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
• 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]
• 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”

Stream live TV online | No charge to watch
Ad puffer.stanford.edu

Watch live U.S. TV channels (NBC, CBS, ABC, PBS, FOX, Univision) in your browser.
Press articles

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

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

9. a Puffer – Stream live TV in the browser (stanford.edu)
258 points by rowdyranga 3 months ago | hide | 91 comments

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

Thank Stanford researchers for Puffer, a free and open source live TV streaming service that uses AI to improve video-streaming algorithms

By Santana Yan January 17, 2019 11:30 AM
Puffer architecture

VHF/UHF Antenna

ATSC Demodulator

Decoder/Encoder 1

Decoder/Encoder 2

Decoder/Encoder 3

Video Server (ABR)

Video Client
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 × 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

Diagram:

- Network Simulator
- ABR Controller (Neural Net)
- Video Streaming Service
- bitrate selection

Flow:

- offline training
- online control
- update model
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

<table>
<thead>
<tr>
<th>SSIM (dB)</th>
<th>Time spent stalled (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.12</td>
</tr>
<tr>
<td>16.2</td>
<td>0.16</td>
</tr>
<tr>
<td>16.4</td>
<td>0.2</td>
</tr>
<tr>
<td>16.6</td>
<td>0.24</td>
</tr>
</tbody>
</table>

Average SSIM (dB)

<table>
<thead>
<tr>
<th>Method</th>
<th>Streams</th>
<th>Stream-Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBA</td>
<td>637,189</td>
<td>13.1</td>
</tr>
<tr>
<td>MPC-HM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pensieve</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RobustMPC-HM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Better QoE

Francis Y. Yan (Stanford)
February 26, 2020
Takeaways

1. Confidence intervals in video streaming are \textit{bigger than expected}

2. \textbf{A simple ABR algorithm performs better than expected}

3. Our way of outperforming existing schemes is \textit{learning in situ}
Takeaways

1. Confidence intervals in video streaming are \textit{bigger than expected}.

2. A simple ABR algorithm performs \textit{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 Diagram]
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 \textit{(not a throughput predictor)}
  - low-level TCP statistics (min RTT, RTT, CWND, packets in flight, delivery rate)
- Output:
  - probability distribution over transmission time \textit{(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

Average SSIM (dB) vs Time spent stalled (%)

- BBA
- MPC-HM
- Pensieve
- RobustMPC-HM

637,189 streams
13.1 stream-years

Better QoE
SSIM vs stalls

Average SSIM (dB) vs Time spent stalled (%)

- BBA
- MPC-HM
- Pensieve
- Fugu
- RobustMPC-HM

Better QoE

637,189 streams
13.1 stream-years
What happens if Pensieve is retrained on Puffer traces?

244,028 streams
3.8 stream-years

Better QoE
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](http://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*