Inferring Video Conferencing Quality of Experience from Raw Packet Capture Data
CS 110

Sam Liang, Mateo Wang, and Ethan Wu

November 29, 2021

1 Introduction

It is constant progress that we want to improve the users' quality of experience when using the Internet, and how researchers and the network service providing industry can provide better and more efficient Internet services to users will always be an unresolved issue. In order to improve the quality of experience of the users, we must first understand what are the contributing factors that influence the users' experience, and later, building from these results, researchers or the industry will be able to improve or manipulate the networks to improve the users' quality of experience.

Recently, there has been a large shift towards online video conferencing due to COVID-19. While much research has been done on video streaming applications, relatively little has been done for video conferencing applications (VCAs). In particular, some VCAs make it difficult to evaluate the quality of experience (QoE) and even the quality of service (QoS) metrics from the network or even application layer. In particular: Zoom, a popular VCA controlling 76 % of the video-conferencing market as of June 2021 [1]; uses its own closed-source transport protocol [2] and only reports application performance at a minute's granularity [3]. This makes it difficult to evaluate QoE and QoS. Only by drawing relations between raw packet capture data and QoE metrics, we would be able to know more about possible ways we can further optimize the network and the experience of the users, when the network resources are limited, by updating the algorithms used by the programmable switches and servers. The current progress is done on the end-host machines, which enable us to access the WebRTC data, and these QoS data will help us analyze the contributing factors that influence the QoE. However, our ultimate goal is to draw relations from packets captured at the border of UCSB to the Quality of Experience of the students, which means we only have access to encrypted data carried in UDP by that time.

1.1 UDP

UDP, User Datagram Protocol, is a lightweight transport protocol, providing minimal services. UDP is connectionless, so there is no handshaking before the two processes start to com- municate. UDP provides an unreliable data transfer service—that is, when a process sends a message into a UDP socket, UDP provides no guarantee that the message will ever reach the receiving process. Furthermore, messages that do arrive at the receiving process may arrive out of order. UDP does not include a congestion-control mechanism, so the sending side of UDP can pump data into the layer below (the network layer) at any rate it pleases. [4] Since UDP is a minimal transport layer protocol, the things we can get would be the source port, the destination port, and the application-layer-encrypted payloads. The payloads of UDP are encrypted because UDP does not secure the information it carries, so most Software companies add their own layer of encryption and protection at the application layer, which means normally only applications at the receiver side can decrypt the payloads and access the data.

1.2 Quality of Service (QoS)

Quality of service (QoS) is a metric expressing the performance of the network. One important thing we need to know is packets: During video meetings or video conferances through Internet, computers need to send and receive data. The data is packed into chunks, which we call packets, and sent through the links and into the Internet. The protocol used by most VCAs is the User Datagram Protocol (UDP). However, UDP packet headers do not contain any information about the network quality. Instead, a limited amount of network information is recorded by the application-layer protocol, WebRTC.

1.2.1 WebRTC

WebRTC is a program that introduce a set of application-layer protocols used by video conferencing applications. It is built upon RTP. While some of the popular VCAs use WebRTC, Zoom uses its own closed-source extension [2], making it difficult to evaluate QoE and QoS metrics from the network layer. While the Zoom API does provide some QoE information, it is at a minute's granularity [3], which is both at the application layer and is not fine enough for our purposes. Here are some of the elements that WebRTC collects:

Uplink Bitrate The *uplink bitrate* or *uplink capacity* is the rate at which the client sends packets to other users or to a relay server. It is usually measured in Mbps. A low uplink bitrate generally leads to poor audio/video upload quality.

Downlink Bitrate The *downlink bitrate* or *downlink capacity* is the rate at which the client downloads from other users or from a relay server. It is also measured in Mbps. A low downlink bitrate generally leads to low resolution video, poor refresh rates, and other drops in video quality.

Packet Loss Network data may be lost in transit, quantified as *packet loss*. This may be because of software factors such as network congestion, or physical factors such as poor wireless signal or broken cabling. Packet loss is measured as a percentage over a given time period.

1.3 Quality of Experience (QoE)

Quality of Experience is the user's perception of QoS [5]. Because QoE is a subjective metric, it is typically measured using opinion surveys over experiment participants [5], [6]. Since the

QoS metrics do not necessarily translate one-to-one to their perception of service quality, QoE has supplanted QoS as the barometer of user satisfaction [7].

While surveying users is an accurate measure of QoE, it is costly, not real-time, and only viable in a laboratory environment [5], [8]. Because of this, QoE must generally be modeled using the QoS metrics available to network operators. One such approach is that of Dinaki, Shirmohammadi, Janulewicz, *et al.* [7], who use a heuristic for video streaming QoE factoring the playback duration, join time buffering, buffering length, buffering frequency, and average bitrate. Song, Ge, Mahimkar, *et al.* [8] describe another QoE metric, Q-score. These systems use the QoS data in order to infer the user's QoE in real-time, allowing the network operator to make decisions on how to optimize the network performance.

Video conferencing in particular is impacted by QoS metrics such as the *frames per second* (FPS), *quantization parameter*, and the *video resolution* [9].

Quality of experience can be impaired by visual distortions. In video streaming, these distortions may include [10]:

Blocking Effect The reconstructed image contains block-shaped discontinuities due to the frame being encoded in blocks.

Blurring The amount of detail in the image is lost, and the sharpness of edges is reduced.

Color Bleeding Detail is smoothed out and areas with contrasting colors show smears of the other color.

Edginess The edges of the image show distortions [5].

Jerkiness "Snapshots" of a continuous motion appear in succession, producing a "disjointed sequence".

Quality of experience can also be negatively impacted by the format with which the video is compressed. When automatically monitoring for QoE, systems often consider the bitrate and the frame rate [5]. The bitrate is the rate at which the video codec compressing the video emits data, and the frame rate is the number of frames displayed per second. Moreover, the video itself may impact the QoE—videos with low amounts of movement may appear to have higher quality than videos with high amounts of movement when subjected to jitter or packet loss, for example [5].

1.4 Proactive QoE Management

Real-time QoE monitoring can be used to improve the perceived QoE of users on the network. For example, the network may use the QoS parameters to proactively estimate the user's QoE, and choose to reprioritize currently buffered packets in order to optimize said QoE [7].

We intend to use data collected passively from UCSB's network in order to create machine learning models for users' quality of experience using video conferencing from packet capture data available to the network operator. We believe that this research will be helpful for future developments of regional network control and optimization algorithms since it provides the important mapping from raw packet capture data to users' Quality of experience for Video Conferancing Applications.

2 Proposed Solution

In order to map encrypted packet capture data to quality of experience for video conferencing applications, we separate our research into two major parts: Firstly, collect QoS data from the network and perform statistical analysis or even machine learning models to correlate it to QoE data obtained experimentally from devices. Secondly, use the ML model to categorize the encrypted data into different categories of QoS status. By combining these two sub-goals, we can draw a mapping from encrypted packet capture data to the Quality of experience of the users.

To control and optimize QoE using QoS, we will first actively collect QoE and QoS data from video conferencing applications such as Google Meet and Microsoft Teams. The QoE data will comprise statistics reported by the VCAs' respective WebRTC data dumps, including such statistics as frame rate, change in frame rate, and resolution. The QoS data will comprise packet capture data from the network, including such statistics as packet loss, available bandwidth, and upstream/downstream data rate. We will then characterize these data by drawing visualizations of their probability and cumulative density functions, in order to find their distributions. With this information, we will be able to determine which QoS parameters impact QoE. Once the parameters that impact QoE are known, we will be able to set up a machine learning model to predict QoE when passed this parameter but omitting factors that may be unavailable to the network operator.

Later, in order to predict QoE from raw packet capture data, we will perform machine learning training, categorizing sessions of each end-host into different categories of QoS. For example, if an end-host experiences packet loss due to a bad network, the packets containing WebRTC data will still contain that information, though it is encrypted. Machine learning will implicitly hold on to that information and put it into the category of high packet loss. In such a way, we create a general mapping from raw data to QoS metrics. By merging our result from the previous step, we are able to monitor the QoE base on packets captured at the network border and provide a path for other researchers, or our professor's other research group, to implement network optimizing algorithms on top of it.

We expect more difficulties in the second part of our research since it would be hard to find the correct Machine Learning model and obtain a correct and well-fitted mapping from raw data to QoS. The relationships will be tough to draw because the packets also contain information about the videos and audios of the meeting, which are redundancies for our analysis. The encryptions from the application layer will mix the real payloads with the WebRTC data, and create some obstacles for our analysis.

3 Evaluation Plan

In order to evaluate our approach, we need to determine if we have collected enough data that we can generalize our result to a greater scope. The first goal of our project is to ensure that we have reliable data that are significant enough for our research. By having a metric that keeps track of the packet capture rate, we can immediately tell if there will be problems with the data we collected when the capture rate drops.

The earlier goal is to determine how well the trends of QoE data match with QoS data. The way we determine this would be general statistical analysis, which analyzes the relationships, relatabilities, and confidence intervals. By generating plots and graphs, we should be able to view the result very clearly. Moreover, if Machine Learning models are implemented, we can always use metrics, like accuracy, precision, and recall, to evaluate the models. Furthermore, we can always capture packets and generate new training and test sets on our test router, for further training and testing.

The later goal is to make sure that our machine learning model mapping from encrypted packet payloads to QoE does not overfit or underfit the data we use to train it. It should be applicable to real-world networks. It is also very easy to evaluate: by running it on a private router, we would be able to manipulate the QoS like uplink and downlink bitrates by ourselves and see the matches and differences between the calculated QoE and the real QoE.

4 Timeline

- Fall 2021
  - *Weeks 1–6*
    - Read through related papers.
    - Develop a monitoring tool for watching machine statistics.
  - *Weeks 7–10*
    - Collect QoE and QoS data.
    - research in statistical methods that could be used to analyze the data.
    - Find patterns in data.
- Winter 2022
  - *Weeks 1–6*
    - Research on good AI/ML models that fit our project.
    - Implement and start AI/ML training using our data.
  - *Weeks 7–10*
    - Collect more data and train the model.
    - Draw the outline of the research paper.
- Spring 2022
  - *Weeks 1–10*
    - improve the result and drafting the paper.
    - Write the research paper.
    - Wrap up project as needed.
    - Prepare presentation.

References

[1] T. Kim, “Zoom won’t stop at video chat dominance,” *Bloomberg*, 2021. [Online]. Available: `https://www.bloomberg.com/news/newsletters/2021-08-04/zoom-won-t-stop-at-video-chat-dominance/`.

[2] B. Marczak and J. Scott–Railton, “Move fast and roll your own crypto: A quick look at the confidentiality of Zoom meetings,” *Citizen Lab*, 2020. [Online]. Available: `https://citizenlab.ca/2020/04/move-fast-roll-your-own-crypto-a-quick-look-at-the-confidentiality-of-zoom-meetings/`.

[3] N. Walia, “Here’s how Zoom provides industry-leading video capacity,” *Zoom Blog*, 2019. [Online]. Available: `https://blog.zoom.us/zoom-can-provide-increase-industry-leading-video-capacity/`.

[4] J. F. Kurose, *Computer networking : a top-down approach featuring the Internet / James F. Kurose, University of Massachusetts, Amherst ; Keith W. Ross, Polytechnic University, Brooklyn*, eng, Third edition. Boston: Pearson/Addison Wesley, 2005, ISBN: 0321227352.

[5] Y. Chen, K. Wu, and Q. Zhang, “From QoS to QoE: A tutorial on video quality assessment,” eng, *IEEE Communications Surveys and Tutorials*, vol. 17, no. 2, pp. 1126–1165, 2015, ISSN: 1553-877X.

[6] D. Z. Rodrriguez, R. Rosa, E. Costa, J. Abrahao, and G. Bressan, “Video quality assessment in video streaming services considering user preference for video content,” eng, *IEEE transactions on consumer electronics*, vol. 60, no. 3, pp. 436–444, 2014, ISSN: 0098-3063.

[7] H. E. Dinaki, S. Shirmohammadi, E. Janulewicz, and D. Cote, “Forecasting video QoE with deep learning from multivariate time-series,” eng, *IEEE Open Journal of Signal Processing*, vol. 2, pp. 512–521, 2021, ISSN: 2644-1322.

[8] H. Song, Z. Ge, A. Mahimkar, J. Wang, J. Yates, Y. Zhang, A. Basso, and M. Chen, “Q-score: Proactive service quality assessment in a large IPTV system,” eng, in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC ’11, ACM, 2011, pp. 195–208, ISBN: 9781450310130.

[9] K. MacMillan, T. Mangla, J. Saxon, and N. Feamster, “Measuring the performance and network utilization of popular video conferencing applications,” eng, 2021.

[10] M. Yuen and H. Wu, “A survey of hybrid MC/DPCM/DCT video coding distortions,” eng, *Signal Processing*, vol. 70, no. 3, pp. 247–278, 1998, ISSN: 0165-1684.