Learning *in situ:* a randomized experiment in video streaming[†]

https://puffer.stanford.edu

Francis Y. Yan
Senior Researcher

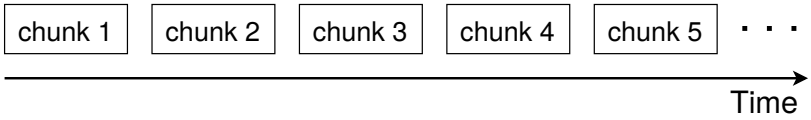
Microsoft Research

[†]This work was completed at Stanford University with Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, Emily Marx, Philip Levis, and Keith Winstein.

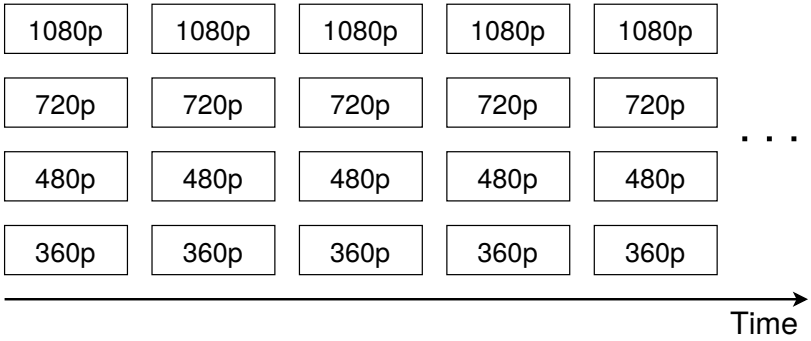
Introduction: adaptive bitrate (ABR) video streaming

- Video streaming dominates Internet traffic
- Adaptive bitrate (ABR) is a key algorithm to optimize quality of experience (QoE)
  - 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: adaptive bitrate (ABR) video streaming

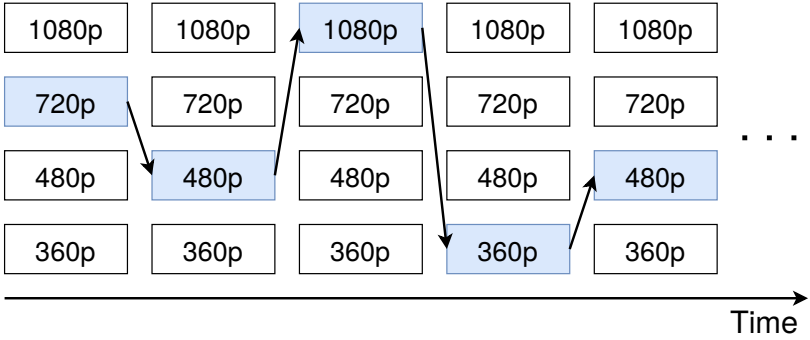


Introduction: adaptive bitrate (ABR) video streaming



Introduction: adaptive bitrate (ABR) video streaming

- ABR decides the quality level of each video chunk to optimize total QoE



Outline

1. **Puffer**: a live streaming platform for video streaming research
2. **Finding**: confidence intervals in video streaming are bigger than expected
3. **Fugu**: an ML-based ABR algorithm learned *in situ*

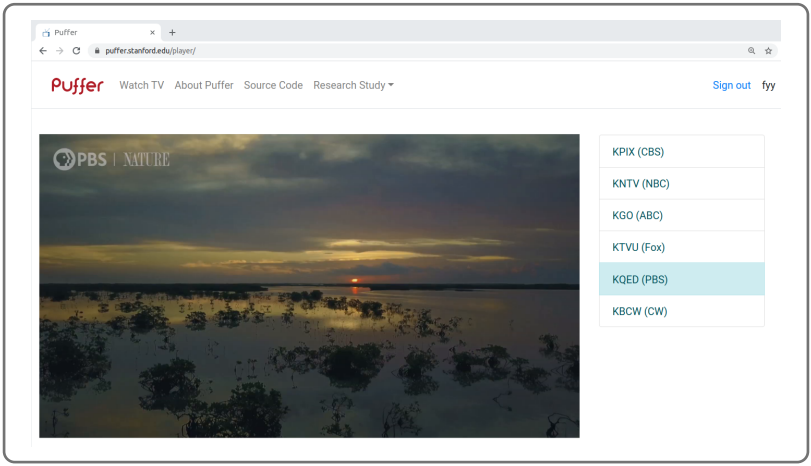
Outline

1. **Puffer**: a live streaming platform for video streaming research
2. **Finding**: confidence intervals in video streaming are bigger than expected
3. **Fugu**: an ML-based ABR algorithm learned *in situ*

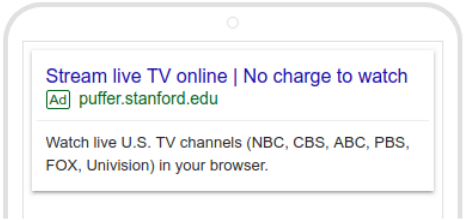
Puffer: a live streaming platform running a randomized experiment

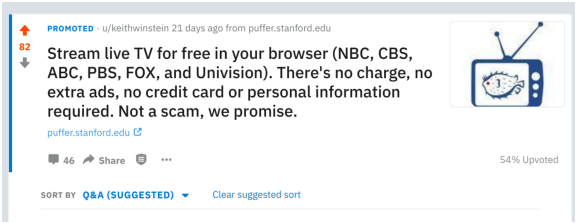
- Free live TV streaming website (puffer.stanford.edu)
- Opened to public December 2018
- User sessions are randomized to different algorithms
- Goal: realistic testbed and learning environment for video streaming research

Website: puffer.stanford.edu

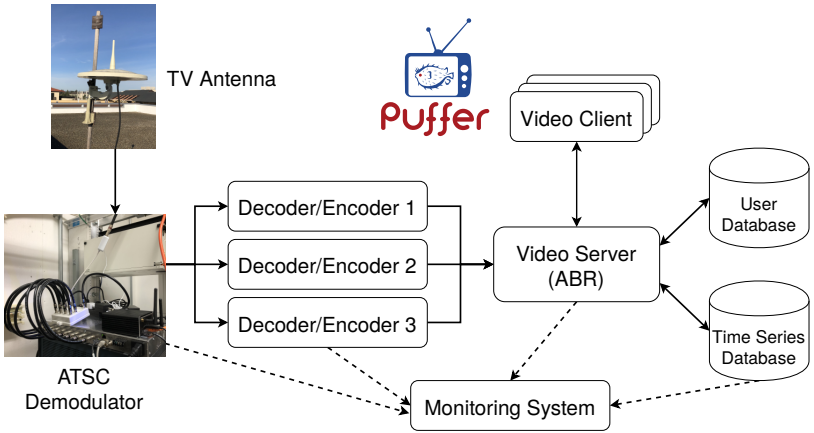


Ads for “live tv” and “tv streaming”





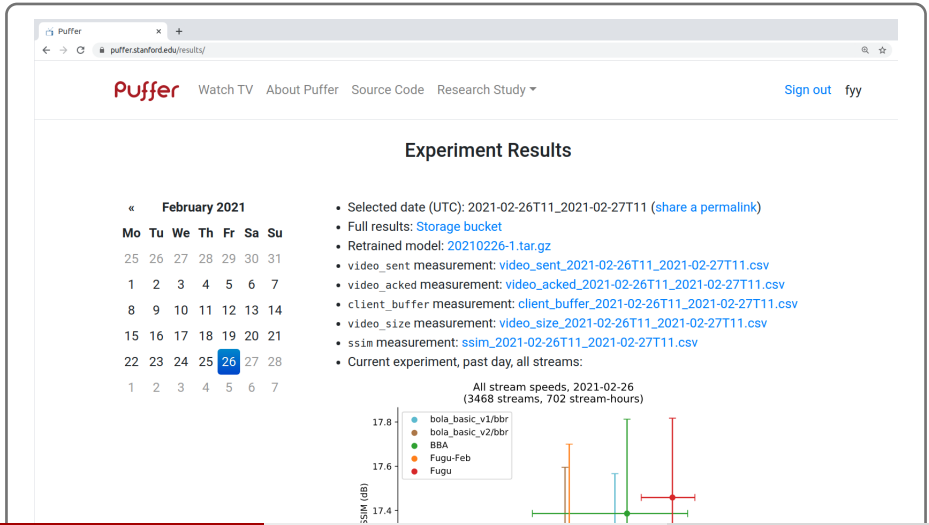
Puffer architecture



Puffer statistics

- 32,000 lines of code
  - 1,606 commits
  - 78,497++
  - 46,623--
- 130,000 real users
- 60 *years* of video streamed

Reproducible research and open platform



Outline

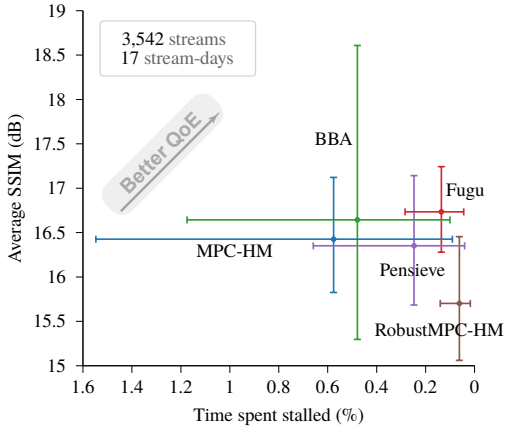
1. **Puffer**: a live streaming platform for video streaming research
2. **Finding**: confidence intervals in video streaming are bigger than expected
3. **Fugu**: an ML-based ABR algorithm learned *in situ*

Confidence intervals in video streaming are bigger than expected

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

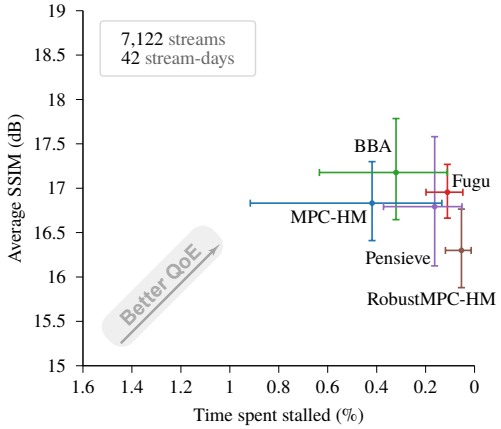
Confidence intervals in video streaming are bigger than expected

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



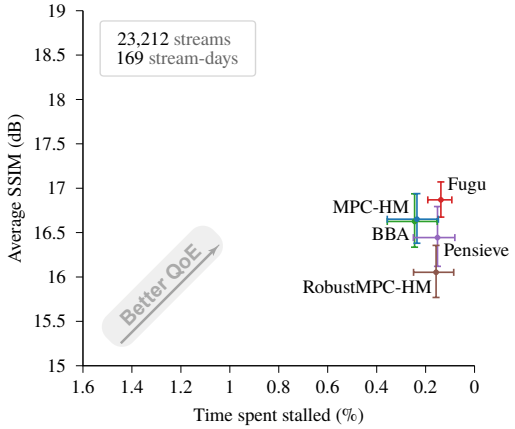
Confidence intervals in video streaming are bigger than expected

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



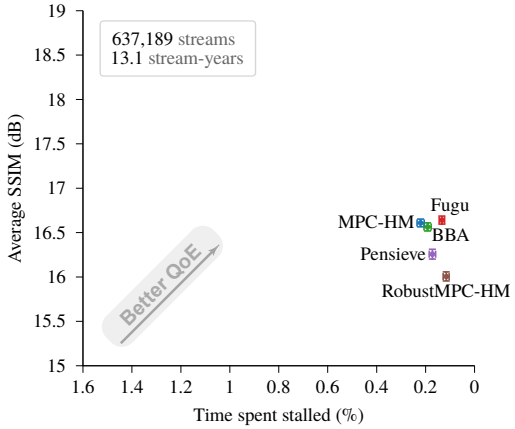
Confidence intervals in video streaming are bigger than expected

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



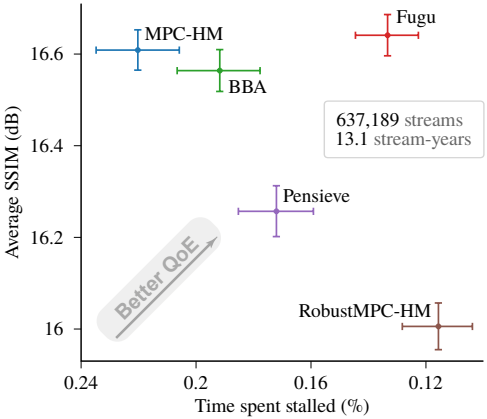
Confidence intervals in video streaming are bigger than expected

- 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

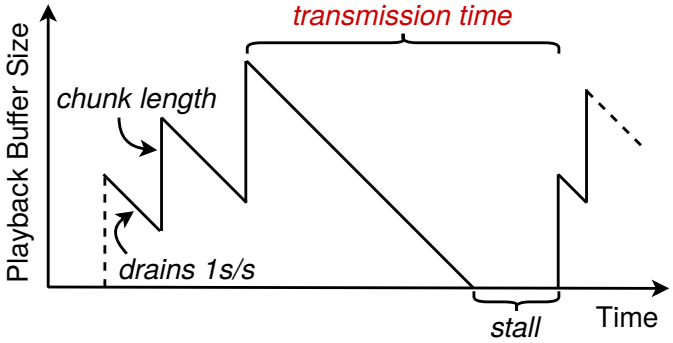


Outline

1. **Puffer**: a live streaming platform for video streaming research
2. **Finding**: confidence intervals in video streaming are bigger than expected
3. **Fugu**: an ML-based ABR algorithm learned *in situ*

System dynamics of ABR

- The only system uncertainty is *transmission time* of each chunk



Fugu's transmission time predictor (TTP)

- Neural network predicts "how long would each chunk take?"

Fugu's transmission time predictor (TTP)

- Neural network predicts "how long would each chunk take?"
- Input:
  - sizes and transmission times of past chunks

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)

Fact: observed throughput varies with file size

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)

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* (in place)

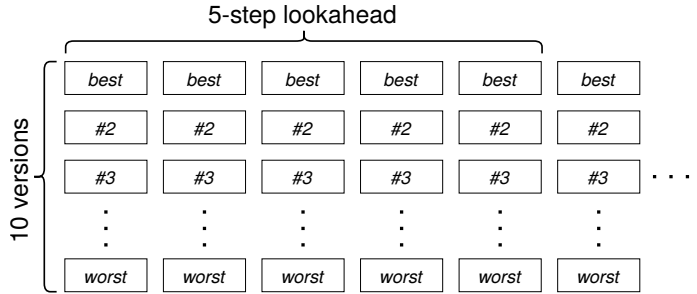
- Training: supervised learning *in situ* on real data from deployment environment
  - chunk-by-chunk series of each individual video stream
  - chunk i: size, timestamp sent, timestamp acknowledged, TCP statistics right before sending

Learning TTP *in situ* (in place)

- Training: supervised learning *in situ* 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

Fugu's model-based controller

- Objective function: expected sum of QoE in the lookahead horizon
- QoE: +video quality, −quality variation, −rebuffering



Fugu's model-based controller

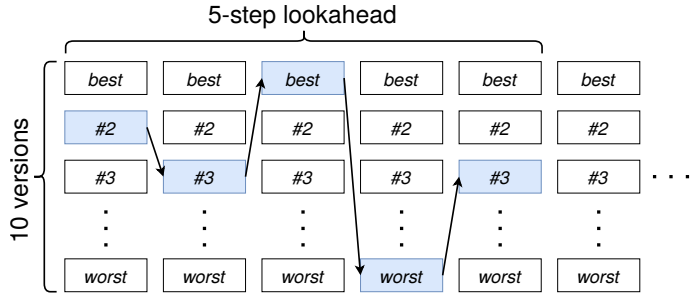
- Given TTP, optimal plan can be computed in real time

with dynamic programming

$$v_i^*(B_i, K_{i-1}) = \max_{K_i^s} \Big\{ \sum_{t_i} \Pr[\hat{T}(K_i^s) = t_i] \cdot \big(QoE(K_i^s, K_{i-1}) + v_{i+1}^*(B_{i+1}, K_i^s)\big) \Big\}$$

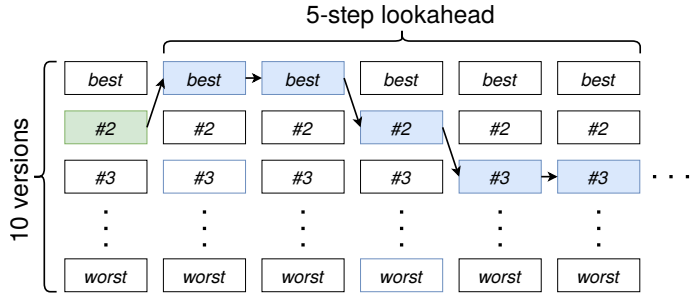
Fugu's model-based controller

- Replan at every step (model predictive control)
- Mitigate accumulation of errors

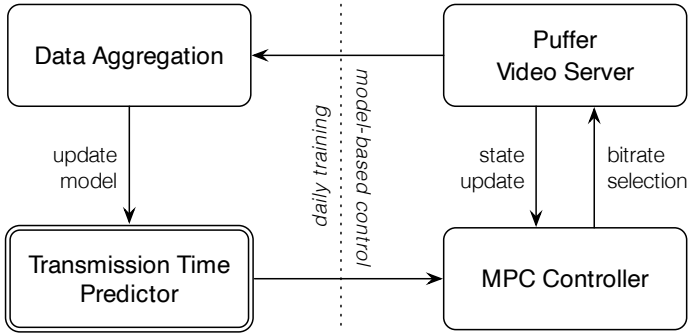


Fugu's model-based controller

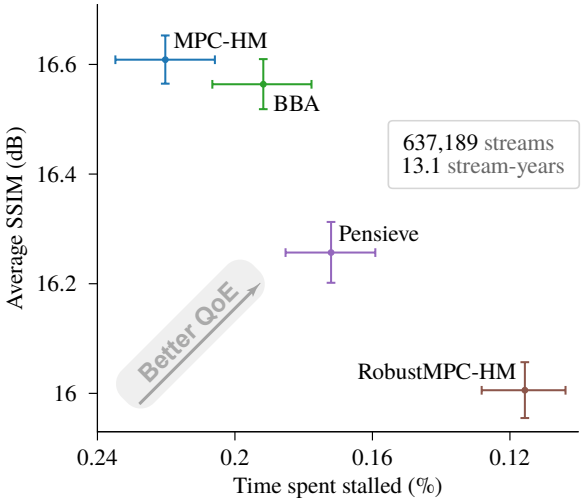
- Replan at every step (model predictive control)
- Mitigate accumulation of errors



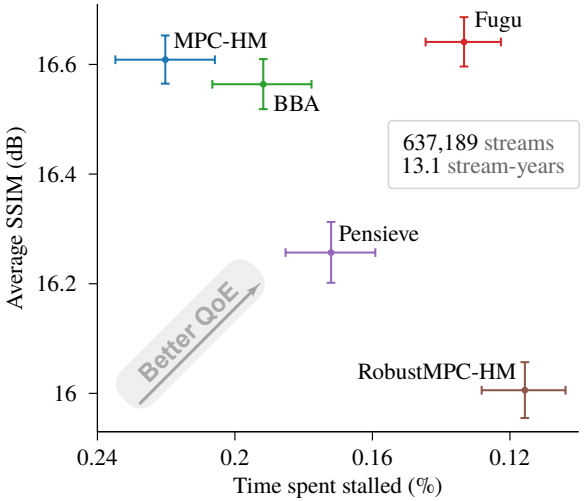
Fugu is a model-based reinforcement-learning algorithm



Evaluation: SSIM vs stalls



Evaluation: SSIM vs stalls

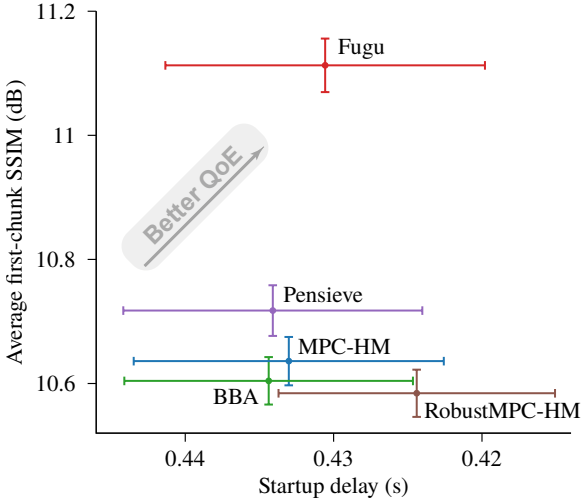


Evaluation: primary results

Results of primary experiment (Jan. 26–Aug. 7 & Aug. 30–Oct. 16, 2019)

Algorithm	Time stalled	Mean SSIM	SSIM variation	Mean duration
Fugu	0.13%	16.64 dB	0.74 dB	33.6 min
MPC-HM	0.22%	16.61 dB	0.79 dB	30.8 min
BBA	0.19%	16.56 dB	1.11 dB	32.1 min
Pensieve	0.17%	16.26 dB	1.05 dB	31.6 min
RobustMPC-HM	0.12%	16.01 dB	0.98 dB	31.0 min

Evaluation: cold-start performance



Takeaways

1. **Puffer**: a video streaming platform opened to research community
   - 130,000+ real users, streamed 60+ years of video
2. **Finding**: confidence intervals in video streaming are bigger than expected
   - we need 2 years of data per scheme to measure 20% precision
3. **Fugu**: an ML-based ABR algorithm learned *in situ*
   - Transmission Time Predictor (TTP)

Francis Y. Yan <francisy@microsoft.com>