Learning *in situ*: a randomized experiment in video streaming

https://puffer.stanford.edu

Francis Y. Yan, Hudson Ayers, Chenzhi Zhu[†], Sadjad Fouladi,
James Hong, Keyi Zhang, Philip Levis, Keith Winstein

Stanford University, [†]Tsinghua University

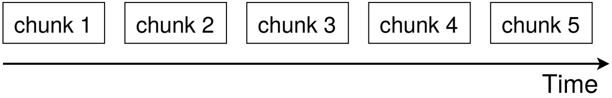
February 26, 2020

Introduction

- Video streaming dominates Internet traffic
- Adaptive bitrate (ABR) is a key algorithm to optimize users' quality of experience (QoE)
  - decides the quality level of each video chunk to send
  - primary goals: higher video quality, fewer stalls
  - prior work: BBA [SIGCOMM '14], MPC [SIGCOMM '15], CS2P [SIGCOMM '16], Pensieve [SIGCOMM '17], Oboe [SIGCOMM '18]

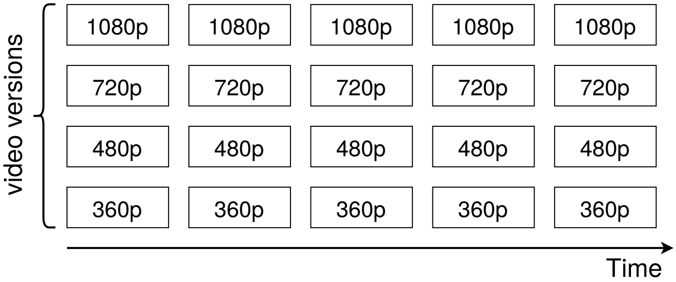
Introduction

- Video streaming dominates Internet traffic
- Adaptive bitrate (ABR) is a key algorithm to optimize users' quality of experience (QoE)
    - decides the quality level of each video chunk to send
    - primary goals: higher video quality, fewer stalls
    - prior work: BBA [SIGCOMM '14], MPC [SIGCOMM '15], CS2P [SIGCOMM '16], Pensieve [SIGCOMM '17], Oboe [SIGCOMM '18]



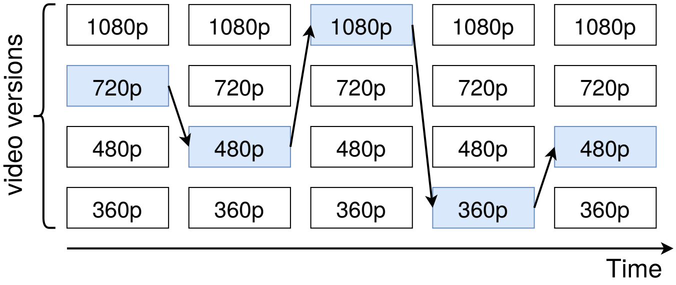
Introduction

- Video streaming dominates Internet traffic
- Adaptive bitrate (ABR) is a key algorithm to optimize users' quality of experience (QoE)
  - decides the quality level of each video chunk to send
  - primary goals: higher video quality, fewer stalls
  - prior work: BBA [SIGCOMM '14], MPC [SIGCOMM '15], CS2P [SIGCOMM '16], Pensieve [SIGCOMM '17], Oboe [SIGCOMM '18]



Introduction

- Video streaming dominates Internet traffic
- Adaptive bitrate (ABR) is a key algorithm to optimize users' quality of experience (QoE)
  - decides the quality level of each video chunk to send
  - primary goals: higher video quality, fewer stalls
  - prior work: BBA [SIGCOMM '14], MPC [SIGCOMM '15], CS2P [SIGCOMM '16], Pensieve [SIGCOMM '17], Oboe [SIGCOMM '18]



Our research study on ABR algorithms

What does it take to create a learned ABR algorithm that robustly performs well over the wild Internet?

- Over the last year, we have streamed 38.6 *years* of video to 63,508 distinct users

Takeaways

1. Confidence intervals in video streaming are *bigger than expected*
2. A simple ABR algorithm performs *better than expected*
3. Our way of outperforming existing schemes is *learning in situ*

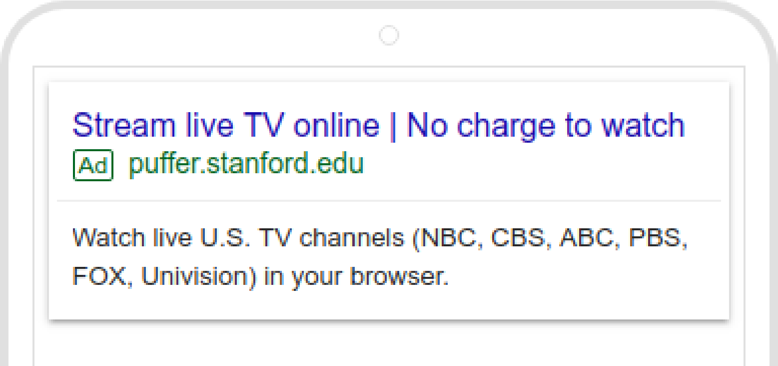
Takeaways

1. **Confidence intervals in video streaming are bigger than expected**
2. A simple ABR algorithm performs *better than expected*
3. Our way of outperforming existing schemes is *learning in situ*

Puffer: a live streaming platform running a randomized experiment

- Free live TV streaming website (puffer.stanford.edu)
- Opened to public December 2018
- More than 100,000 users today
- User sessions are randomized to different algorithms

Google ad for “tv streaming”



Press articles

Article | Talk | Read

WIKIPEDIA
The Free Encyclopedia

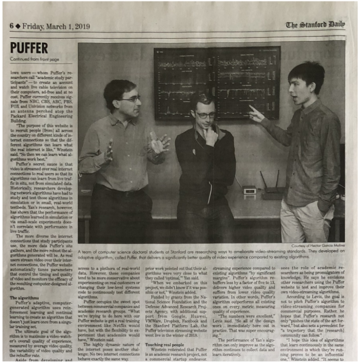
Puffer (TV service)

From Wikipedia, the free encyclopedia

Puffer is an American over-the-top internet television

Try 'Puffer': An Open-Source Free Live TV Streaming Service By Stanford

By Manisha Priyadarshini - January 18, 2019


6 ◆ Friday, March 1, 2019 — The Stanford Daily

PUFFER

Continued from front page

A team of computer science doctoral students at Stanford are researching ways to ameliorate video-streaming standards. They developed an adaptive algorithm, called Puffer, that delivers a significantly better quality of video experience compared to existing algorithms.

Hacker News new | past | comments | ask | show | jobs | submit | 2019-07-21

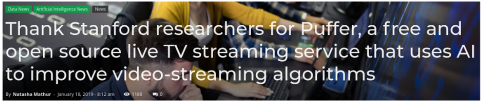
9. Puffer – Stream live TV in the browser (stanford.edu)
258 points by rowdyranga 3 months ago | hide | 51 comments

News & Analysis

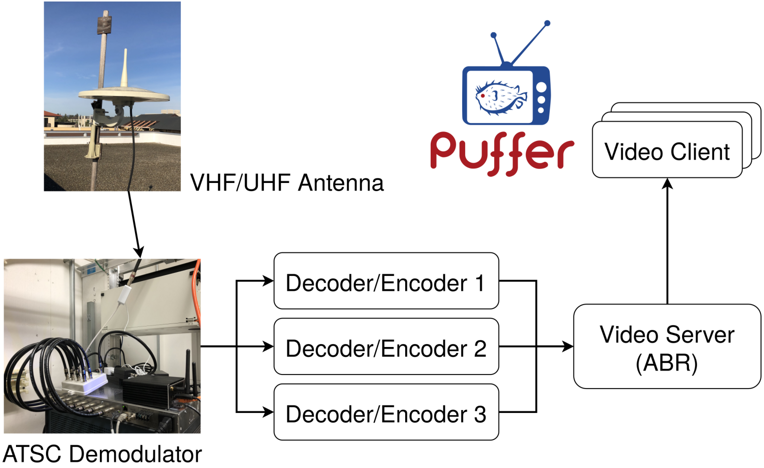
Stanford University Launches a Streaming TV Service (for Science)

A research team at Stanford University launched the Puffer streaming service in a bid to improve video streaming algorithms. While live, a computer will be taught how to design new algorithms to reduce glitches and stalls while improving image quality.

By Matthew Humphries January 18, 2019 10:13AM EST



Puffer architecture

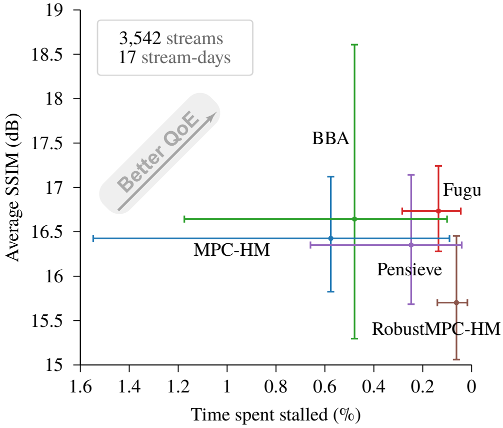


Confidence intervals in video streaming are bigger than expected

- Existing ABR algorithms found benefits like 10%–20% based on experiments lasting *hours* or *days* between *a few* network nodes
- We found: **2 years of data per scheme are needed to measure 20% precision**

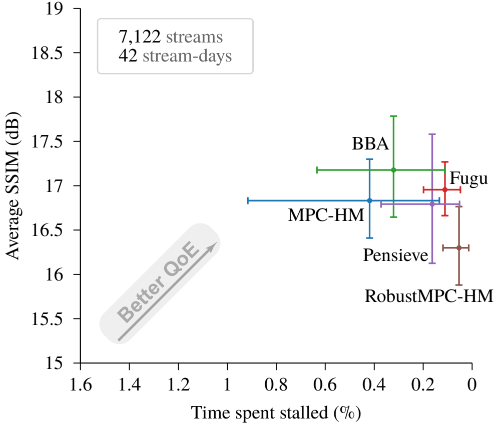
Confidence intervals in video streaming are bigger than expected

- Results on the *day* of Jan. 26, 2019, with 17 days of video streamed to 600 users



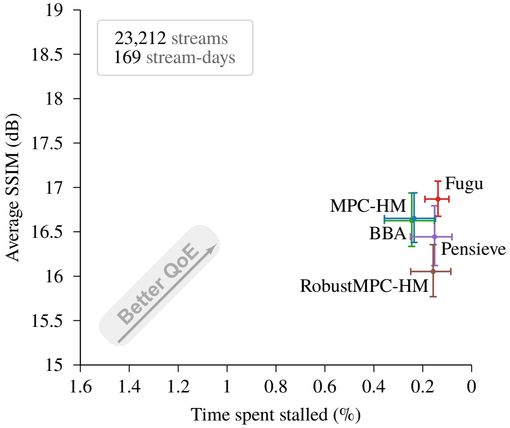
Confidence intervals in video streaming are bigger than expected

- Results in the *week* starting from Jan. 26, 2019, streaming 42 days of video



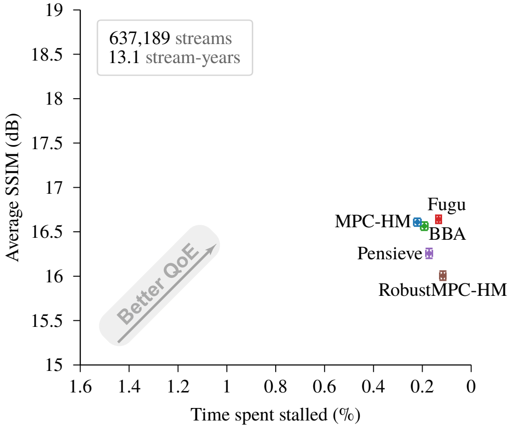
Confidence intervals in video streaming are bigger than expected

- Results in the *month* starting from Jan. 26, 2019, streaming 169 days of video



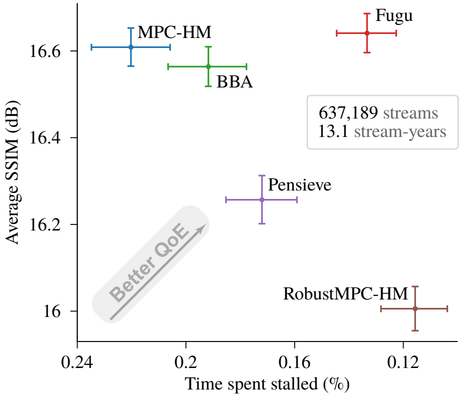
Confidence intervals in video streaming are bigger than expected

- Results in an *eight-month* period after Jan. 26, 2019, streaming > 13 years of video



Confidence intervals in video streaming are bigger than expected

- Need **2 years of video per scheme to reliably measure a 20% difference**
- Reason: Internet is way more noisy and heavy-tailed than we thought
  - Only 4% of the 637,189 streams had *any* stalls
  - Distributions of throughputs and watch times are highly skewed



Takeaways

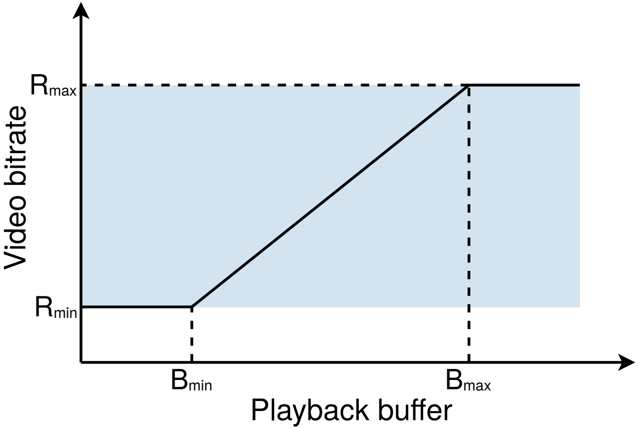
1. **Confidence intervals in video streaming are bigger than expected**
2. A simple ABR algorithm performs *better than expected*
3. Our way of outperforming existing schemes is *learning in situ*

Takeaways

1. Confidence intervals in video streaming are *bigger than expected*
2. **A simple ABR algorithm performs better than expected**
3. Our way of outperforming existing schemes is *learning in situ*

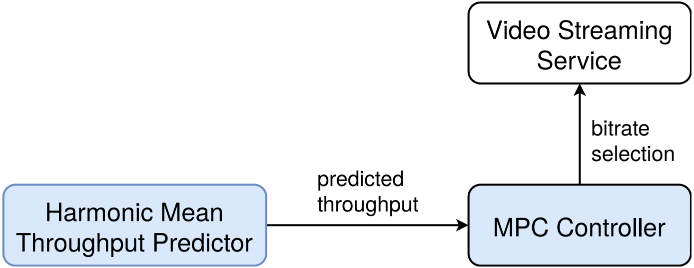
BBA [SIGCOMM '14]

- BBA is a simple buffer-based ABR algorithm



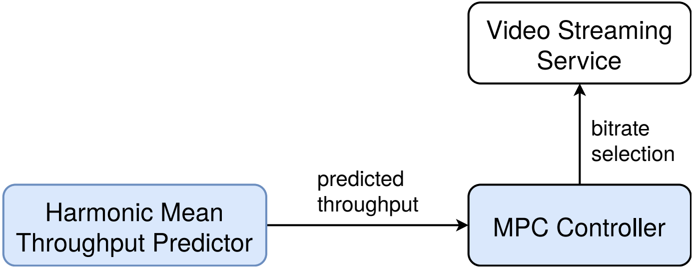
MPC-HM [SIGCOMM '15]

- MPC-HM predicts throughput using the harmonic mean (HM) of past throughputs
  - assumes throughput can be modeled with HM
  - assumes transmission time = predicted throughput × chunk size



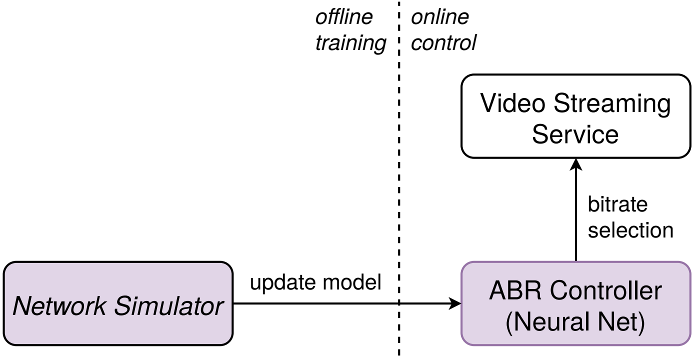
MPC-HM [SIGCOMM '15]

- MPC-HM predicts throughput using the harmonic mean (HM) of past throughputs
  - assumes throughput can be modeled with HM **?**
  - assumes transmission time $=$ predicted throughput $\times$ chunk size **?**



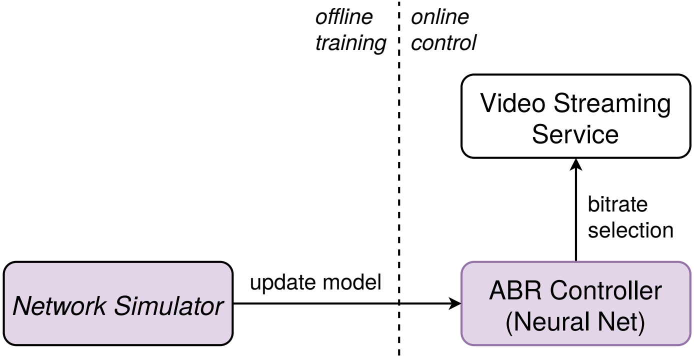
Pensieve [SIGCOMM '17]

- Pensieve learns an end-to-end ABR control
  - requires network simulators as training environments
  - assumes training in simulation generalizes to wild Internet

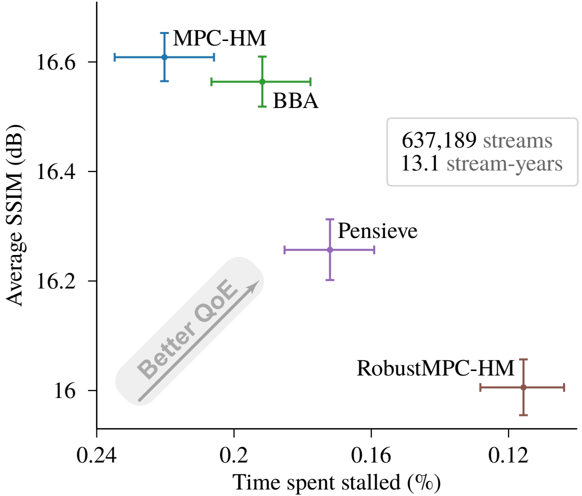


Pensieve [SIGCOMM '17]

- Pensieve learns an end-to-end ABR control
  - requires network simulators as training environments
  - assumes training in simulation generalizes to wild Internet **?**



SSIM vs stalls



Takeaways

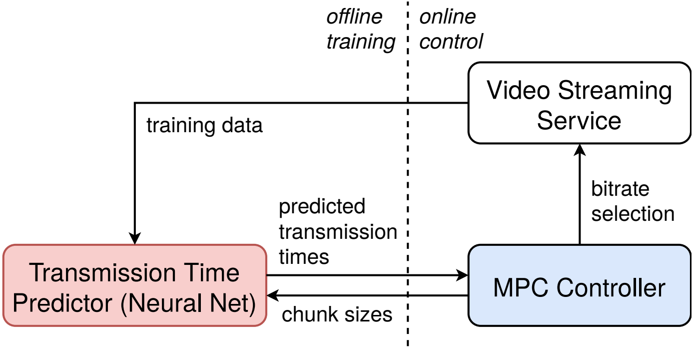
1. Confidence intervals in video streaming are *bigger than expected*
2. **A simple ABR algorithm performs better than expected**
3. Our way of outperforming existing schemes is *learning in situ*

Takeaways

1. Confidence intervals in video streaming are *bigger than expected*
2. A simple ABR algorithm performs *better than expected*
3. **Our way of outperforming existing schemes is learning in situ**

Fugu uses classical model predictive control

- Fugu replaces the throughput predictor in MPC-HM with a transmission time predictor



Fugu's transmission time predictor (TTP)

- Neural network predicts "how long would each chunk take?"
- Input:
  - sizes and transmission times of past chunks
  - **size of a chunk to be transmitted** (*not* a throughput predictor)
  - low-level TCP statistics (min RTT, RTT, CWND, packets in flight, delivery rate)
- Output:
  - probability distribution over transmission time (*not* a point estimate)

Fugu's transmission time predictor (TTP)

- Neural network predicts "how long would each chunk take?"
- Input:
  - sizes and transmission times of past chunks
  - **size of a chunk to be transmitted** (*not* a throughput predictor)
  - **low-level TCP statistics** (min RTT, RTT, CWND, packets in flight, delivery rate)
- Output:
  - probability distribution over transmission time (*not* a point estimate)

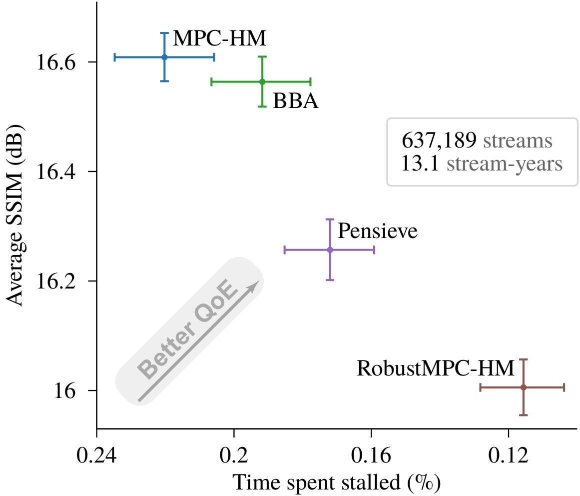
Fugu's transmission time predictor (TTP)

- Neural network predicts "how long would each chunk take?"
- Input:
  - sizes and transmission times of past chunks
  - **size of a chunk to be transmitted** (*not* a throughput predictor)
  - **low-level TCP statistics** (min RTT, RTT, CWND, packets in flight, delivery rate)
- Output:
  - **probability distribution over transmission time** (*not* a point estimate)

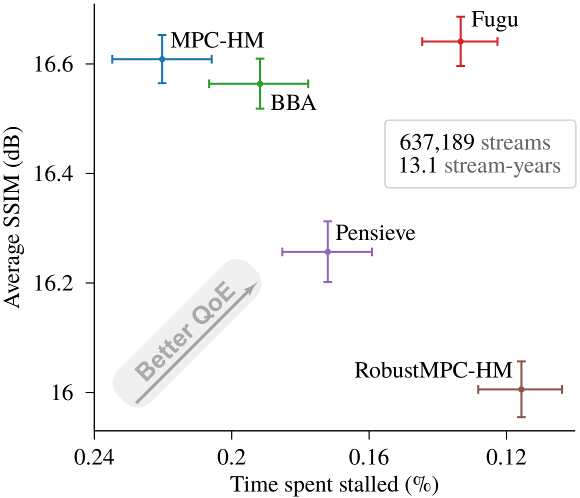
Learning TTP *in situ*

- Training: supervised learning *in situ* (in place) on real data from deployment environment
  - chunk-by-chunk series of each individual video stream
  - chunk i: size, timestamp sent, timestamp acknowledged, TCP statistics right before sending
- Learning *in situ* does **not** replay throughput traces or require network simulators
  - we don't know how to faithfully simulate the Internet

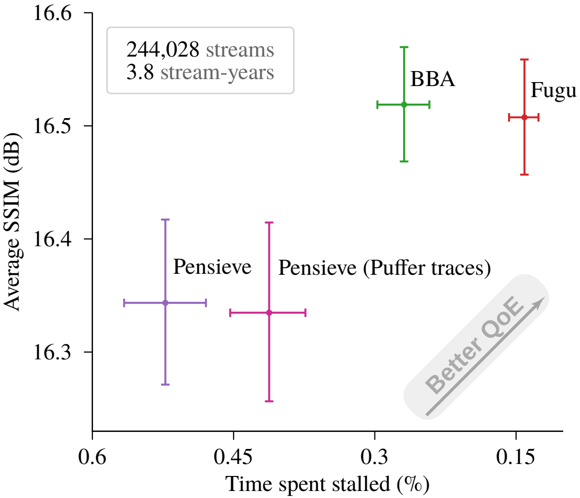
SSIM vs stalls



SSIM vs stalls



What happens if Pensieve is retrained on Puffer traces?



Takeaways

1. Confidence intervals in video streaming are *bigger than expected*
   - we need 2 years of data per scheme to measure 20% precision
2. A simple ABR algorithm performs *better than expected*
   - algorithms that make fewer assumptions are more general?
3. Our way of outperforming existing schemes is *learning in situ*
   - we don't know how to faithfully simulate the Internet

- Puffer (`puffer.stanford.edu`) is an open research platform for
  - ABR schemes, network and throughput prediction, congestion control

Thank You: Emily Marx, Puffer participants, NSF, Google, Huawei, VMware, Dropbox, Facebook, Stanford Platform Lab