

웹 페이지에서의 QUIC과 TCP의 성능 비교

김정근*, 박창훈, 송영준, 엄원주, 김건환, 조유제

경북대학교

{kjg818, pch4495, syj5385, dnjswn9612}@knu.ac.kr, {kgh76, yzcho}@ee.knu.ac.kr

Performance Comparison of QUIC and TCP on the Webpage

Jeong-Keun Kim*, Chang-Hoon Park, Yeong-Jun Song, Won-Ju Eom,

Geon-Hwan Kim and You-Ze Cho

School of Electronics Engineering, Kyungpook National University

요약

2012년 Google은 TCP의 Head-of-Line Blocking 문제를 해결하기 위한 새로운 전송 프로토콜인 QUIC을 제안하였다. QUIC은 암호화와 안정적인 데이터 전송을 제공하는 UDP 기반의 저지연 전송 프로토콜로서 초기 연결 시 데이터를 직접 전송해 연결 시간을 줄이는 것을 목표로 한다. 본 논문에서는 웹 페이지에서의 파일 다운로드, 페이지 로딩 시간, 스트리밍 시간을 측정하여 QUIC과 TCP 성능을 비교하였다.

I. 서론

최근 인터넷 사용자가 늘어남에 따라 SPDY, HTTP 등 많은 웹 성능을 높이기 위한 프로토콜 연구가 진행되고 있다. 하지만 TCP 기반의 프로토콜인 HTTP는 웹 페이지의 이용률이 증가함에 따라 급증하는 트래픽으로 인해 응답시간이 느려지는 문제점이 발생하고 있다. 이로 인해 대부분의 사용자는 이를 기다리지 않고 해당 웹 페이지를 종료한다 [1].

기존 TCP는 3-way handshake에 의한 1-RTT (Round Trip Time)의 지연시간이 발생하며 패킷손실 또는 재전송이 이루어질 때 패킷이 전송되지 못하고 대기 상태에 머물러 있는 Head-of-Line (HoL) Blocking 문제가 발생하여 웹 성능이 저하될 수 있다 [2]. 이러한 문제를 해결하고 웹 성능을 향상시키기 위해 2012년 Google은 새로운 전송 프로토콜인 QUIC (Quick UDP Internet Connection)을 제안하였다 [3]. QUIC은 UDP 위에서 작동하며 TCP, TLS (Transport Layer Security)를 포함한 기존 프로토콜을 기반으로 설계되었고 초기 연결 시 데이터를 직접 전송해 연결 지연시간을 줄이는 것을 목표로 한다 [4].

Chiper Spec의 추가과정을 거치게 되어 3-RTT가 요구된다. 이에 비해 QUIC은 Client가 Server에 암호화되지 않은 *CHLO* 패킷을 전송한다. Server는 *Connection ID*를 기반으로 하는 암호화된 토큰을 포함한 *Reject* 패킷을 전송한다. 이때 전송된 토큰은 Client가 Server에 보내는 모든 요청을 암호화하는 데 사용된다. 서버에 보내는 연속적인 요청에 동일한 토큰을 사용함으로써 Server와의 연결 시간이 줄어든다 [5]. 따라서 QUIC은 한 번이라도 클라이언트와 서버가 데이터 전송을 수행한 적이 있으면 0-RTT로 작동한다 [6].

본 논문에서는 TCP와 QUIC의 웹 페이지에서의 연결 시간에 따른 성능을 비교하기 위해 파일 다운로드 시간, 웹 페이지 로딩 시간, Youtube 스트리밍 시간을 측정하여 비교 및 평가하였다.

II. 본론

A. TCP와 QUIC의 패킷 처리방식

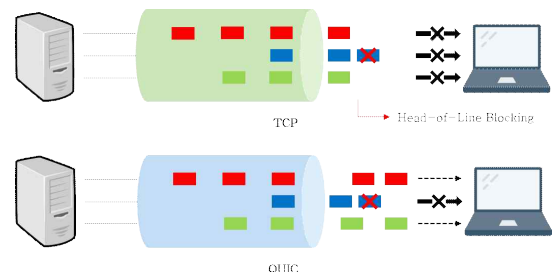


그림 2. TCP, QUIC의 패킷 처리방식 비교

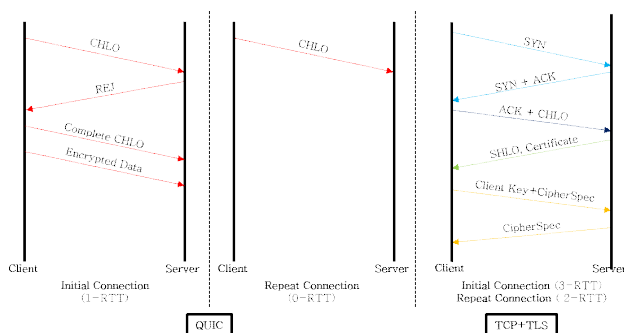


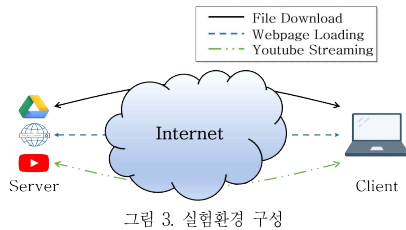
그림 1. QUIC, TCP+TLS Connection

그림 1은 QUIC과 TCP+TLS의 연결과정을 보여준다. TCP는 연결 시 *SYN*, *SYN+ACK*, *ACK* 과정을 거친다. TCP를 웹에서 사용하기 위해서는 전송 계층 보안 (TLS)에 의한 암호화 과정인 *Client Hello (CHLO)*,

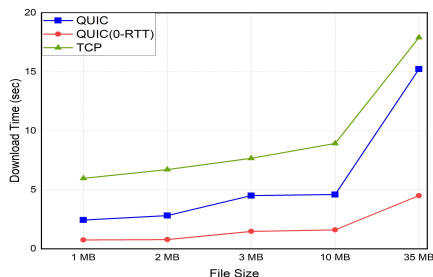
TCP는 신뢰성 있는 데이터 전송을 보장한다. 하지만 패킷손실이 발생하거나 재전송될 때 발생하는 지연시간에 의해 패킷이 전송되지 못하는 문제가 발생할 수 있다. 이에 비해 QUIC은 하나의 연결로 여러 데이터를 동시에 전송하는 스트림 다중화 (Stream Multiplexing)를 제공하여 동일한 스트림에 속하는 모든 데이터는 순서대로 전달되지만, 데이터의 손실

이 발생했을 때 다른 스트림의 데이터 전달을 차단하지 않아 HoL Blocking을 방지할 수 있다.

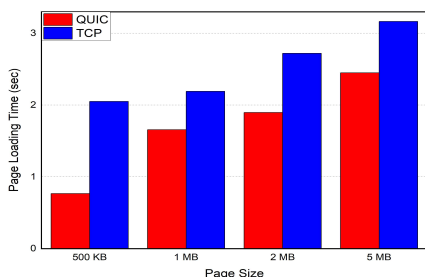
B. QUIC과 TCP의 웹 경험 비교



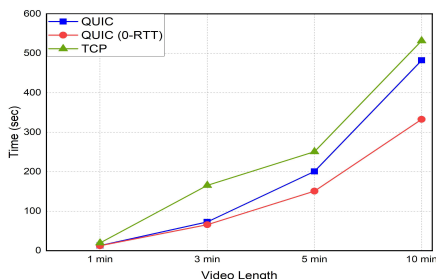
QUIC과 TCP의 성능을 비교하기 위하여 그림 3의 환경에서 실험을 진행하였다. 실험은 Google Chrome의 `chrome://flags`에서 QUIC의 사용을 Enabled 및 Disabled 하여 파일 다운로드, 웹페이지 로딩 시간, Youtube 스트리밍 시간을 측정하였다. 이때, QUIC은 클라이언트가 서버와 한 번이라도 연결을 한 경우 0-RTT로 작동한다.



(a) File Size 별 다운로드 시간



(b) Page Size 별 웹 페이지 로딩 시간



(c) Video Length 별 스트리밍 시간

그림 4. 웹페이지에서의 QUIC vs. TCP 성능 비교

Google Drive를 통해 1 ~ 35MB 크기의 파일의 다운로드 시간을 그림 4.(a)에 나타내었다. 재연결된 QUIC은 0-RTT의 연결 시간으로 인해 처음 연결하는 QUIC보다 빠른 다운로드 속도를 보였고 TCP가 가장 긴 시간이 걸렸다.

그림 4.(b)는 웹페이지의 크기 별 로딩 시간을 보여준다. 웹페이지에 포함된 구성요소를 불러오는데 걸리는 시간을 측정하였으며 짧은 연결 시간

에 의해 1-RTT의 QUIC이 3-RTT의 TCP+TLS보다 빠른 로딩 시간을 보였다.

Youtube 영상 길이별 스트리밍 시간을 그림 4.(c)에 나타내었다. 1분 길이의 짧은 영상에서는 초기연결 QUIC, 0-RTT의 QUIC, TCP의 큰 성능 차이가 나타나지 않았지만, 영상의 길이가 길어짐에 따라 0-RTT의 QUIC, 초기연결의 QUIC, TCP 순서로 빠른 스트리밍 시간을 보였다.

III. 결론

본 논문에서는 QUIC과 TCP의 파일 다운로드 시간, 페이지 로딩 시간, Youtube 스트리밍 시간을 측정하여 웹 성능을 비교 및 평가하였다. 재연결된 QUIC은 0-RTT의 연결 시간으로 인해 가장 짧은 응답시간을 보였고 초기연결의 QUIC이 다음으로 짧은 시간을 보였다. TCP는 패킷손실과 재전송으로 인한 Head-of-Line Blocking이 발생하여 QUIC에 비해 느린 응답시간을 보였다.

ACKNOWLEDGMENT

This research was supported by Next Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. NRF-2017M3C4A7083676) and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2018R1A6A03025109) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1A2C1006249).

참고 문헌

- [1] 김평수, “웹페이지 로딩시간 개선을 위한 IETF HTTP 표준화 개발 동향”, 정보통신기획평가원 주간기술동향 1901호, pp 15-26
- [2] Y. Cui, T. Li, C. Liu, X. Wang and M. Kühlewind, “Innovating transport with QUIC : Design approaches and research challenges”, IEEE Internet Computing., vol. 21, no. 2, pp. 72-76, 2017.
- [3] “QUIC, a multiplexed stream transport over UDP”, The Chromium Projects, (<https://www.chromium.org/quic>).
- [4] O. Nalawade, A. Dhanwani, and T. Prabhu, “Comparison of Present-day Transport Layer network Protocols and Google’s QUIC”, 2018 International Conference on Smart City and Emerging Technology, 2018
- [5] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, W. Chang, and Z. Shi. “The QUIC Transport Protocol : Design and Internet-Scale Deployment”, SIGCOMM ’17: Proceedings of the Conference of the ACM Special Interest Group on Data Communication, pp. 183-196, 2017
- [6] G. Carlucci, L. D. Cicco, S. Mascolo, “HTTP over UDP: an experimental investigation of QUIC”, SAC ’15: Proceedings of the 30th Annual ACM Symposium on Applied Computing, pp. 609-614, 2015