

QUIC & WebTransport: An Investigation Into the Future of Video Conferencing

Alex Paterson

2382854p

Proposal

Motivation

The internet is in a constant state of change, with practices and protocols constantly evolving to address long-standing issues and push development forward. One prolific example of this is the recent standardisation of QUIC. The protocol claims to offer improved performance and security compared to its contemporaries, namely TCP, and has quickly been picked up by developers to introduce accompanying new technologies. Two notable instances of this are HTTP/3 and subsequently WebTransport.

WebTransport uses HTTP/3, which in turn uses QUIC, and aims to be the definitive protocol to use for client-server connections in certain scenarios. Specific use cases include sending application state at regular intervals with minimal latency to a server, receiving media streams pushed from a server with minimal latency independent of other data streams and receiving notifications pushed from a server while a web page is open. The latter two use cases are particularly intriguing for applying to a video conferencing context. Currently, only one mainstream video conferencing service, Google Duo, utilises QUIC.

Utilising QUIC (and WebTransport to facilitate this use of QUIC) could prove beneficial to both developers and users. Developers could have more flexibility and an easier development experience in comparison to contemporary technologies such as WebRTC and users could have a better experience.

Aims

The main goal of this project is to quantitatively evaluate the performance of several builds of a simple video conferencing application using different technologies (WebRTC, WebTransport), programming languages (python, C/Rust) and methods for transferring data (streams, datagrams). Quantitative metrics for measuring performance shall include latency, video/audio quality, video stall and general performance under forced congestion control and simulated network degradation. Furthermore, another key question this project aims to answer is whether or not the extra overhead of developing such technologies and implementing them makes a practical difference to the user experience or not - qualitative experiments shall be carried out to evaluate participants' experience when using the different builds. Overall, the project aims to investigate the suitability of using WebTransport, HTTP/3 and QUIC in a live video conferencing application and evaluate whether or not the effort required for using such developing technologies is practically worthwhile with respect to the end-user experience.

Progress

- Project spec more fully fleshed out – focus and aim changed to work with live video conferencing and WebTransport.
- Python example server found and adapted.
- WebRTC implementation with audio and video working.
- WebTransport datagrams implementation with video working but not very effectively (python server).
- WebTransport streams implementation with video nearly working, some bugs to iron out, but will likely not run effectively (python server).
- Plenty reading, evidence, notes to write about in dissertation.

Problems and risks

Problems

- WebTransport has proven to be very hard to develop with
 - o Documentation is sparse and sometimes even incorrect or unhelpful. There has been a lot of trial and error in getting the implementations working which has made things a lot more time-consuming than expected.

Risks

- I may not get streams to work well at all.
 - o This would leave me with not a lot to work with and get data from.
- Translating the server from python to C/Rust may be too time-consuming or difficult
 - o An example server was not publicly available – again, a disadvantage of using WebTransport.
 - o Translating the server to a faster language would be beneficial for user experiments as otherwise I can expect the participants to give negative feedback. This would be a problem as it would really be evaluating Python instead of WebTransport and QUIC.
- I might not find an effective way to evaluate the builds.
 - o There has not been much effort devoted to this so far as most of my time has been spent on implementation due to the previously discussed problems.

Plan

Christmas Break

- Finish all implementations and get them in a good state
 - o Get streams working
 - o Translate server to C/Rust or accept that python will have to do.

Semester 2

- **Week 1 – Week 4 (ending 6th Feb):** investigate methods for quantitative data studies.
 - o **Deliverable:** quantitative studies will be fully designed.
- **Week 5 (ending 13th Feb):** conduct quantitative data studies.
 - o **Deliverable:** all quantitative data studies would have been run and data necessary for the qualitative data studies and dissertation will be gathered.
- **Week 6 (ending 20th Feb):** design user studies.
 - o **Deliverable:** qualitative user studies will be fully designed.
- **Week 7 (ending 27th Feb):** conduct user studies.
 - o **Deliverable:** all qualitative data studies would have been run and data necessary for the dissertation will be gathered.
- **Week 7 – Week 10 (ending 20th March):** work on dissertation.
 - o **Deliverable:** first draft of dissertation submitted by 11th March for Colin to evaluate.
- **Week 11 (due 25th march):** finish dissertation.
 - o **Deliverable:** final draft of dissertation.