Swift: Adaptive Video Streaming with Layered Neural Codecs

Mallesham Dasari, Kumara Kahatapitiya,
Samir R. Das, Aruna Balasubramanian, Dimitris Samaras

NSDI 2022

Stony Brook University

Carnegie Mellon University
Outline

- Background & Motivation
  - Adaptive Video Streaming & Today’s Practice
  - Existing Alternatives: Layered Streaming & Challenges
- Swift
  - Layered Neural Video Coding
  - Encoder, Decoder, Streaming Protocol
  - Evaluation & Comparison with Existing Work
- Conclusions
Video Streaming Today

Fundamental problems: 1) Limited throughput, 2) Variability

Bitrate Selection: A Decade-long Research!
(e.g., Festive [CoNEXT’12], MPC [SIGCOMM’15], BOLA [INFOCOM’16], Pensieve [SIGCOMM’17], Fugu [NSDI’20])
Limitations of Today’s Streaming Algorithms

Throughput trace is from ACM/IEEE TON’2020
Limitations of Today’s Streaming Algorithms

Slow Reaction 75sec

- Throughput
- BOLA

Bitrate (Mbps)

Time (sec)
Limitations of Today’s Streaming Algorithms

- Fast Reaction: 25 sec
- Discard poor quality chunks: 75 sec

Graph showing throughput comparison between Throughput, BOLA-FS, and BOLA over time.
Limitations of Today’s Streaming Algorithms

![Graph showing bitrate and time with annotations: Fast Reaction, 75sec, 25sec, Discard poor quality chunks, Bandwidth waste.]
Limitations of Today’s Streaming Algorithms

- Bandwidth Efficient
- Slow Reaction -> Poor QoE
  - BOLA, Pensieve, MPC

- Bandwidth Inefficient
- Fast Reaction -> Better QoE
  - BOLA-FS
Existing Alternative: Layered Video Streaming

- **Layered Coding**
- **A well-suited technique for streaming under variable network conditions**

Layered Video Streaming:

- **Layer 1 (c₀):** 352x288 (1Mbps)
- **Layer 2 (c₁):** 640x480 (1Mbps)
- **Layer L:** 4K/8K (1Mbps)

Classical/Regular Coding:

- **25Mbps:** 4K/8K
- **2Mbps:** 640x480
- **1Mbps:** 352x288

Layered Coding is well-suited for streaming under variable network conditions.
Existing Alternative: Layered Video Streaming

Challenges of using traditional layered coding in practice

- Cross Layer Compression Overhead
- High Coding Latency

Graphs showing PSNR vs. Bits Per Pixel and Decoding Latency vs. Number of Layers.
Existing Alternative: Layered Video Streaming

Challenges of using traditional layered coding in practice

- Cross Layer Compression Overhead
- High Coding Latency

Remains an open problem & Not used in practice!
Swift: Layered Coding Using Neural Codecs

e.g., AutoEncoder
Swift: Layered Coding Using Neural Codecs

Swift’s Components: 1) Encoder, 2) Decoder, 3) Streamer
1) Layered Neural Video Encoder

Residual codes (Toderici et al [CVPR’16], VCII [ECCV’18], HLVC [CVPR’19])

Layered codes transmitted over the Internet
2) Layered Neural Video Decoder

Challenge 1. Iterative Decoding is Slow
- Need real-time decoding for playback

Single-shot Decoder
2) Layered Neural Video Decoder

Challenge 1. Iterative Decoding is Slow
- Need real-time decoding for playback

Single-shot Decoder

Key Takeaway
Latency is now independent of layers
2) Layered Neural Video Decoder

Challenge 2. Variable Compute Capacity
- Software-driven
- Need to scale well with other applications
2) Layered Neural Video Decoder

Challenge 2. Variable Compute Capacity
- Software-driven
- Need to scale well with other applications

Multiple Exits
2) Layered Neural Video Decoder

Challenge 2. Variable Compute Capacity
- Software-driven
- Need to scale well with other applications

Video quality is a function of variable compute and network capacity
3) Streaming Protocol/Algorithm (ABR)

- **Question**
  - When and How to Enhance Buffered Low-Quality Segment?

- **Objective**
  - Maximize QoE e.g., $F(\text{Quality, Stalls, Smoothness})$

Network history
Buffer occupancy
Download qualities
Compute history
Decoded qualities
Playback times
Quality matrix for next segment

**ABR**

Enhance or fresh download?
How many layers?
Putting Everything Together: Swift
Swift’s Performance Comparison

- **Compression Baselines**
  - H.264
  - SHVC

- **Streaming Baselines**
  - BOLA [INFOCOM’16]
  - BOLA-FS [MMSys’18]
  - Pensieve [SIGCOMM’17]
  - Grad [MM’20]

- **Swift Evaluation**
  - QoE (Quality, Stalls, Temporal Smoothness)
  - Bandwidth Utilization
  - Reaction time
  - Compression Efficiency & Latency
Swift’s Performance

The chart illustrates the performance of Swift in terms of bitrate (Mbps) over time (sec). It shows that Swift reacts faster compared to other implementations such as Grad, Pensieve, BOLA-FS, and BOLA. The y-axis represents bitrate ranging from 0 to 30 Mbps, and the x-axis represents time in seconds from 0 to 250.
Swift’s Performance

Swift Reacts Faster

Bitrate (Mbps)

Time (sec)

0 50 100 150 200 250

0 10 20 30

Swift Reacts Faster

Normalized Bandwidth Usage

CDF

0.0 0.2 0.4 0.6 0.8 1.0

Swift
Grad
BOLA-FS
Pensieve
BOLA
Swift’s Performance

Swift’s faster reaction improves QoE by 45% while saving the bandwidth by 18%
Summary

- Layered coding provides fine-grained rate adaptation but traditionally it is nontrivial to devise an algorithm.

- Neural video codecs are becoming mainstream and can efficiently realize layered coding on GPU-equipped devices.

- We present Swift, an adaptive video streaming system using layered neural coding, that includes an encoder, a decoder, and a streaming protocol.
Thank you!

Questions? malleshd@andrew.cmu.edu