Application-header based traffic engineering for ABR video streaming*

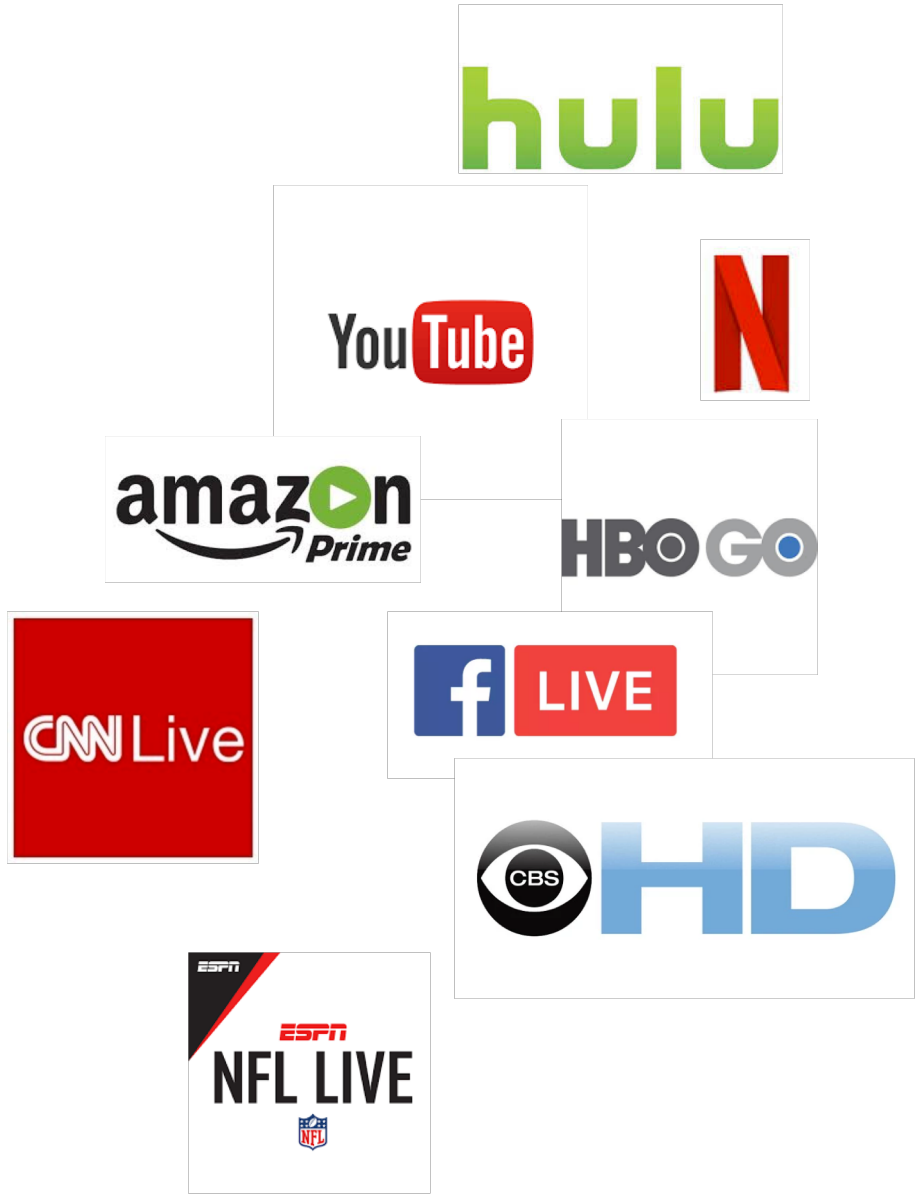
Presented By: Divya Bhat

*Presented as a demo at LCN 2018:
Application-based QoS support with P4 and OpenFlow: A demonstration using Chameleon -
Divyashri Bhat, Jason Anderson, Paul Ruth, Michael Zink and Kate Keahey

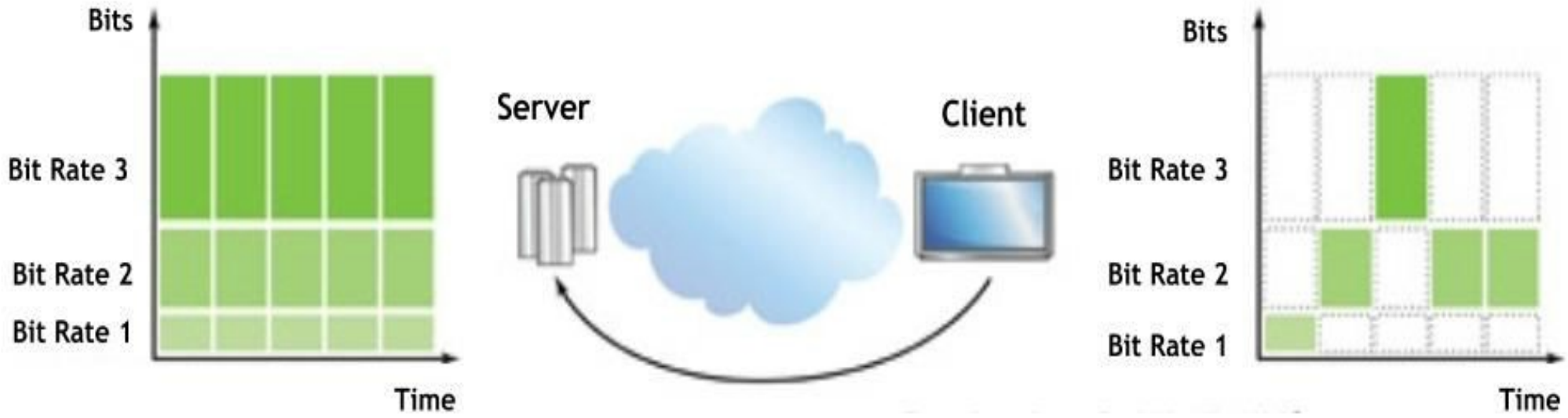
Video Streaming

- Killer Application
  - Over 71% of downstream Internet traffic at peak hours in North America is video.
  - Predicted to increase to 80% by 2020
  - Live, Video-on-Demand and User Generated

- Challenges
  - High bandwidth
  - Low delay
  - Various encoding formats – Real Time
  - and many more…



Adaptive Bit-Rate (ABR) Video Streaming



- Single Video split into multiple segments in various qualities
  - Sequentially request video segments
- On-demand and Live – Dynamic Adaptive Streaming over HTTP (DASH)
- Download Buffer – stores segments in queue for playback

ABR Streaming – QoE Metrics

- Average Quality Bitrate
  - Segment download rate
- Number of Quality Changes
  - Quality gaps
- Magnitude of Quality Changes
- Rebuffering Ratio
  - Segment transfer time

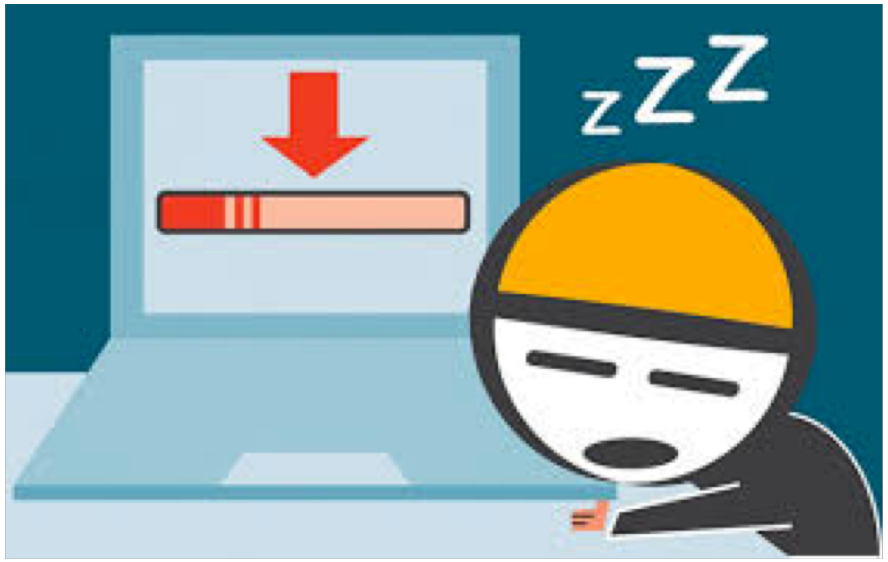
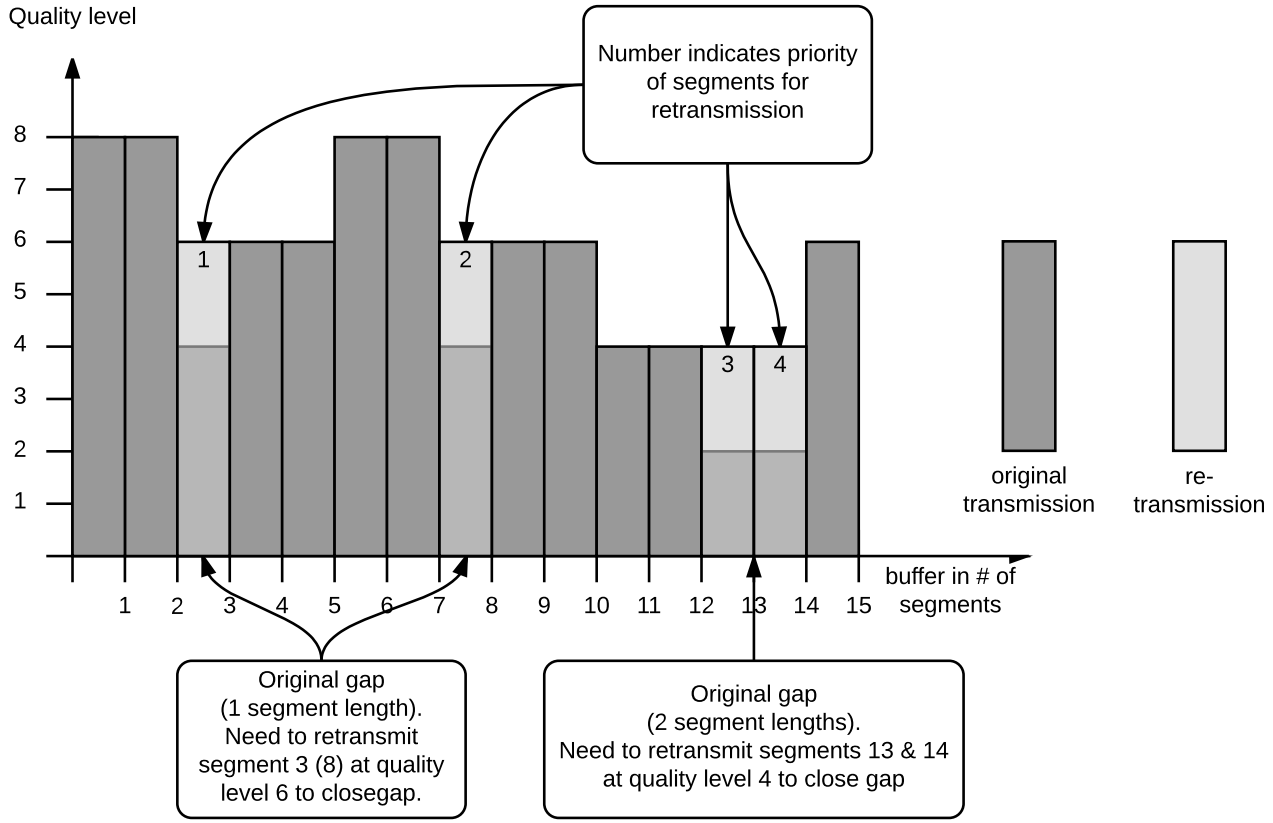
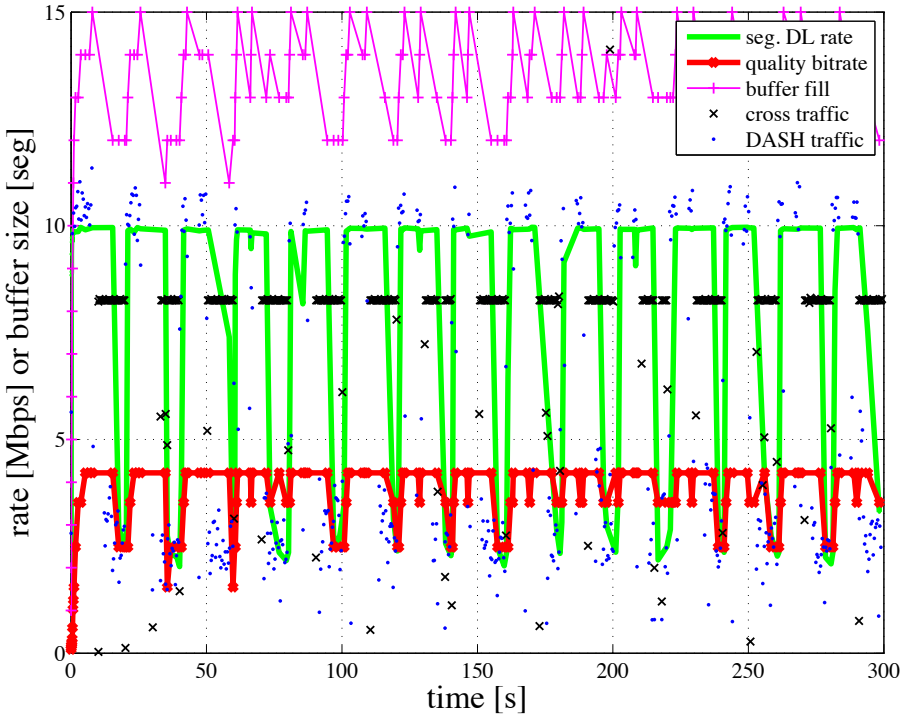


Image courtesy:
https://www.tipard.com/

ABR Segment retransmissions using HTTP/2

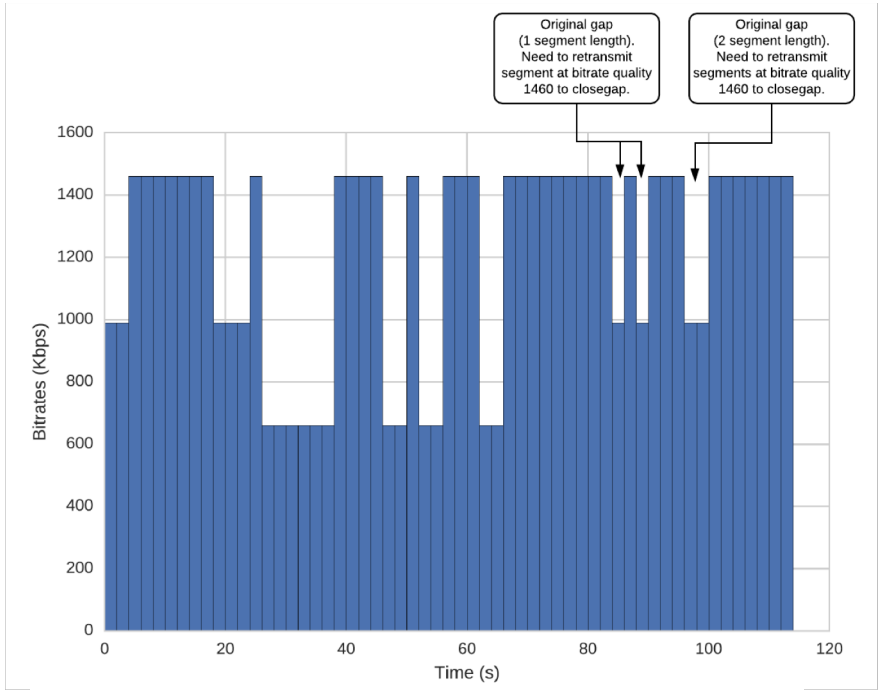


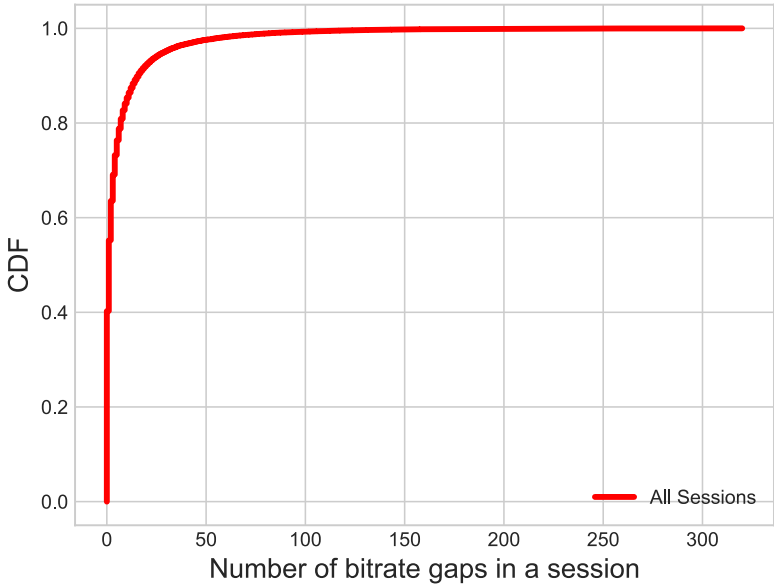


- [1] Wang, Cong, Divyashri Bhat, Amr Rizk, and Michael Zink. "Design and analysis of QoE-aware quality adaptation for DASH: a spectrum-based approach." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 13, no. 3s (2017): 45
- [2] Divyashri Bhat, Rajvardhan Deshmukh, and Michael Zink. 2018. Improving QoE of ABR Streaming Sessions through QUIC Retransmissions. In 2018 ACM Multimedia Conference on Multimedia Conference (MM '18). ACM, New York, NY, USA, 1616-1624.

Analysis of Quality Gaps - Akamai

- Akamai - world's largest CDN provider
  - Delivers 15%-30% of global Internet traffic
  - 3-day trace in June 2014
- 36.19% of sessions (~5 million) have at least one quality gap
- ~10% sessions experience quality changes higher than 40 per session

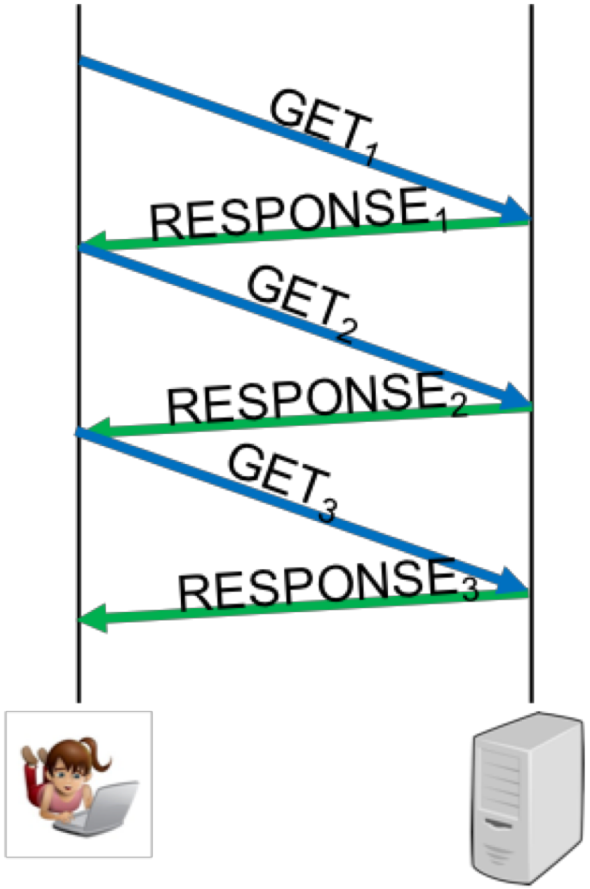




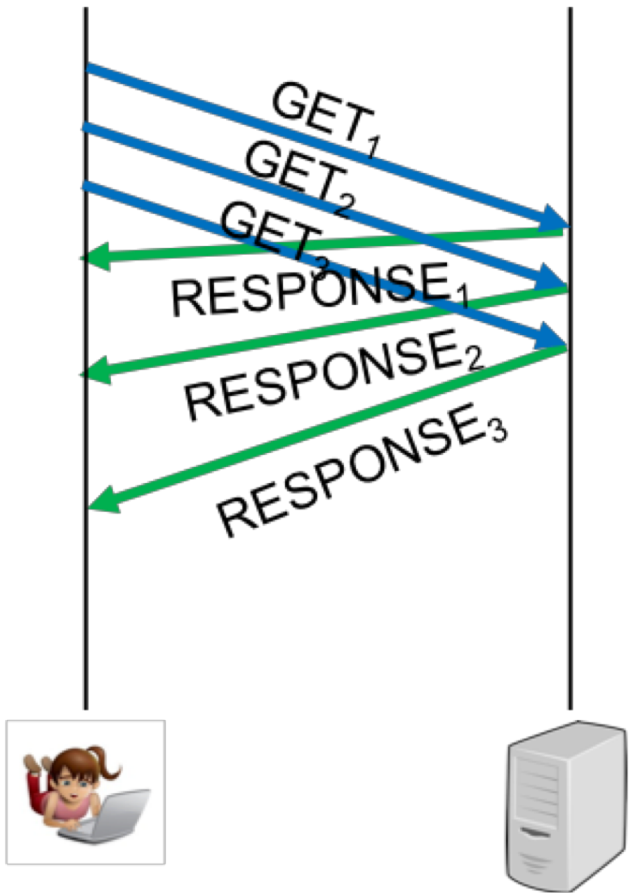
[2] Divyashri Bhat, Rajvardhan Deshmukh, and Michael Zink. 2018. Improving QoE of ABR Streaming Sessions through QUIC Retransmissions. In 2018 ACM Multimedia Conference on Multimedia Conference (MM '18). ACM, New York, NY, USA, 1616-1624.

Network Protocols

HTTP/1.1

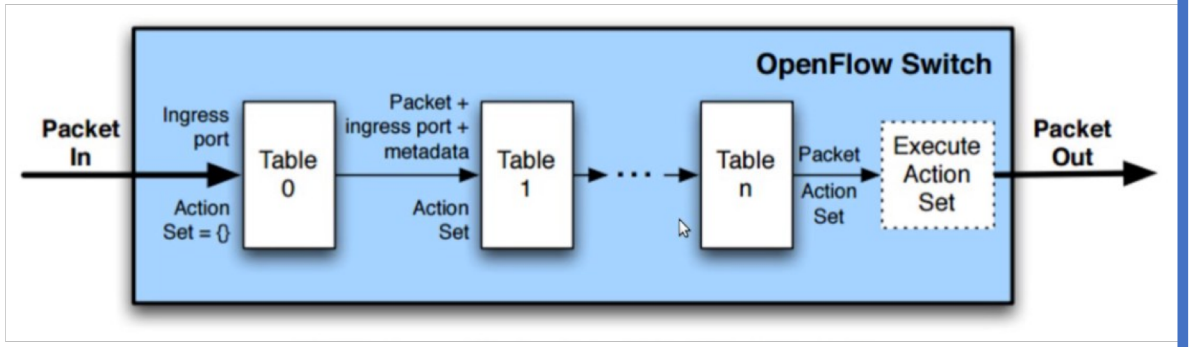


HTTP/2 (Streams)



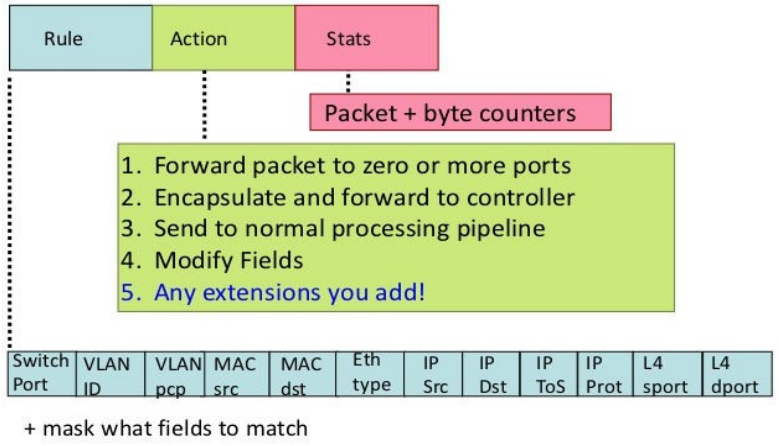
Software-Defined Networks

OpenFlow

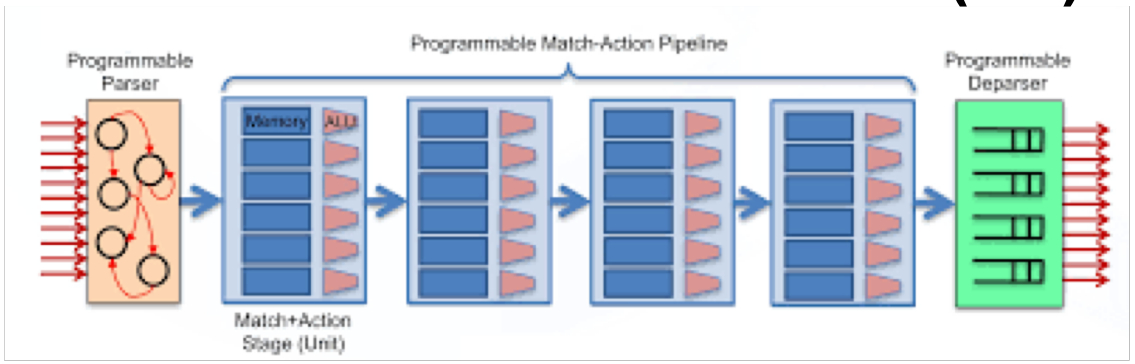


OpenFlow Basics

Flow Table Entries



Programmable Protocol Independent Packet Processors (P4)



HTTP/2 COMMON FRAME HEADER

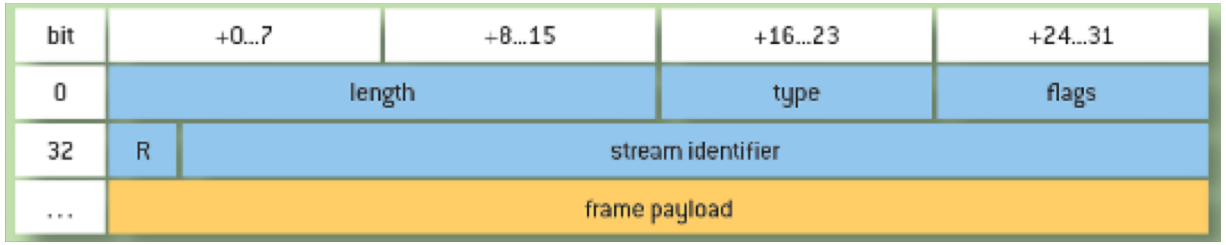
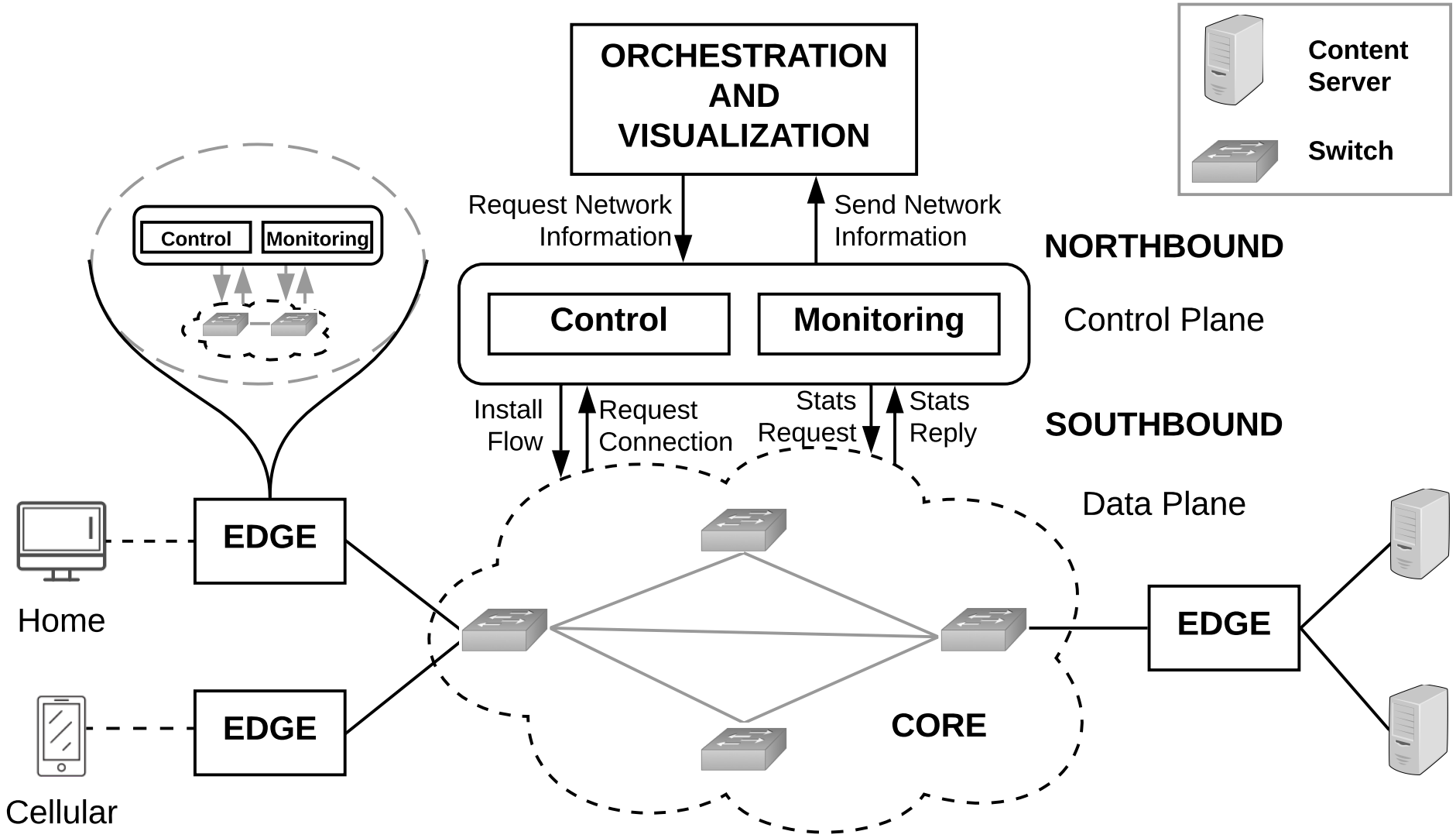


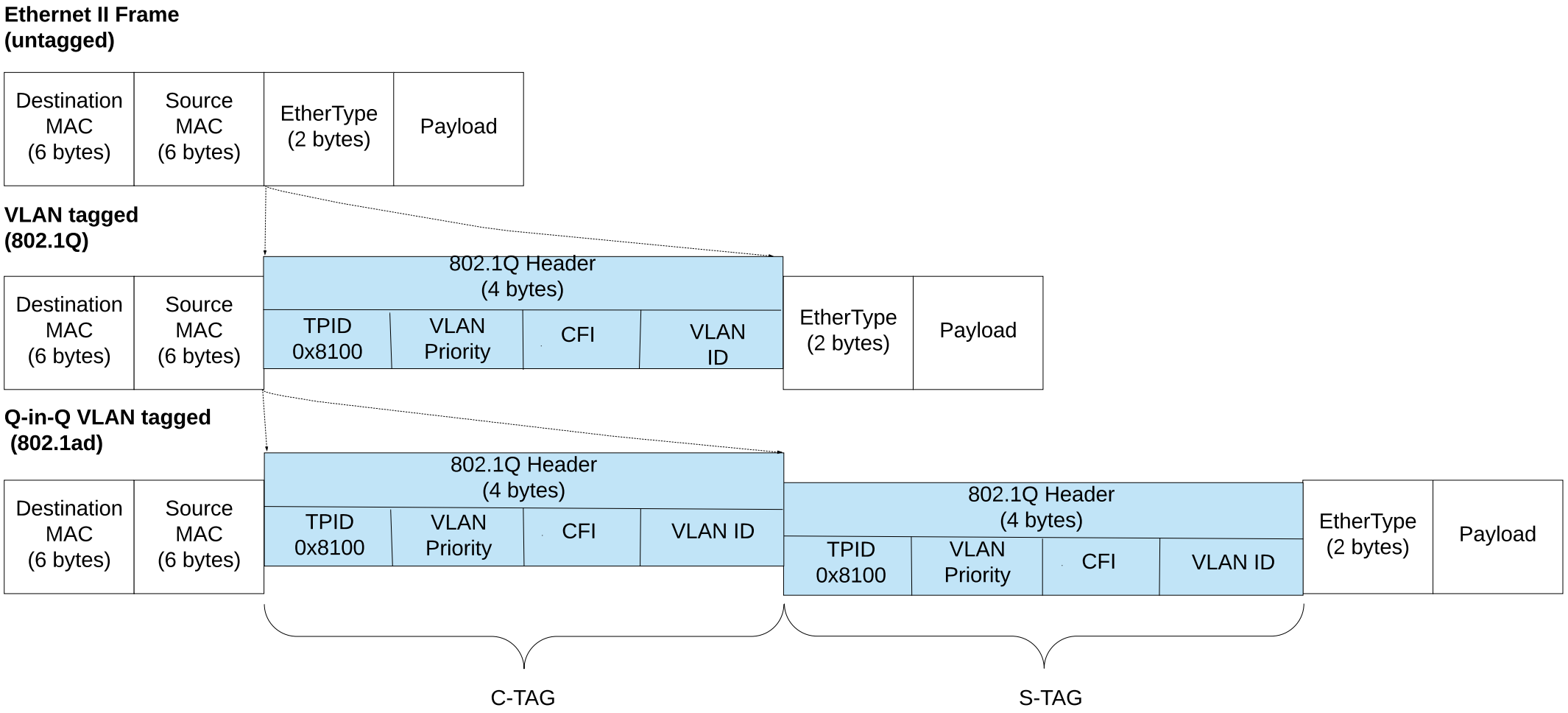
Image Courtesy:
https://queue.acm.org/detail.cfm?id=2555617
https://opencord.org/
https://www.opennetworking.org/

How can SDN be used to improve QoE of ABR Streaming?

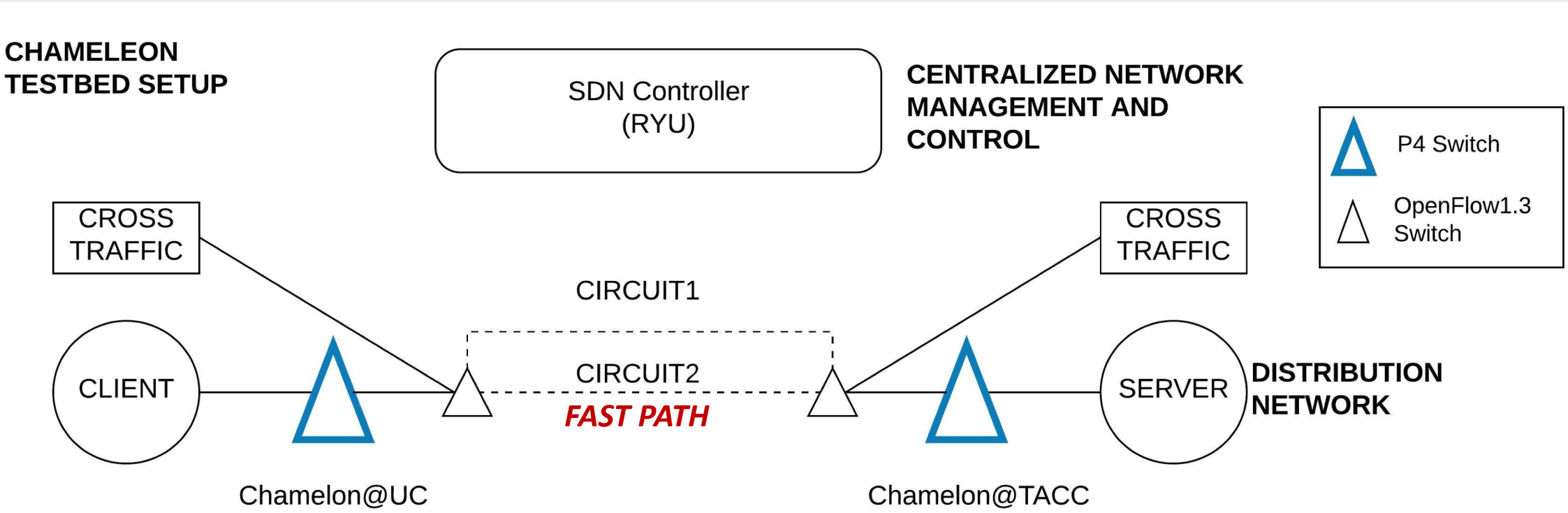
System Architecture



Q-in-Q VLAN tagging



Experimental Setup (BYOC)



Compute and Storage

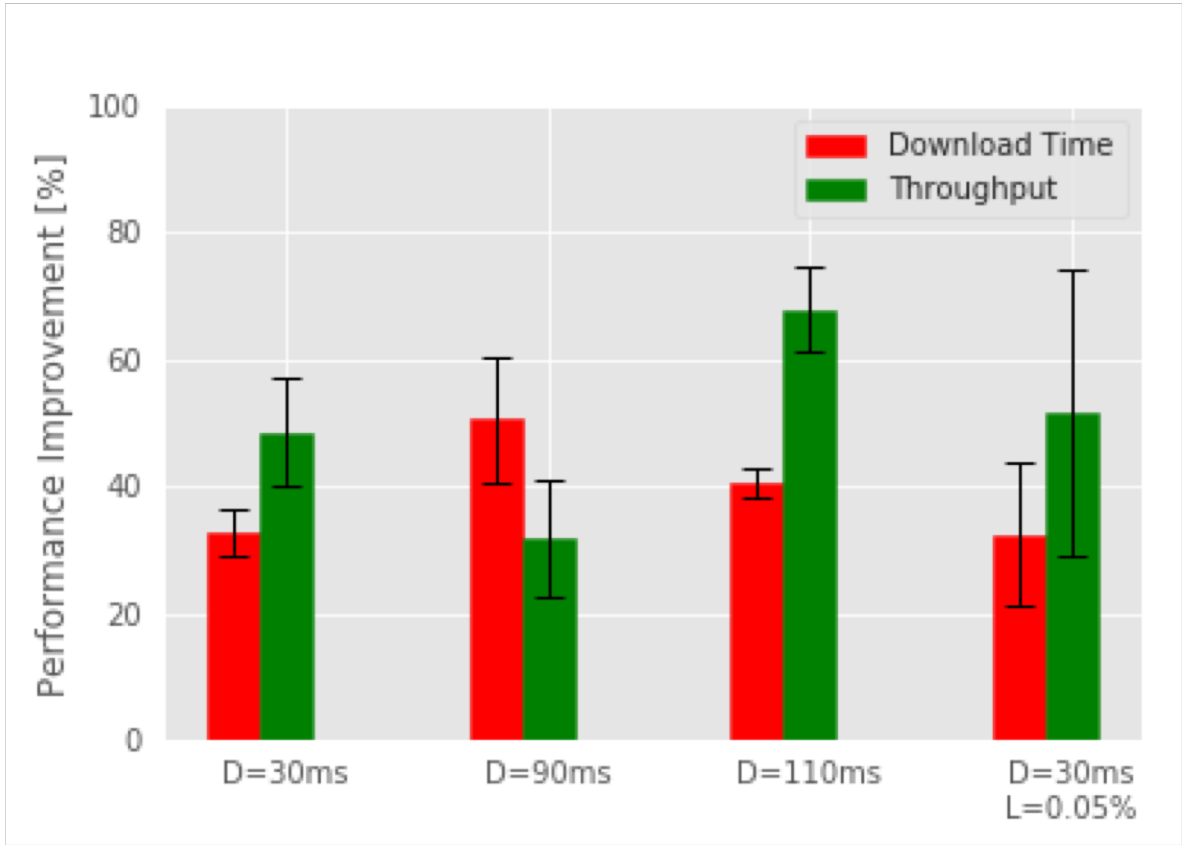
- All instances – bare-metal machines with Ubuntu 16.04
- Client – Python *hyper* library for HTTP/2
- Server – *Apache2*
- SDN Controller – *RYU*
- Cross Traffic – *Iperf3*

Networking

- Edge: P4 – Software switch *BMV2*
- Core: *Corsa* switches at TACC and UC
- Circuit1&2 – AL2S by Internet2 - 10Gigabit
- 802.1Q tag – HTTP/2 *Stream ID*

Fast Path for ABR Retransmissions

- Baseline – Single path (Circuit1) for original and retransmitted segment
- Performance Improvement over Baseline
- Loss drastically affects TCP in the form of retransmissions
- Fast path for retransmission can provide up to ~50% reduction in download time and ~70% improvement in average download rate



Conclusion

- Prototype of P4 at the edge and OpenFlow at the core (BYOC) – Two paths (Fast and Slow) between Austin, TX and Chicago, IL to provide better QoE
- Use of Fast Path for retransmissions significantly reduces segment transfer time and simultaneously increases download rate
- ABR Streaming clients can retransmit in higher quality over Fast Path to increase QoE
- https://www.chameleoncloud.org/blog/2018/09/20/application-based-qos-support-p4-and-openflow/

Future Scope

- For popular videos, retransmissions can be made once and reused by caching (CDNs)
- Extended to other applications such as GridFTP, QUIC (Quick UDP Interconnections), etc.

Repeatability with Jupyter

- JupyterLab - "*JupyterLab* is an interactive development environment for working with notebooks, code, and data."
- *IPython* kernels for executing notebooks
- Orchestrate and Visualize from single vantage point
- Chameleon and Jupyter
  1. Launch Docker container with JupyterLab in a Chameleon instance
  2. Create Notebook
  3. Run Experiment
  4. Visualize
  5. Repeat and Share
- Demo: 6:00pm today