

큰 대역폭 지연 곱 망에서의 TCP 혼잡 제어 알고리즘의 비교에 관한 연구

김신규, 정형수, 한혁, 염현영
서울대학교 컴퓨터공학부

e-mail:{sgkim, jhs, hhyuck, yeom}@dcslab.snu.ac.kr

A Study on Algorithms of TCP Congestion Control in High Bandwidth Delay Product Network

Shin Gyu Kim, Hyungsoo Jung, Hyuck Han, Heon Y. Yeom
Dept of Computer Science and Engineering, Seoul National University

요 약

AIMD(Additive Increase Multiplicative Decrease)를 기본으로 하는 TCP 혼잡 제어 알고리즘은 점점 더 사용처가 늘어가고 있는 큰 대역폭 지연 망에서 높은 사용률을 달성하기 힘들다는 문제점을 지니고 있다. 본고에서는 이 문제를 해결하기 위하여 제시된 다양한 TCP 혼잡 제어 알고리즘을 비교 분석해 보고자 한다.

1. 서론

현재 인터넷에서 사용되고 있는 TCP 혼잡 제어 알고리즘(Congestion Control Algorithm)은 패킷 손실 여부에 따라 망의 혼잡 상황을 추측하여 윈도우의 크기를 조절함으로써 망의 혼잡 상황을 제어하고 있다. 대표적으로 TCP Reno의 경우 낮은 대역폭 지연 곱 망에서는 좋은 성능을 보여주지만 큰 대역폭 지연 곱 망에서는 패킷 손실에 의하여 윈도우의 크기가 줄어든 경우 망의 사용률을 다시 끌어올리는 데 매우 긴 시간이 걸린다는 단점을 지니고 있다.

이러한 문제점을 해결하기 위하여 크게 두 가지 방향으로 많은 연구들이 진행되었다. 하나는 라우터의 도움 없이 종단 간 시스템끼리 혼잡 제어를 하는 방식이고, 다른 하나는 라우터의 도움을 받는 방식이다. 이런 연구들을 통해 더 좋은 성능을 얻을 수 있었지만 아직까지 표준으로 받아들여진 알고리즘은 사실상 없다고 할 수 있다. 따라서 이 논문에서는 현재까지 제시된 다양한 TCP 혼잡 제어 알고리즘이 어떤 장단점을 갖고 있는지 살펴보고, 시뮬레이션을 통하여 큰 대역폭 지연 곱 상황에서 어떤 특징을 보이는지 살펴볼 것이다.

본 논문은 2장에서 각 알고리즘의 특징을 설명하고, 3장에서 시뮬레이션을 통한 결과를 비교 분석한 후 4장에서 이 논문을 마무리 지을 것이다.

2. 큰 대역폭 지연 망을 위한 TCP 혼잡 제어 알고리즘

2.1 종단 간 노드에 의한 제어

대부분의 연구들은 라우터의 도움을 받기보다는 종단

간 노드 사이의 패킷의 손실에 의한 혼잡 제어에 기반을 두고 있다. 이 방식은 규약(Protocol)을 배포할 때 라우터를 수정하지 않아도 된다는 큰 장점을 갖고 있지만 망의 혼잡 상황에 대해서 알 수 있는 정보가 제한적이기 때문에 망의 상황을 패킷의 손실이나 응답 시간의 변화를 통하여 추측하여야만 한다는 약점이 있다.

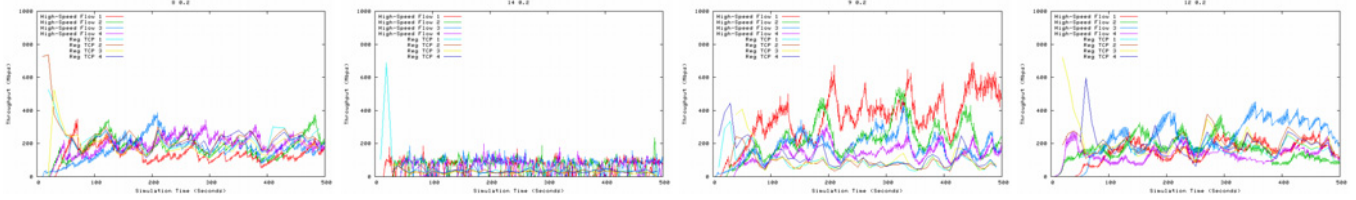
본고에서는 현재까지 제안된 여러 가지 규약들 중에서 HSTCP[1], FAST[2], STCP[3], BIC[4]의 네 가지 규약을 분석해보고자 한다.

2.2 라우터에 의한 제어

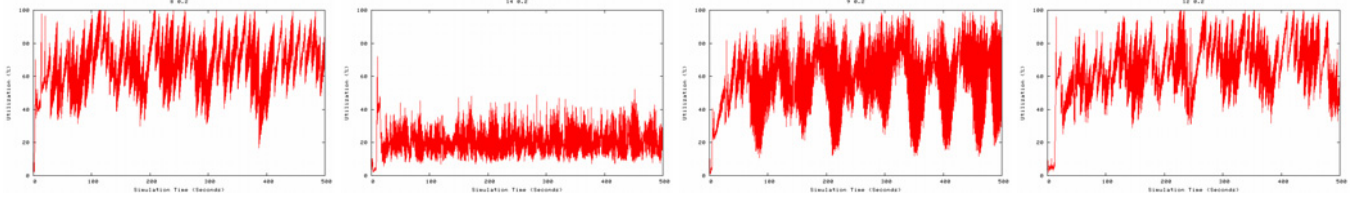
라우터의 도움을 받는 경우에는 망의 혼잡 상황을 정확히 알 수 있기 때문에 보다 세밀하고 정확한 제어가 가능하게 된다. 대표적인 규약인 XCP[5]의 경우 패킷의 손실도 없을 뿐만 아니라, 패킷의 손실이 있다 해도 굉장히 짧은 시간 내에 완벽히 복구가 가능하다. 또한 망의 사용률도 100%에 가까운 수준으로 유지할 수도 있다. 하지만 이를 위해서는 모든 라우터가 XCP라우터로 바뀌어야 한다는 큰 제약이 존재한다. 이런 이유로 높은 성능에도 불구하고 본고에서는 비교의 대상에서 제외하였다.

3. 시뮬레이션을 통한 비교

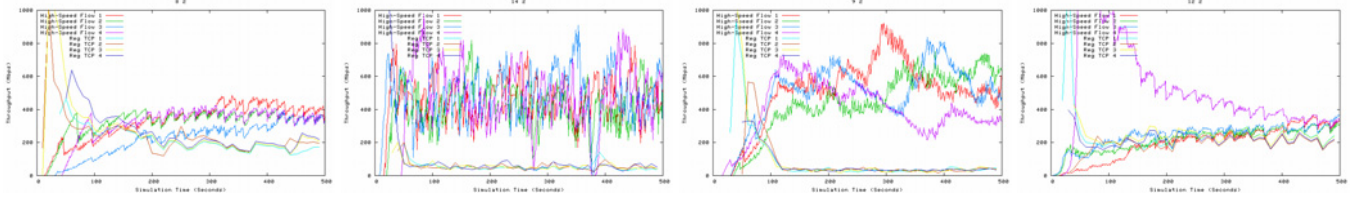
이 장에서는 NS2를 이용하여 큰 대역폭 지연 곱 망에서의 각 알고리즘의 성능을 비교해보도록 하겠다. 시뮬레이션에 사용한 망의 구성은 그림5와 같은 아령 위상(dumbbell topology)이고 병목 구간은 2.5Gbps의 대역폭과 10ms의 지연 시간을 갖도록 하였다. 또한 실제 상황과 비슷하게 하기 위하여 비교 대상이 되는 알고리즘 이외에



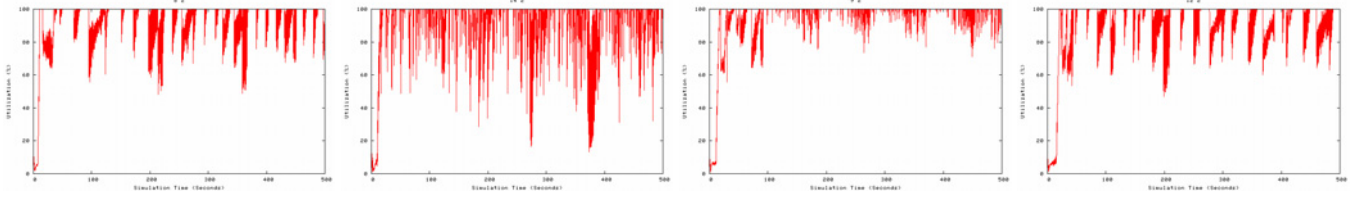
(a) HSTCP (b) FAST (c) STCP (d) BIC
(그림 1) 왕복 시간이 같은 경우 개별 흐름의 사용률



(e) HSTCP (f) FAST (g) STCP (h) BIC
(그림 2) 왕복 시간이 같은 경우 병목 구간에서의 사용률

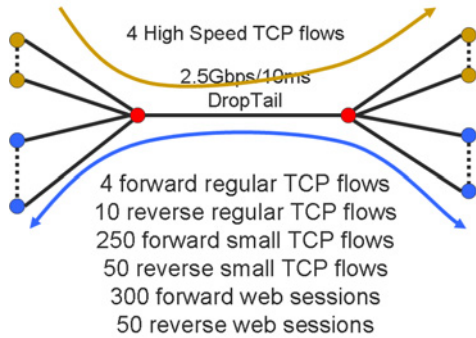


(a) HSTCP (b) FAST (c) STCP (d) BIC
(그림 3) 왕복 시간이 다른 경우 개별 흐름의 사용률



(e) HSTCP (f) FAST (g) STCP (h) BIC
(그림 4) 왕복 시간이 다른 경우 병목 구간에서의 사용률

전송 시간이 짧은 웹 트래픽과 FTP와 같은 전송 시간이 긴 트래픽도 함께 섞어주었다. 총 시뮬레이션 시간은 500 초였고, 비교 대상이 되는 알고리즘은 총 네 개의 흐름(flow)이 10초마다 시작되도록 하였다.



(그림 5) 시뮬레이션에 사용된 망의 구성

3.1 왕복 시간이 같은 경우

왕복 시간(Round Trip Time)은 225ms로 설정하였고, 시뮬레이션의 결과는 그림1과 그림2에 나타내었다.

먼저 그림1을 통하여 개별 흐름별로 각 알고리즘을 비

교해보도록 하겠다. 안정성(stability)의 측면에서 살펴봤을 때 HSTCP의 경우 비교적 안정적인 모습을 보여주고 있지만, STCP와 BIC의 경우 전 구간에서 걸쳐 불안한 모습을 보여주고 있다. FAST의 경우 큰 변화가 없는 모습을 보여주고 있지만 모든 흐름이 매우 낮은 사용률에 머물러 있음을 주목하여야 한다.

그림2를 통하여 병목 구간의 전체 총 사용률을 보면 FAST를 제외하고는 사용률의 변화가 심한 것을 볼 수 있다. 이것은 패킷의 손실에 의한 사용률의 복구가 기존의 TCP 알고리즘보다 빠름을 보여주는 것이다. 하지만 이것도 충분하다고는 할 수 없다. 그래프에서 보듯이 여전이 남아있는 대역폭이 상당한 부분을 차지하고 있기 때문이다. 특히 FAST의 경우에는 전 구간에 걸쳐 50% 미만의 매우 낮은 사용률을 보이고 있음을 알 수 있다.

3.2 왕복 시간이 다른 경우

이번에는 왕복 시간을 각각 50, 100, 150, 200ms 으로 설정하여 서로 다른 성격의 흐름이 섞였을 경우를 비교해보도록 하겠다.

그림3을 통하여 각각의 흐름별로 살펴보면 네 가지 규약 모두 높은 사용률을 보이는 흐름과 낮은 사용률을 보이는 흐름으로 나뉘어져 있는 것을 볼 수 있다. 이런 현상은 FAST와 STCP에서 특히 두드러지게 관찰되고 있다. 이런 특성은 각각의 규약이 실제로 사용되기 매우 어렵다는 것을 뜻한다. 이런 규약의 흐름이 현재의 TCP 규약과 섞이게 된다면 기존 규약을 사용하는 사용자들은 매우 큰 피해를 보게되기 때문이다.

그림4를 통해 병목구간에서의 사용률을 살펴보면 3.1에서와 달리 대부분 100%까지 달성하고 있음을 볼 수 있다. 하지만 이것은 하나의 규약이 다른 흐름을 압도하면서 발생된 현상으로 바람직한 상황이라고 볼 수 없다.

4. 결론

본 논문을 통하여 지금까지 새롭게 제시된 대표적인 TCP규약들을 살펴보았다. 각 규약들은 기존의 TCP가 갖고 있던 약점들을 많이 보완하여 큰 대역폭 지연 곱 망에서도 높은 사용률을 달성하고 있음을 알 수 있었다. 하지만 기존의 TCP규약과 함께 사용할 경우 기존의 TCP 규약을 사용하는 흐름의 경우 다른 TCP 흐름에 의해 낮은 사용률에 머물게 된다는 문제점이 있다는 것을 알 수 있었다.

감사의 글

이 연구는 서울시 산학연 협력 사업에서 부분적으로 지원 받았으며, 서울대학교 컴퓨터 연구소는 이 연구의 시설을 제공하였습니다.

참고문헌

- [1] S. Floyd, "HighSpeed TCP for Large Congestion Windows", IETF, INTERNET DRAFT, draft-floyd-tcp-highspeed-02.txt, 2002
- [2] C. Jin, D. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, H. Newman, F. Paganini, S. Ravot, and S. Singh, "FAST Kernel: Background Theory and Experimental Results", Presented at the First International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2003), February 3-4, 2003, CERN, Geneva, Switzerland
- [3] T. Kelly, "Scalable TCP: Improving Performance in Highspeed Wide Area Networks", Submitted for publication, December 2002
- [4] L. Xu, K. Harfoush, and I. Rhee, "Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks," In Proceedings of IEEE INFOCOM 2004,

March 2004

- [5] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth-Delay Product Networks. SIGCOMM'02, August 2002.