\hat{B} \times RTT_{min,r}$ 에 의해 값이 설정된다. 즉 이로부터 (3)와 같이 윈도우 크기 변화에 대한 유체모델을 정의할 수 있다.

$$\frac{d}{dt} w_r(t) = \frac{w_r(t)}{RTT_r} \left[\min\left(\frac{\gamma}{w_r(t)^{3/2}}, \frac{1}{w_r(t)^{3/2}}\right) - \frac{q_r}{RTT_r} w_r(t)p_r \right] \quad (3)$$

여기서 $q_r = RTT_r - RTT_{min,r}$ 나타내며 γ 는 하나의 멀티패스에서 링크간의 공유하는 연결 변수를 나타낸다. min() 함수는 멀티패스의 개별 링크의 증가(min(왼쪽))가 단일 링크의 증가(min(오른쪽))보다 클 수 없다는 것을 나타낸다. 다음 식은 (2)와 유사한 방식으로 구해진 (3)의 고정 소수점 식을 나타낸다.

$$\min\left(\frac{\gamma}{\hat{w}_r^{3/2}}, \frac{1}{\hat{w}_r^{3/2}}\right) = \frac{q_r}{RTT_r} \hat{w}_r \hat{p}_r \quad (4)$$

만약 SMPTCP가 경로 r 상에 있다면, (2)로부터 그 경로의 패킷 손실률은 $p_r = RTT_r / \{q_r (\hat{w}_r^{SMPTCP})^{5/2}\}$ 라 할 수 있다. 그러면 p_r 를 (4)에 적용하게 되면 다음과 같은 식을 얻을 수 있다.

$$\hat{w}_r^{SMPTCP} = \max\left(\frac{\hat{w}_r}{\sqrt[5]{2/5}}, \hat{w}_r\right) \quad (5)$$

SMPTCP는 향상된 처리율과 단일 패스 전송과의 공평성을 가져야 한다. 따라서 SMPTCP에서는 전체 경로의 처리율이 가장 좋은 단일 패스의 처리율과 동등하도록 설정되어야 한다[9].

$$\sum_r \frac{\hat{w}_r}{RTT_r} = \max_r \left\{ \frac{\hat{w}_r^{SMPTCP}}{RTT_r} \right\} \quad (6)$$

다음으로 (6)에 (5)을 적용시킴으로써 다음과 같은 식을 얻을 수 있다.

$$\sum_r \frac{\hat{w}_r}{RTT_r} = \max_r \left\{ \max \left(\frac{\hat{w}_r}{\gamma^{2/5} RTT_r}, \frac{\hat{w}_r}{RTT_r} \right) \right\} \quad (7)$$

실제로 (7)에서 \hat{w}_r 는 절대 0이 될 수 없으므로 다음과 같이 식 변형이 가능하다($0 < \gamma < 1$).

$$\sum_r \frac{\hat{w}_r}{RTT_r} = \max_r \left\{ \frac{\hat{w}_r}{\gamma^{2/5} RTT_r} \right\} \quad (8)$$

그 결과 γ 는 다음과 같이 나타낼 수 있다.

$$\gamma = \frac{\max_r \{ (\hat{w}_r / RTT_r)^{5/2} \}}{(\sum_r \hat{w}_r / RTT_r)^{5/2}} \quad (9)$$

위 식에서 γ 는 멀티패스 경로들 사이에서 최대 처리율과 관련된 연결 변수로 설정된다. 그러나 이 경우 같은 환경의 경로 두 개가 있다고 가정하면 혼잡한 경로의 Cwnd는 가법계 부하가 실린 경로의 Cwnd보다 빈번하게 작아지게 된다. 따라서 부하의 적절한 분산을 위해서는 혼잡한 경로의 Cwnd가 다른 좋은 경로보다 서서히 증가될 필요가 있다[8]. 이는 혼잡한 경로의 γ 는 다른 경로의 γ 보다 더 작아야 함을 의미한다. 따라서 다음과 같이 식을 약간 변경하여 나타낼 수 있다.

$$\gamma_r = \hat{w}_r^{5/4} \frac{\max_r \{ \hat{w}_r^{5/4} / RTT_r^{5/2} \}}{(\sum_r \hat{w}_r / RTT_r)^{5/2}} \quad (10)$$

5. 시뮬레이션 및 성능 평가

본 시뮬레이션에서는 SMPTCP가 무선환경에서 스트리밍 서비스 전송 시 얼마나 안전하고 공평하게 부하를 적절히 분산하는지를 평가한다. 이를 위해 본 논문에서는 NS-2[10]를 이용하여 시뮬레이션 환경을 구성하고 평가하였다. 본 시뮬레이션 환경에서 멀티패스 전송 프로토콜은 SACK[11] 옵션을 사용하였으며, 1000-byte의 데이터 패킷을 사용하여 (1),(3)에서 보인 것을 기반으로 시나리오 환경을 구성하였다. 그림 1은 이런 환경으로부터 구성된 시나리오이다. 각 링크는 5Mbps의 대역

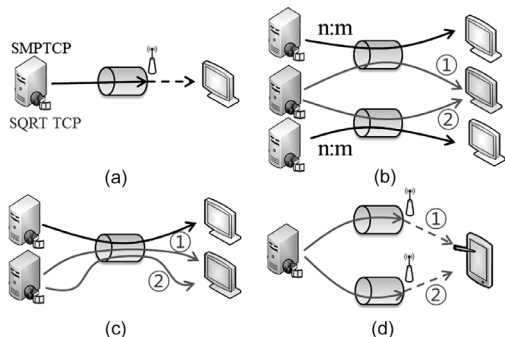


그림 1 멀티패스 시뮬레이션 시나리오

폭과, 60ms의 지연을 가진다. 그리고 (a)(c)의 경우 유선은 10Mbps, 58ms, 무선은 5Mbps, 2ms로 구성된다.

5.1 무선 환경에서 단일패스 SMPTCP 성능 비교

이번 절에서는 무선 환경에서의 SMPTCP와 TCPW, SQRTP TCP의 성능을 비교하였다. 그림 1(a)는 시뮬레이션에 대한 시나리오를 나타낸다. SQRTP TCP는 스트리밍 서비스를 위해 완전한 혼잡 제어를 사용한다. 하지만 혼잡 회피 동작 시 필요 이상으로 Cwnd를 줄이는 문제가 있다. 또한 TCPW는 손실이 많은 네트워크에서 그 성능이 급격히 떨어지는 문제가 있다. 따라서 본 논문에서는 SQRTP TCP의 완전한 혼잡 제어 기법과 TCPW의 대역폭 측정 기법을 통해 가용 용량에 맞게 Cwnd를 줄이는 SMPTCP 혼잡 제어를 제안하였다. 그림 2는 이렇게 만들어진 SMPTCP 혼잡 제어와 기존 SQRTP TCP의 혼잡 제어의 성능을 비교한 것으로 시뮬레이션 환경은 무선환경에서 패킷 손실률 0.01에서부터 5%까지 600초간 측정할 수 있도록 구성하였다. 그림 2에서 볼 수 있듯이 패킷 손실률 0.01에서 1%까지 SMPTCP가 TCPW와 SQRTP TCP 보다 대략적으로 증가된 성능을 보여주었다.

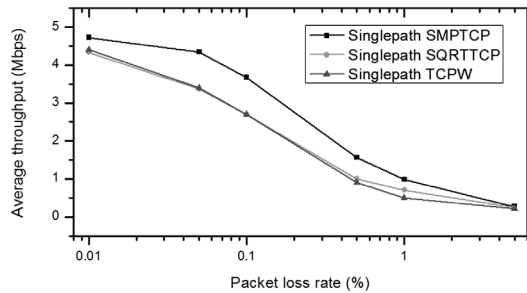


그림 2 단일패스 성능 비교

5.2 멀티패스 환경에서의 SMPTCP 평가

5.2.1 SMPTCP의 혼잡 윈도우 변화 정도 평가

두 개의 멀티패스가 있다고 가정하고 그 패스 상에 똑 같은 부하를 싣는다고 가정하자. 그렇다면 패킷이 각 패스로 전송되었을 때 멀티패스 프로토콜은 안전하게 동작할까[7-9]? 실제로 멀티패스 프로토콜은 부하를 똑같이 분산한다고 해도 안전하지 않다. 왜냐하면 TCP 패스들은 완전히 동시에 사용될 수 없기 때문이다. 또한 각 패스의 환경의 차이는 더욱더 프로토콜을 안전하지 않게 만들 수 있다.

이번 절에서는 본 논문의 멀티패스 혼잡 제어 기법의 Cwnd 변화 정도를 통해 얼마나 프로토콜이 안전한지를 살펴본다. 이를 위해 그림 1(b)와 같이 환경을 구축하였으며 각 링크에는 10개의 백그라운드 부하를 생성하는

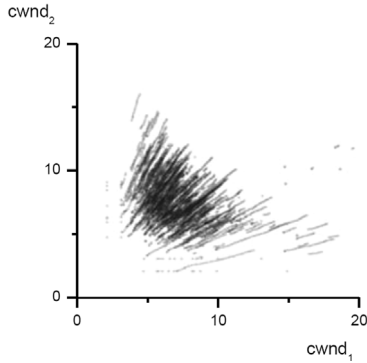


그림 3 Two-패스 SMPTCP의 혼잡 윈도우 변화 정도

TCP 연결이 구성된다. 두개의 패스에 동일한 크기의 부하가 가해지기 때문에 각 패스에는 동일한 패킷 손실이 일어날 것이라는 것을 경험적으로 알 수 있다. 그림 3은 위와 같은 환경에서 각 패스의 Cwnd의 변화 정도를 보여준다. 그림에서 볼 수 있듯이 SMPTCP는 두 패스의 Cwnd(cwnd₁, cwnd₂)의 증가 정도가 모든 방면으로 적절히 분포된 것을 알 수 있다. 따라서 본 논문의 기법이 안전하다고 말할 수 있다.

5.2.2 SMPTCP 공평성 평가

이번 절에서는 SMPTCP가 단일 패스(SQRT TCP)와 얼마나 공평하게 자원을 공유하는지를 살펴보도록 하겠다. 그림 1(c)는 본 실험을 위해 구성된 시나리오이다. 그림에서 볼 수 있듯이 단일패스 기기와 멀티패스 기기가 하나의 링크를 공유하고 있다. 이런 상황에서 우리는 Cwnd의 크기 변화를 통해 두 프로토콜이 얼마나 공평하게 자원을 나눠 쓰는지 살펴보았다. 그림 4는 SMPTCP와 SQRT TCP와의 Cwnd 크기 변화를 나타낸 것이다. 그림에서 볼 수 있듯이 SQRT TCP는 40pkts 정도를 유지하고 있으며, SMPTCP는 각각 패스당 20pkts 정도를 유지하고 있다. 따라서 SMPTCP는 자원을 공평하게 공유한다고 할 수 있다.

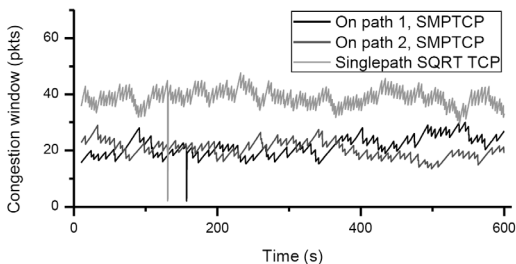


그림 4 SMPTCP와 단일 패스와의 Cwnd 크기 변화

5.2.3 무선 환경에서의 성능 평가

이번 절에서는 무선 환경에서의 멀티패스 SMPTCP

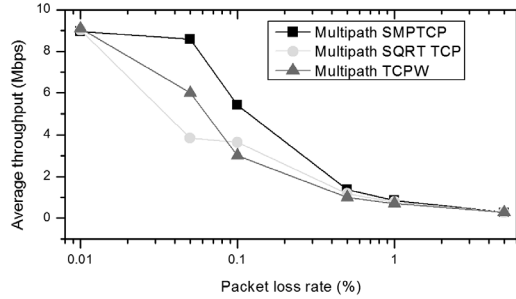


그림 5 멀티패스 성능 비교

와 SQRT TCP의 성능을 비교하였다. 그림 1(d)는 시뮬레이션에 대한 시나리오를 나타낸다. 그림 5는 이렇게 만들어진 SMPTCP 혼잡 제어 기법과 기존 TCPW와 SQRT TCP의 혼잡 제어 기법의 성능을 비교한 것으로 시뮬레이션 환경은 무선환경에서 패킷 손실률 0.01에서 5%까지 600초간 측정하였다. 그림에서 볼 수 있듯이 패킷 손실률 0.01에서 1%까지 SMPTCP가 TCPW와 SQRT TCP 보다 상당히 뛰어난 성능을 보였다.

5.2.4 부하 분산 정도 평가

스트리밍 전송에서 부하 분산은 매우 중요하다. 만약 부하가 심한 링크를 통해 스트리밍을 전송하게 된다면 서비스의 품질이 떨어지기 때문이다. 본 논문은 이를 극복하기 위해 멀티패스를 통해 부하를 혼합한 망에서 혼합하지 않은 망으로 옮김으로써 그 성능이 단일 패스 중 가장 좋은 링크의 성능이 나오도록 하였다.

실험 환경은 그림 1(b) 시나리오와 같이 구성하였다. 가용 용량과 전송 지연을 링크1(10Mbps, 40ms), 링크 2(6Mbps, 160ms)로 설정하였으며, 링크의 부하를 다르게 하기 위해 첫 번째 링크에는 8개의 단일 SMPTCP 흐름을 설정하고 두 번째 링크에는 3개의 단일 SMPTCP 흐름을 설정하였다.

그림 6은 SMPTCP의 부하 분산 정도를 측정한 결과를 나타낸다. 그림에서 볼 수 있듯이 첫 번째 경로(On link 1)에 대한 전송률이 매우 낮게 나와 있다. 이는 첫

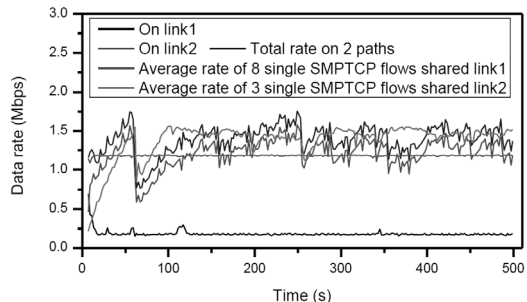


그림 6 SMPTCP의 부하 분산 평가

번째 링크가 매우 혼잡한 상황에 있기 때문이다. 하지만 두 번째 경로(On link 2)로 부하가 옮겨짐으로써 기대치 만큼(단일패스중 가장 좋은 링크의 성능)의 성능이 나오는 것을 확인할 수 있었다.

6. 결론

본 논문에서는 무선 환경에서 스트리밍 전송을 위한 멀티패스 TCP 혼잡 제어 기법에 대해서 소개하였다. TCP는 혼잡 제어 메커니즘에 따라 전송률을 변화시키기 때문에 스트리밍에 적합하지 않다. 즉, TCP 기반의 스트리밍 전송을 위해서는 스트리밍 전송에 적합한 혼잡 제어 스킴의 선택과 적절한 부하 분산을 통해 혼잡을 줄이는 것이 중요하다.

이를 해결하기 위해 본 논문에서는 TCPW, SQRT TCP, 멀티패스 기법을 소개하고 각 기법의 유체모델(fluid model)를 설정하여 각 특성을 결합함으로써 스트리밍 전송에 적합한 혼잡 제어 기법을 제안하였다. 특히 본 기법에서는 MPTCP의 설계 목적인 처리율 향상, 공평성, 부하 분산을 함께 고려함으로써 기법의 완성도를 높였다. 성능평가에서는 성능 평가를 위해 NS-2을 이용하였으며, 평가는 혼잡 윈도우 변화 정도, 공평성, 무선 환경에서의 성능, 부하 분산 정도에서 싱글/멀티패스 SQRT TCP와 SMPTCP를 서로 비교함으로써 그 성능을 입증할 수 있었다.

참고 문헌

- [1] M. Allman and V. Paxson, "TCP Congestion Control," *ITEF RFC 5681*, Sep. 2009.
- [2] Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi and Ren Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," *ACM SIGMOBILE*, 2001.
- [3] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," *IEEE INFOCOM*, 2001.
- [4] C. Raiciu, M. Handley and D. Wischik, "Coupled multipath-aware congestion control," *IETF Internet Draft*, Jul. 2010.
- [5] D. Wischik, M. Handley and M. B. Braun, "The resource pooling principle," *ACM SIGCOMM CCR*, Oct. 2008.
- [6] L. A. Grieco and S. Mascolo, "Mathematical analysis of Westwood + TCP congestion control," *IEEE PROCEEDINGS-CONTROL THEORY AND APPLICATIONS*.
- [7] D. Wischik, C. Raiciu and M. Handley, "Balancing Resource Pooling and Equipoise in Multipath Transport," *ACM SIGCOMM*, 2010.
- [8] D. Wischik, C. Raiciu, A. Greenhalgh and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," *Proc. Of*

the USENIX NSDI'11 Conf., Mar. 2011.

- [9] D. Wischik, M. Handley and C. Raiciu, "Control of multipath TCP and optimization of multipath routing in the Internet," *Network Control and Optimization*, 2009.
- [10] NS-2 network simulator, [Online] <http://www.isi.edu/nsnam/ns/>
- [11] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, "TCP selective acknowledgment options(SACK)," *IETF RFC 2018*, Oct. 1996.



강 형 규

2010년 2월 제주대학교 컴퓨터공학과(공학사). 2010년 2월~현재 경희대학교 컴퓨터공학과 석사과정. 관심분야는 멀티패스 혼잡제어, 멀티패스 패킷 스케줄링, 분산처리시스템, 웹



Le Tuan Anh

He received the college diploma in Telecommunications from Posts and Telecoms Training Center II, Ho Chi Minh city, Vietnam in 1996, and his B.S. degree in Information Technology from Natural Sciences University, Ho Chi Minh city, Vietnam in 2002. During 1997-2003, He received MS in Computer Engineering from Kyung Hee University, Korea in 2005. He is currently a Ph.D Candidate in the Department of Computer Engineering at Kyung Hee University, Korea. His research interests are traffic engineering, flow control, congestion control, and resource management.



홍 충 선

1983년 경희대학교 전자공학과(공학사) 1985년 경희대학교 전자공학과(공학석사) 1997년 Keio University, Department of Information and Computer Science (공학박사). 1988년~1999년 한국통신통신망연구소 수석연구원/네트워킹 연구실장. 1999년~현재 경희대학교 컴퓨터공학과 교수. 관심분야는 인터넷 서비스 및 망 관리구조, 미래인터넷, IP mobility, Sensor Networks, Network Security