Multipath SQRT Congestion Control for Multimedia Streaming

Tuan-Anh Le, and Choong Seon Hong

{letuanh, cshong}@khu.ac.kr

Dept. of Computer Engineering, Kyung Hee University

Abstract

In this paper, we introduce a multipath congestion control algorithm for audio/video streaming, called MPSQRT. MPSQRT is derived from single-path Square-Root TCP (SQRT TCP) designed for multimedia streaming, where a lower variation of sending rate is important. Based on the fluid model of SQRT for single-path, we extend it towards spreading concurrently packets across multiple paths ensuring load-balancing and fairness to SQRT at shared bottleneck. Through simulations, we evaluate the proposed protocol under various network conditions.

1. Introduction

In recent years, multimedia applications such as audio/video streams have been becoming the most popular in the Internet. Multipath TCP (MPTCP) allows multiple TCP connections across multiple paths between a two-end host pair [4]. Two main goals of MPTCP's congestion control [2] are improvement the performance, resilience and resource pooling [5]. MPTCP's congestion control is implemented by a coordinated congestion controller. Such a coordinated mechanism moves the traffic off from most congested paths to a lightly congested path. However, MPTCP is not suitable for multimedia applications because it reduces half of the transmission rate upon detecting a network congestion signal, so its data rate is very high variant. To guarantee the Internet stability, multimedia streaming should be smooth rate and should compete fairly with TCP traffic. In this paper, we introduce an extended version of single-path Square-Root TCP (SQRT-TCP [1]) for multipath transmission, called MPSQRT, which benefits smooth transmission rate, fairness to TCP traffic, improved throughput and load-balancing.

The rest of this paper is organized as follows: Section 2 presents the extension process of single-path SQRT TCP towards multipath transmission. The simulation results are introduced in Section 3. Section 4 gives our conclusion.

본 연구는 방송통신위원회의 차세대통신네트워크원천기술개발사업의 연구결과로 수행되었음 (KCA-2011-08913-05002). Dr. CS Hong is the corresponding author.

2. Multipath SQRT Congestion Control Algorithm

In this section, we briefly present single-path SQRT algorithm and then its extension process for spreading simultaneously data packets across multiple paths within a multipath TCP session.

SQRT algorithm [1] fulfills the requirement of the smoothed rate of multimedia stream by increasing inversely proportional and decreasing proportional to the square-root of the congestion window as follows:

Increase: $w^{sq} \leftarrow w^{sq} + \alpha/(w^{sq})^{k+1}$,

Decrease: $w^{sq} \leftarrow w^{sq} - \beta(w^{sq})^{l}$, (1)

where the superscript ^{sq} denotes the single-path SQRT algorithm with setting $k = l = 1/2$, and $\alpha = 1$ and $\beta = 1/2$.

We agree that any multipath transport congestion control algorithm should achieve three goals: (i) improved throughput; (ii) fairness to existing TCP traffic; (iii) load-balancing as described in MPTCP [2].

Now, we propose a multipath SQRT, called MPSQRT, where each sub-flow on path s carries out the congestion control as follows:

Increase: $w_s \leftarrow w_s + \min(\delta\alpha/w_s^{k+1}, \alpha/w_s^{k+1})$,

Decrease: $w_s \leftarrow w_s - \beta w_s$, (2)

where δ determines a coordinated control parameter between sub-flows within a multipath session.

MPSQRT above implies that whenever the sub-flow receives a positive ACK on path s, it increases its congestion window w_s by $\delta\alpha/w_s^{k+1}$. This increase amount is bounded by $,\alpha/w_s^{k+1}$ in order to guarantee fairness to TCP over that path. Moreover, the improved throughput and fairness goals suggest that the total throughput of a multipath SQRT flow should equal that of a single-path TCP flow on the best path for it [3] This implies that

$$\sum_r \hat{w}_r/RTT_r = \max_r\{\hat{w}_r^{sq}/RTT_r\}, \qquad (3)$$

where $\hat{w}$ and $\hat{w}^{sq}$ are the equilibrium values of w and w^{sp}, respectively.

By balancing increase and decrease rules in (1) and (2), substituting such results into (3), we obtain

$$\delta = \max_r\left\{\left(\frac{\hat{w}_r}{RTT_r}\right)^{k+l+1}\right\}\Big/\left(\sum_r \frac{\hat{w}_r}{RTT_r}\right)^{k+l+1}.$$

δ in the equation above is shared between the paths and just depends on the maximum throughput among paths. However, the congestion window sizes on the congested paths are always smaller than that on the lightly loaded paths. To perform load-balancing between paths, the congestion windows on the congested paths are increased more gradually than that on the better paths [6]. This means that δ for the worse paths would be smaller. Therefore we slightly modify δ, and then replace δ with δ_s as

$$\delta_s = \hat{w}_s^{\gamma}\max_r\left\{\left(\frac{\hat{w}_r}{RTT_r}\right)^{k+l+1-\gamma}\right\}\Big/\left(\sum_r \frac{\hat{w}_r}{RTT_r}\right)^{k+l+1},$$

where γ is a trade-off parameter between resource pooling and protocol fluctuation, $0 \le \gamma \le k+l+1$. Through our simulations, we found that $\gamma = 1$, which is reasonable trade-off between fluctuation and load-balancing.

3. Simulation results

In this section, we evaluate the variance of transmission rate of MPSQRT, compared with MPTCP [2]. The three goals for the design of multipath transport protocols (fairness, throughput improvement and load-balancing) are investigated by simulations. We use NS-2 [7] with RED, SACK option, and a 1000-byte data packet in all simulations.

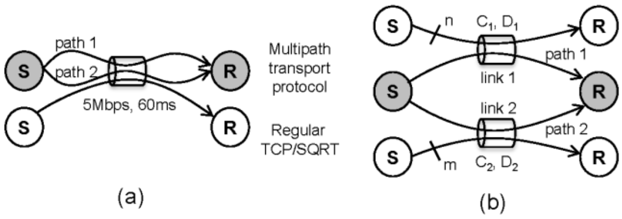

Figure 1. Simulation topologies.

3.1 Fairness and variance of transmission rate

We show how a multipath transport protocol fairly shares with a single-path protocol at a common bottleneck without any common bottleneck detection mechanism. We use a dumbbell scenario as shown in Fig. 1(a), where two multipath protocol sub-flows compete against a SQRT/TCP flow at the shared link with the same RTT. The top plots in Fig. 2 show that the single-path flows receive their data rate twice than that of each sub-flow. Hence the total throughput of a multipath protocol is equivalent to that of one single-path flow as shown in the bottom plots.

Moreover, Fig. 2 shows that MPSQRT gives smoother transmission rate than MPTCP because MPSQRT decrease their current window inversely proportionally to the square-root of the current window while MPTCP decreases its window by half.

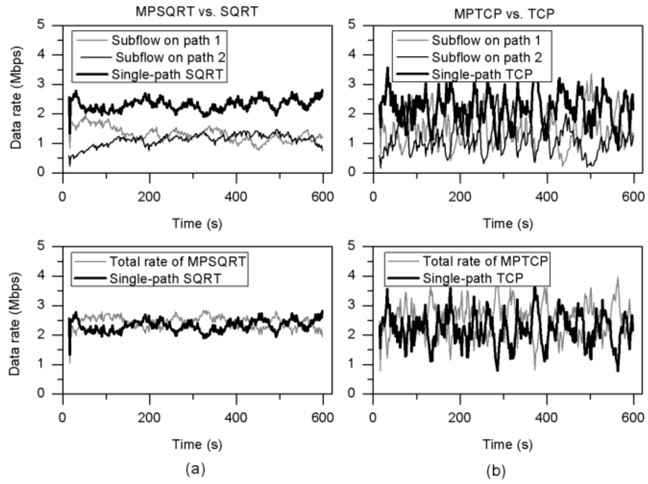

Figure 2. Fairness of a two-path MPSQRT flow vs. single-path SQRT (a); a two-path MPTCP flow vs. single-path TCP (b) at a shared bottleneck.

3.2 Throughput

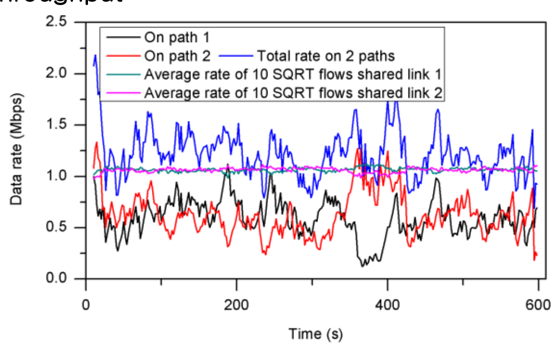

Figure 3. Throughput improvement of MPSQRT.

We investigate the throughput of multipath protocols in a scenario shown in Fig. 1(b), where two links have the same parameters with C_1 = C_2 = 12Mbps and D_1 = D_2 = 20ms, and the background traffic is generated by ten single-path SQRT flows on each link (n = m = 10). Fig. 3 shows that the total throughput of the multipath transport protocols is at least equal to the average throughput of ten SQRT flows. Therefore, such multipath transport protocols can yield more advantage since more paths are used.

3.3 Load-balancing

We investigate the capability of shifting traffic away the congested path towards the less congested path in an adverse network configuration, where RTT on the less congested paths is longer than that on the more congested paths. For such configuration, the congestion windows of RTT-dependent protocols on the longer RTT paths are grown more slowly than expected.

Experiments were run in the simulation scenario as shown in Fig. 1(b), where link capacity and propagation delay of links 1 and 2 are set to (10Mbps, 30ms) and (8Mbps, 90ms), respectively. To generate heavy load in link 1 and light load in link 2, we use SQRT flows as the background traffic with ten flows on link 1, and six flows on link 2. Fig. 4 shows that MPSQRT reduces its rate on the congested and shorter path (path 1), and surge their sending on the lightly loaded and longer RTT path (path 2). The total throughput of two MPSQRT sub-flows reaches the expected value. This is due to the MPSQRT sub-flow can compensate the congestion window increase for the lightly loaded and longer RTT paths.

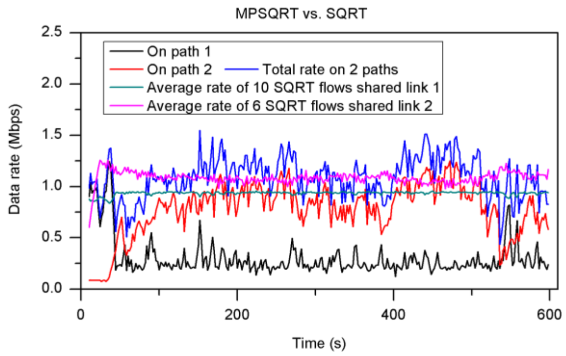

Figure 4. MPSQRT can compensate the congestion window increase meanwhile load-balancing as round-trip times of paths are divergent.

4. Conclusion

In this paper, we propose an extended version of SQRT for multiple paths, called MPSQRT. To start with the analysis model of SQRT, MPSQRT congestion control is designed as a coordinated control between paths which allows trade-off between load-balancing and fluctuation, and fair sharing to single-path SQRT flows at bottlenecks. Our simulations show that MPSQRT can achieve transmission rate smoother than MPTCP, fairness single-path SQRT flows, and load-balancing under various network conditions.

References

[1] D. Bansal and H.i Balakrishnan, “Binomial Congestion Control Algorithms,” IEEE INFOCOM 2001, 2001.

[2] C. Raiciu, M. Handley, and D. Wischik, “Coupled Congestion Control for Multipath Transport Protocols,” IETF RFC 6356, Oct. 2011.

[3] D. Wischik, M. Handley, and C. Raiciu, “Control of multipath TCP and optimization of multipath routing in the Internet,” Proc. NetCOOP 2009.

[4] A. Ford, C. Raiciu, and M.Handley, “TCP extensions for multipath operation with multiple addresses,” IETF Internet Draft, work in progress, Mar. 2011.

[5] D. Wischik, M. Handley, and M. B. Braun, “The resource pooling principle,” in ACM SIGCOMM CCR, Oct. 2008.

[6] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, “Design, implementation and evaluation of congestion control for multipath TCP,” Proc. of the 8th USENIX NSDI Conf., March 2011.

[7] NS-2 network simulator.