

# FQ-CoDel 병목 버퍼를 공유하는 MPTCP와 단일 경로 TCP 간의 공정성 평가

조효섭, 조유제

경북대학교 전자전기공학부

{yg01175, yzcho}@knu.ac.kr

## Fairness Evaluation between MPTCP and Single-Path TCP Sharing FQ-CoDel Bottleneck Buffers

Hyo-Sup Cho\*, You-Ze Cho

School of Electronic and Electrical Engineering, Kyungpook National University

### 요약

본 논문은 공유 병목 링크에서 대기열 관리 기법에 따른 MPTCP와 단일 경로 TCP 간의 공정성을 평가 실험을 진행하였다. 기본적인 대기열 기법인 Tail-Drop을 적용한 경우 공유 병목 링크를 함께 지나는 MPTCP와 TCP가 공정하게 대역폭을 차지하였다. 하지만 FQ-CoDel 기법의 경우 MPTCP가 TCP 보다 두 배 많은 대역폭을 차지하여 불공정하게 병목 링크를 공유하는 문제를 확인했다.

### I. 서론

Multipath TCP (MPTCP)는 IETF의 RFC 6824에서 표준화된 전송 계층 프로토콜로 다수의 경로를 통해 동시에 데이터를 전송함으로써 처리량을 향상시킬 수 있다. MPTCP의 설계 및 배치를 위해서는 최상의 경로를 사용하고 있는 TCP보다 높은 처리율을 가져야 하며, 단일 경로 TCP 플로우와 동일한 병목 링크를 공유하는 경우 TCP 플로우와 대역폭을 공정하게 사용하는 조건이 충족되어야 한다 [1].

MPTCP와 단일 경로 TCP의 공동 병목 링크에서의 공정성은 각 플로우의 중단 간 왕복 지연시간, 혼잡제어 알고리즘, 병목 버퍼의 대기열 관리 기법과 같은 다양한 요소에 의해 결정된다. 현재까지는, MPTCP와 TCP의 여러 혼잡제어 종류에 따라서 공정성을 평가하거나 새로운 혼잡제어 알고리즘을 제안하는 연구가 주로 소개되었다 [2]. 하지만, 공유 병목 버퍼에 적용된 대기열 관리 기법에 따른 MPTCP와 TCP의 공정성 평가에 관한 연구는 소개되지 않았다.

따라서, 본 논문에서는 공유 병목 버퍼에 적용된 대기열 관리 기법에 따른 MPTCP와 단일 경로 TCP의 공정성 평가 실험을 진행하였다. 그림 1과 같이 기본적인 대기열 관리 기법인 Tail-Drop과 Active Queue Management (AQM) 기법의 하나인 Flow Queue-CoDel (FQ-CoDel) [3]을 공유 병목 버퍼에 적용했을 때 MPTCP와 단일 경로 TCP가 서로 공정성을 유지하는지 확인하였다. FQ-CoDel은 버퍼블로트 문제 [4]를 해결하기 위해 고안된 Controlled Delay (CoDel) 기법 [5]을 기반으로 제안되었으며, 단일 큐를 사용하는 CoDel과 달리 모든 플로우에 서브 큐를 할당하여 TCP 플로우 간의 공정성을 향상시킨 기법이다.

### II. 본론

그림 2와 같이, 두 개의 서브플로우를 가지는 MPTCP (SF1 및 SF2)와 단일 경로 TCP가 같은 병목 링크를 지날 때, 병목 버퍼에 적용된 대기열 관리 기법에 따른 MPTCP와 TCP의 성능을 비교하였다. 미니넷 에뮬레이터 [6]를 이용하여 실험환경을 구성하였으며, 실험 파라미터들은 아래와 같다.

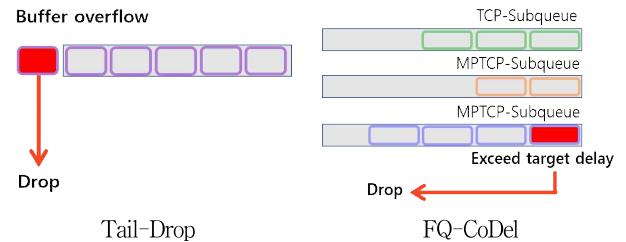


그림 1. 대기열 관리 기법 구동방식

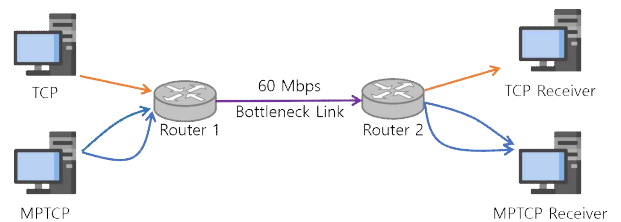
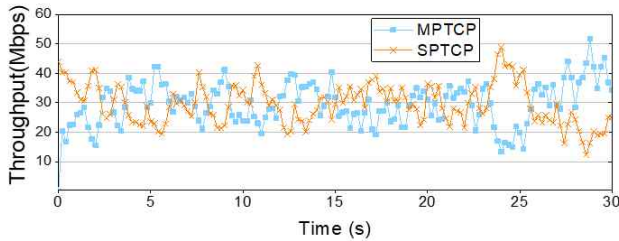


그림 2. 공유 병목 링크 실험환경

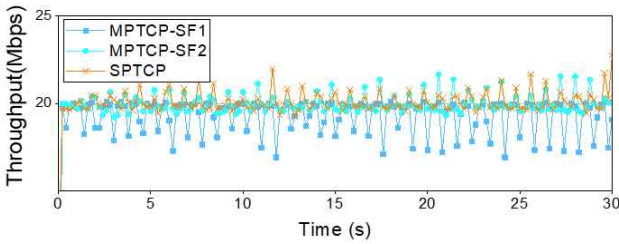
- 공유 병목 버퍼 크기: 1BDP
- 공유 병목 링크의 대역폭: 60Mbps
- 중단 간 왕복 지연시간: 10ms
- 혼잡제어 알고리즘: LIA (MPTCP) [7], Reno (TCP)
- 대기열 관리 기법: Tail-Drop, FQ-CoDel (default setting)

그림 3은 공유 병목 링크를 지나는 MPTCP 서브플로우와 단일 경로 TCP 플로우의 처리량 변화를 나타내며, 동일한 실험을 10회 반복하여 얻은 평균 처리량 비를 그림 4를 통해 나타내었다.

Tail-Drop 기법을 적용한 공유 병목 링크에서 MPTCP SF1과 SF2의 합계 처리량과 TCP 플로우의 처리량이 약 30Mbps로 거의 유사한 평균 처리량을 가지는 것을 그림 3(a)로부터 확인할 수 있다. 하지만 그림 3(b)에서는 병목 버퍼에 적용된 FQ-CoDel 기법에 따라 MPTCP의 SF1과 SF2 그리고 TCP 플로우에 각각의 독립적인 서브 큐를 생성하게 되어



(a) Tail-Drop



(b) FQ-CoDel

그림 3. MPTCP 서브플로우와 TCP 플로우의 시간에 따른 처리량 변화

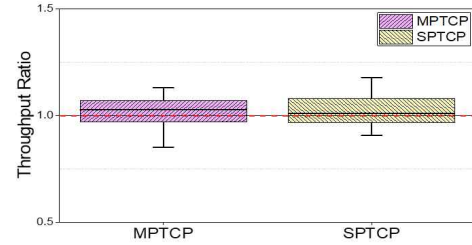
MPTCP 서브플로우와 TCP 플로우 모두 19~20Mbps 사이의 동일한 처리량을 가지는 것을 확인할 수 있다. 따라서, MPTCP는 TCP보다 2배 더 많은 대역폭을 차지하게 된다.

MPTCP와 단일 경로 TCP 간의 공정성을 시각적으로 비교하기 위해, MPTCP와 TCP의 평균 처리량 비를 그림 4를 통해 나타냈다. 그림 4(a)에서는 Tail-Drop이 적용된 공유 병목 링크에서 MPTCP와 TCP가 대역폭을 공정하게 공유할 때를 나타내는 1에 위치하는 것을 확인할 수 있다. 하지만 그림 4(b)에서 FQ-CoDel을 적용한 병목 링크를 공유하는 MPTCP의 평균 처리량 비는 약 1.4, TCP는 0.7에 가깝게 위치한다. 이는, MPTCP가 TCP를 과도하게 압도하여 더 높은 처리량을 갖는 것을 나타낸다. 따라서 한정된 병목 링크의 대역폭을 MPTCP가 더 많이 차지하여 공정성 문제가 발생하는 것을 알 수 있다.

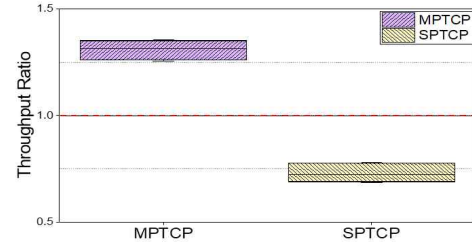
기본적인 대기열 관리 기법인 Tail-Drop을 적용한 병목 링크에서는 MPTCP와 단일 경로 TCP가 한정된 병목 링크 대역폭 60Mbps를 각각 30Mbps씩 공정하게 공유한다. 하지만 Tail-Drop 기법과 다르게 FQ-CoDel 기법이 적용된 공유 병목 링크에서는 MPTCP의 SF1과 SF2가 각각 TCP 플로우와 같은 대역폭인 20Mbps를 차지하기 때문에, MPTCP가 TCP에 비해 두 배 많은 대역폭인 40Mbps를 차지하게 되고 TCP는 20Mbps의 대역폭을 차지하게 된다. 따라서 60Mbps로 한정된 병목 링크 대역폭을 공정하게 공유하지 못하는 문제를 확인했다.

### III. 결론

본 논문에서는 FQ-CoDel 기법을 적용한 병목 버퍼를 MPTCP와 단일 경로 TCP가 함께 지날 때, 병목 링크 대역폭을 공정하게 차지하는지를 평가하였다. 기본 대기열 관리 기법인 Tail-Drop을 적용한 경우와 비교하였으며, Tail-Drop 기법을 적용한 병목 링크에서는 MPTCP와 TCP가 공정하게 대역폭을 공유했다. 하지만 FQ-CoDel 기법을 적용한 병목 링크에서는 MPTCP가 TCP에 비해 두 배 많은 대역폭을 차지하며 불공정하게 병목 링크를 공유하는 것을 확인했다. 추후에는 다양한 AQM 기법들에 대한 종합적인 성능 평가를 수행하고, MPTCP에 가장 적합한 대기열 관리 기법을 제안하고자 한다.



(a) Tail-Drop



(b) FQ-CoDel

그림 4. MPTCP와 TCP의 평균 처리량 비

### ACKNOWLEDGMENT

This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2018R1A6A1A03025109) and by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1A2C1006249).

### 참고 문헌

- [1] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," IETF document RFC 6824, IETF, Jan. 2013.
- [2] S. Ferlin, Ö. Alay, T. Dreiholz, D. A. Hayes, D. A. Hayes, and M. Welzl, "Revisiting congestion control for multipath TCP with shared bottleneck detection," in Proc. IEEE INFOCOM, Apr. 2016.
- [3] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet, "The flowqueue CoDel packet scheduler and activequeue management algorithm," RFC 8290, Jan. 2018.
- [4] Pan, R., Natarajan, P., Baker, F., and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem", RFC 8033, DOI 10.17487/RFC8033, February 2017.
- [5] K. Nichols, V. Jacobson, A. McGregor, Ed., and J. Iyengar, Ed., "Controlled Delay ActiveQueue Management," RFC 8289, Jan. 2018.
- [6] B. Lantz, B. Heller, and N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks," in Proc. ACM SIGCOMM Workshop Hot Topics Netw., Oct. 2010.
- [7] C. Raiciu, M. Handley, and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols," IETF document RFC 6356, IETF, Oct. 2011.