

이종 망에서 MPTCP 성능 향상을 위한 리눅스 기반의 스케줄러 구현 및 성능 평가

안종원 · 김도호 · 김민섭 · 이재용 · 김병철

충남대학교

Implementation and Performance Evaluation of Linux-based Scheduler for improving MPTCP Performance in Heterogeneous Networks

Jong-won Ahn · Do-ho Kim · Min-seob Kim · Jae-yong Lee · Byung-chul Kim

Chungnam National University

E-mail : jongwonahn@cnu.ac.kr

요 약

Multipath TCP (MPTCP)는 다수의 인터페이스를 동시에 사용하여 유한한 네트워크 자원을 최대한 활용하고자 표준화된 전송 계층 프로토콜이다. 각 경로의 특성이 같을 경우 기존 단일 TCP보다 안정성과 대역폭활용 측면에서 장점이 있지만 경로 특성이 상이할 경우 단일 TCP보다 성능이 낮아지는 현상을 볼 수 있다. 이는 여러 가지 복합적인 이유가 있지만 가장 큰 영향을 미치는 문제는 지연시간을 급격히 증가시키는 버퍼블로트를 예로 들 수 있다. 본 논문에서는 지연시간을 급격히 증가 시키는 버퍼블로트로 인한 MPTCP 성능 저하를 개선한 알고리즘을 리눅스 기반의 테스트베드에 구현하고 기존 MPTCP 스케줄러와 성능 비교를 진행하였다.

ABSTRACT

Multipath TCP (MPTCP) is a standardized transport layer protocol for maximizing the use of finite network resources by using multiple interfaces simultaneously. If the characteristics of each path are the same, there is an advantage in terms of stability and bandwidth utilization compared to the existing single TCP. However, if the path characteristics are different, the performance is lower than that of a single TCP. There are many complex reasons for this, but one of the biggest impacts is the bufferbloat, which dramatically increases the latency. In this paper, we implemented an algorithm that improved MPTCP performance degradation due to bufferbloat in Linux - based testbed and compared performance with existing MPTCP scheduler

키워드

TCP, Multipath TCP, Bufferbloat, Scheduler

1. 서 론

기존 TCP의 네트워크 자원 활용의 한계를 극복하고자 2013년 IETF (Internet Engineering Task Force)에서 Multipath TCP (MPTCP)[1]를 표준화 했다. 대다수의 이동단말이 3G (혹은 4G)와 WiFi 등 두 개 이상의 인터페이스를 가지고 있지만 상이한 망의 특성, 버퍼블로트 등의 현상으로 인하여

MPTCP가 제 성능을 내지 못하고 있다. 이러한 문제들을 개선하기 위해 여러 연구들이 선행 되었다.

본 논문에서는 위와 같은 문제점 중 버퍼블로트의 발생원인에 대해 설명하고 리눅스 기반으로 경로차단알고리즘을 구현한다. 또한 이종망 특성을 가진 유선 테스트베드를 구축하고 성능 평가를 진행하였다.

II. 버퍼블로트 발생 원인 및 문제점

최근 급속도로 발달한 반도체 기술에 따라 메모리의 가격이 하락하였다. 또한 고품질 스트리밍 서비스나 무손실 음원 서비스 같이 방대한 트래픽의 수요가 증가함에 따라 손실을 방지하고자 중간 노드들의 버퍼사이즈가 과도하게 확장되었다. 이 때 TCP-Reno와 같은 손실 기반 TCP 혼잡 제어는 손실이 나기전에는 해당 경로의 혼잡 윈도우를 줄이지 않는다. 따라서 혼잡한 경로로 유입된 패킷은 큰 버퍼에 긴 시간 대기하게 되며 이는 수 초 이상의 Round-Trip Time을 야기하는 버퍼블로트 현상을 일으킨다.

III. 실험 환경 구성

실험환경은 <그림 1>과 같이 구성하였다. 총 네대의 PC를 사용하여 실험환경을 구성하였다. 현재 mptcp v0.90 미만의 버전에서는 mptcp 커널과 네트워크 파라미터 조절에 필수적인 역할을 하는 Netemulator와 간헐적인 충돌이 발생하고 redundant 스케줄러 사용시 빈번한 kernel panic문제가 있어 정상적인 실험이 불가능하다. 따라서 문제점을 개선한 mptcp 0.94 버전을 선택하여 설치 하였다 [2][3].

성능 평가를 진행할 때 서브플로우의 특성은 100Mbps 대역폭과 10ms 지연으로 설정하였다. 또 Head-of-line blocking에 의한 지연시간의 차이를 만들기 위해 Dummy PC 1번의 버퍼사이즈를 키우고 Destination으로 UDP 트래픽을 96Mbps로 20초간 흘려보냈다.

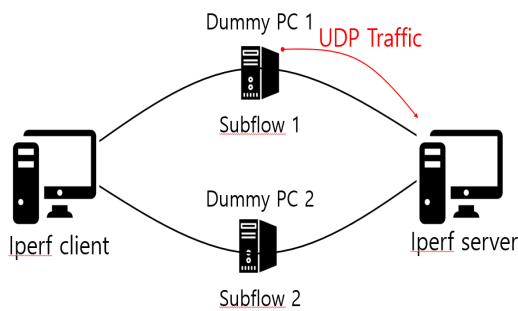


그림 1. 실험 환경 구성도

IV. 리눅스에 MPTCP DAPB 스케줄러 구현

DAPB (Delay Alerted Path-blocking) 스케줄러[4]의 구현을 위해 우선 mptcp 0.94 version을 다운로드 받아 커널 컴파일을 진행한다. 그리고 커널내부 mptcp_v0.94/net/mptcp에 있는 redundant.c 코드를

기준으로 하여 sRTT측정, 혼잡윈도우 크기조절, 차단결정 알고리즘, 순차적으로 패킷을 전송하는 round-robin 방식을 함수형태로 만들어 추가하였다.

구현은 다음과 같이 진행하였다. 먼저 sRTT측정 부분은 min_srtt 변수를 선언하고 srtt가 측정될 때마다 비교하여 가장 낮은 값을 저장하는 방식이다. 두 번째 혼잡윈도우 크기조절부분은 next_segment 함수에서 $tp->cwnd == 1$ 로 설정한다. 즉 차단된 경로의 cwnd를 1로 제한하는 방법이다. 다음으로 차단결정 알고리즘은 min_srtt와 현재 경로의 srtt를 비교하며 경로의 srtt가 min_srtt수치의 3배를 넘어가면 차단을 결정하는 알고리즘이다. 마지막으로 커널 내부의 mptcp_rr.c를 참고하여 각 경로에 순차적으로 최대세그먼트크기 만큼의 segment를 보내는 Round-robin 방식의 전송 알고리즘을 추가하여 구현을 완성하였다.

V. 측정 방법 및 스케줄러 별 성능

MPTCP 스케줄러 별 성능 측정을 위해 iperf[5]를 사용하였다. 60초간 데이터를 전송하며 20초~40초 구간에는 앞서 2장에서 언급한 UDP 트래픽을 20초간 흘려보내어 경로간 특성 차이를 유발하였다. LowestRTT와 Round-robin 스케줄러는 지연시간이 급격하게 증가한 상황에서 스케줄러 특성상 대처하지 못하는 양상을 보였다[6].

먼저 Round-robin 스케줄러는 경로 특성을 전혀 고려하지 않고 번갈아 보내기 때문에 지연시간이 급격하게 증가한 20초~40초 구간에서 가장 낮은 전송률을 보인다. 이로 인해 가장 낮은 성능을 보인다.

다음으로 Lowest RTT 스케줄러의 경우에는 낮은 RTT순으로 전송하는 방식이다. Round-robin 보다는 높은 패킷 전송률을 보이지만 지연시간이 급격하게 증가할 때 Subflow 1에도 적은 양의 데이터가 전송되어 성능이 하락한다.

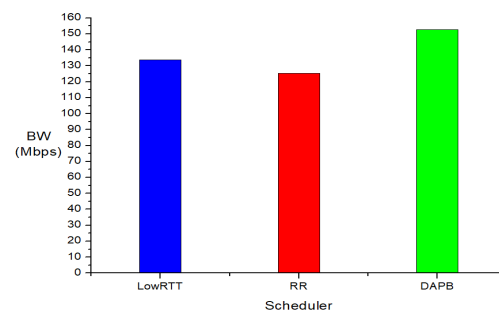


그림 2. MPTCP 스케줄러 별 성능

하지만 <그림 2>에서 볼수 있듯이 DAPB의 경우에는 Subflow 1의 지연시간이 증가하여 일정 수

준 이상이 되면 해당 Subflow를 차단하고 서브플로우 2에 모든 데이터를 전송한다. 이로 인해 UDP 트래픽이 추가되는 20~40초 구간에서도 단일 경로의 성능은 보장하여 전체적으로 가장 높은 전송효율을 보인다.

VI. 결론 및 향후 연구

본 논문에서는 이중 망 특성을 가지는 유선 테스트베드를 구축하고 리눅스 기반에 DAPB 스케줄러를 구현하였다. 그리고 이중 망의 특성을 가지는 환경을 구성하고 Lowest RTT, Roundrobin 그리고 DAPB 스케줄러의 성능을 측정하였다. 결과적으로 DAPB의 성능이 가장 우수하였음을 알 수 있었다. 향후 유선 테스트베드 뿐만이 아니라 무선 환경에서 MPTCP의 효율적인 데이터 통신을 위해서 더 많은 연구와 실험을 진행 할 예정이다.

Acknowledgement

본 연구는 이공분야기초연구 - 기본연구 사업의 지원을 받았음.

References

- [1] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation With Multiple Addresses", IETF document RFC 6824, Jan 2013
- [2] Multipath TCP - Linux Kernel implementation : <https://www.multipath-tcp.org/>
- [3] <https://github.com/multipath-tcp/mptcp/issues>
- [4] 김민섭; 이재용; 김병철. 이중 망에서의 지연 경보 경로차단 스케줄러를 이용한 MPTCP 성능 개선방안. 전자공학회논문지, 2017, 54.2: 28-37.
- [5] iPerf - The TCP, UDP and SCTP network bandwidth measurement tool : <https://iperf.fr/>
- [6] Paasch, Christoph, et al. "Experimental evaluation of multipath TCP schedulers." Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop. ACM, 2014.