M-Lab Research Proposal

Sanjay Chandrasekaran
sanjaychandrasekaran@ucsb.edu

April 2020

1 Introduction

The proliferation of Internet-connected devices (e.g., mobile phones, smart devices, etc.) and applications (e.g., video streaming and conferencing, augmented reality, etc.) has made the management of networks onerous, complex, and seemingly intractable. These devices must rely on their underlying networks to provide a high quality of experience (QoE) to the end user. Although network operators, policymakers, or any entity that does not control either end of network connections (we can refer to them as third parties) are interested in measuring QoE of the end-users, measuring it at scale is non-trivial.

Any third party interested in measuring QoE can either use active and/or passive measurements. Passive measurement techniques extract insights from packet streams in the network to infer QoE for a subset of services. For example, NetMicroscope [1] and Requet [2], use features extracted from the network and transport layer of packets to infer QoE metrics, such as buffer size, resolution, etc. However, these learning models use network data collected from controlled settings to generate labelled data for training, and it remains unclear how well this data samples real-world workloads. As a result, these learning models may not perform well when deployed in realistic network settings. Most of the passive measurement techniques to measure QoE are hard to scale, and although there are proposals to use programmable switches to infer QoE at line rates, most face a very high adoption threshold.

Active measurement techniques typically entail running measurements from end-host devices. Typically, these tests entail downloading files from remote servers to measure access/bottleneck link capacity, and related metrics such as latency, jitter, packet loss rate, etc. More sophisticated tests entail running video streaming and conferencing sessions to collect QoE metrics from the end hosts; and although using active measurement techniques to measure QoE is promising, they are not always representative of end-users QoE as they are influenced by various factors, such as device types, signal strengths (wireless links), and other competing network traffic. These biases within active measurements can be countered by collecting a more diverse set of samples, both in terms of space (e.g., devices, network conditions, etc.) and time. However, scaling active measurements for more users/devices, to achieve more confidence in the data, can be infeasible.

As an effort to democratize networking research, especially in the areas of AI and ML, researchers at UCSB, the University of Oregon, and the University of Chicago have proposed to develop an infrastructure that collects fine-grained, labeled data from campus networks. I have been working as a lead student on this project at UCSB. As part of this effort, I have developed a data processing pipeline that leverages programmable switches to anonymize and process the incoming packet stream. Currently, we handle all packets coming in and out of our campus' border routers. In the future, we plan to ingest traffic from various other vantage points on the campus network. Our pipeline can collect network packet traces for durations as long as several days. Additionally, we can combine these packet traces from logs collected from devices, such as the wireless Aruba controllers, within our network to add more context about the shared resources between different hosts.

2 Methodology

I am currently leading the effort to deploy hundreds of single-board computers (e.g., Raspberry Pi, Jetson Nano, etc.) across UCSB's campus. I am currently developing the backend that manages these devices to programmatically run a wide variety of active measurements, such as video streaming and conferencing sessions (e.g., Youtube, Zoom), on demand. Being able to run active and passive measurements concurrently at such a scale offers an incredible opportunity to explore fundamental questions related to the measurement of user's QoE. Below, I list the questions that I plan to investigate using this infrastructure at UCSB.

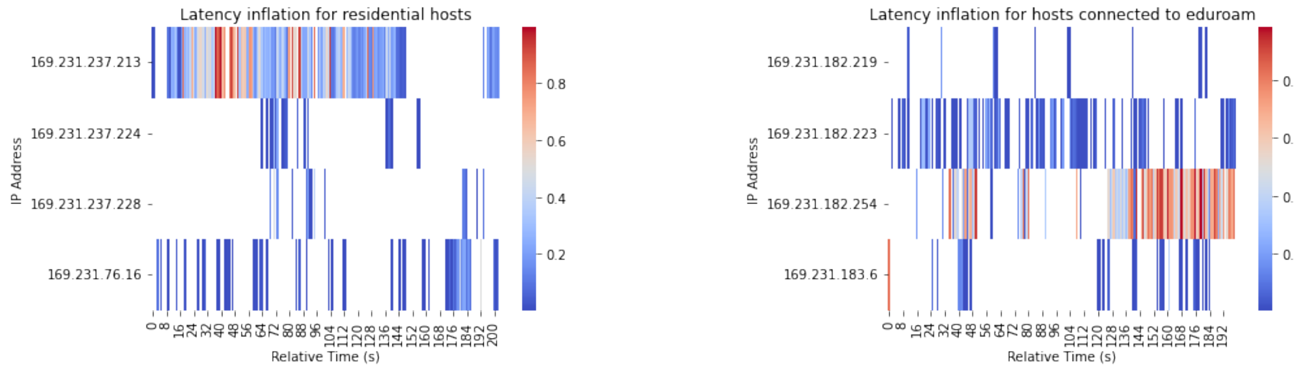

Figure 1: These heatmaps show the latency inflation for a few selected hosts every 1 second for users in residential homes, as well as hosts on campus connected. We define latency inflation for a window as the difference between the maximum last-mile RTT of any packet in the current window and the minimum last-mile RTT seen for over all windows for that IP address. Areas where the RTT fluctutates are indicated by oscillating blue / light blue lines and significantly high RTTs (inflated by up to 1 second) are indicated by the red lines.

1. **Can we draw a relationship between QoE and lightweight metrics, collected through both active and passive techniques?**
   For example, a fluctuation in last-mile round-trip time (RTT) can indicate a reduction in QoE for video conferencing applications which suffer from high latency. We can passively measure the last-mile RTT for TCP packets as the time difference between a data packet and its corresponding acknowledgement. Figure 1 shows a sample of these measurements for hosts that have a last-mile RTT with high variance. We can use `traceroute` and `ping` tests to measure the RTT from various hops within the last mile of the network to help zoom in further on any issue. We can combine these measurements with QoE labels generated from a diverse set of network conditions, to give us ground truth about the end users' experience. Through M-Lab's active measurements, we can explore other metrics as well such as throughput and loss to identify features that we can correlate to changes in QoE.

2. **Can we characterize any biases within M-Lab's active measurements?**
   Using this deployment, we will have more context for every active measurement that we run. This will allow us to characterize how certain factors such as background traffic, signal strength, etc. can affect the quality of our measurements.

Answering these questions are critical in helping bridge the gap between what we can observe from the network to how we can improve the user's experience. These findings can enable other researchers to make better use of M-Lab data with respect to improving users' QoE.

3 Limitations/Challenges

One limitation for this project stems from the use of low commodity hardware for active measurements in an attempt to scale up the number of vantage points we can generate data from. We face a tradeoff between scalability and cost where must ensure that the active measurements are not skewed by the device type/capabilities. Specifically, in the case of QoE labels, we want to ensure that we can produce application behavior that does not signifcantly differ from the needs of more common devices such as laptops, phones, etc. that are used more frequently.

4 Timeline

January - Backend

- Complete initial deployment (5-10 devices)
- Test the fidelity of active measurements
- Process and combine passive and active measurements

Febuary/March - Data analysis

- Analyze data from initial deployments
- Slowly scale up data collection, iterate and improve on the pipeline

March/April:

- Large scale data collection / analysis

May - Documenting

- Writing research articles, open sourcing the data, code, etc.

References

[1] C. Gutterman, K. Guo, S. Arora, X. Wang, L. Wu, E. Katz-Bassett, and G. Zussman, "Requet: Real-time qoe detection for encrypted youtube traffic," in *Proceedings of the 10th ACM Multimedia Systems Conference*, MMSys '19, (New York, NY, USA), p. 48–59, Association for Computing Machinery, 2019.

[2] F. Bronzino, P. Schmitt, S. Ayoubi, N. Feamster, R. Teixeira, S. Wassermann, and S. Sundaresan, "Lightweight, general inference of streaming video quality from encrypted traffic," *CoRR*, vol. abs/1901.05800, 2019.