FlexStream: Towards Flexible Adaptive Video Streaming on End Devices using Extreme SDN

Ibrahim Ben Mustafa
Old Dominion University
iben@cs.odu.edu

Tamer Nadeem
Virginia Commonwealth University
tnadeem@vcu.edu

Emir Halepovic
AT&T Labs - Research
emir@research.att.com

ACM Multimedia Conference, Seoul, Korea, 22-Oct-2018

All presented information does neither reflect nor imply an actual business case for AT&T and associated companies. They are generalized data used to assess performance of the algorithms only.

© 2018 AT&T Intellectual Property. All rights reserved. AT&T, the AT&T logo and all other AT&T marks contained herein are trademarks of AT&T Intellectual Property and/or AT&T affiliated companies. The information contained herein is not an offer, commitment, representation or warranty by AT&T and is subject to change.
Mobile Video Traffic

• Witnessing tremendous growth in mobile data traffic.
• Mobile data traffic is predicted to increase seven fold between 2016 and 2021
  • Mobile Video would be responsible for more than %75 by 2020. (*)

(*) Cisco Visual Networking Index - 2017
HTTP Adaptive Streaming (HAS)

HTTP Server
with video content in different qualities (e.g. Apache, IIS or HTTP CDN)

Network
with variable Bandwidth (Internet)

Heterogeneous Devices
requesting the right quality for smooth playback and quick start, no special server logic needed

Images retrieved from: https://bitmovin.com/
Problem & Related Work
Performance Issues with HAS

• When HAS players compete over the bottleneck:
  • Instability in the quality
  • Playback stalls
  • Unfairness
  • Long startup delay

• Root cause: ON/OFF traffic pattern (*)

Existing Solutions

Client-based solutions (improving players’ adaptation algorithm)

Server/proxy-based solutions (throttling or bitrate control)

Edge-based solutions (traffic management and control)
Issues with existing Solutions

• Existing solutions are either:

  1. Not effective, since they can not:
     • Address the main performance issues.
     • Comply with network policies.

  2. Invasive: Players have to follow specific adaptation logic.

  3. Not generic: Specific for HAS.


  5. Infeasible (in practice):
     • Requires CDN edge server changes.
     • Require player feedback and interactions.
Our Solution: FlexStream

• SDN-based framework that leverages:

  • **Centralized/edge component:**
    • Enables global view of network condition.
    • Context-aware through end device feedback.
    • Specifies a policy controlling resource allocation, using an optimization function.

  • **Distributed SDN component:**
    • Monitors and reports various context information.
    • Implements network policies.
    • Offloads fine-grained functionality to the end device.
FlexStream Benefits

• Offloads intrusive or resource-demanding tasks from the network to end devices.

• Allows for fine-grained and intelligent management of bandwidth based on real time context awareness and specified policy.

• Flexible implementation of network policies.

• Improves video QoE:
  • Reduces quality switching by 81%, stalls by 92%, and startup delay by 44%.

• Offers universal approach to work across network technologies, WiFi and cellular.

• Has no dependency on the internal network support.
System Overview and Architecture
Utilizing SDN on End Device (extreme SDN)

Resource management (cloud or wireless network infrastructure)

Applying network policies using SDN components on an end device.
FlexStream – Overview

1. **Data Collection**
   - Flow statistics (OVS), device context, user context

2. **Control & Management**
   - Resource allocation, admin policies

3. **Policy Implementation**
   - Fine-grained control over data flows

- **Device Agent**
- **Video Player**
- **SDN**
- **User Preferences**
- **Control Channel**
  - Video meta-data, device context
  - Policy rules

- **Data Channel**
  - Video traffic

- **FlexStream Controller**
- **Media server**

© 2018 AT&T Intellectual Property. All rights reserved. AT&T, the AT&T logo and all other AT&T marks contained herein are trademarks of AT&T Intellectual Property and/or AT&T affiliated companies. The information contained herein is not an offer, commitment, representation or warranty by AT&T and is subject to change.
FlexStream – Architecture

**End Device**
- QoE Monitor
- Policy Engine
- Context Monitor
- Rate Handler
- HTTP Inspector
- Bitrate Estimator
- HAS Client
- Local Controller
- SB Interface
  - netstat
  - log
  - Sensor Manager
  - Kernel Space
  - WiFi/LTE Driver

**FlexStream Controller**
- Network Monitor
- Optimization Module
- Policy Manager
- Admin Interface
  - Stream Table
  - Policy Table

**Media Server**
- HTTP Server
- Media Presentation Description (MPD)
- Media Segments

**Context data and policies**

© 2018 AT&T Intellectual Property. All rights reserved. AT&T, the AT&T logo and all other AT&T marks contained herein are trademarks of AT&T Intellectual Property and/or AT&T affiliated companies. The information contained herein is not an offer, commitment, representation or warranty by AT&T and is subject to change.
FlexStream – Context-Awareness

• Supports various management policies based on the different contexts for:
  
  • Fair and balanced watching experience.
  
  • Maximizing videos bitrates.
  
  • Better bandwidth utilization.
Optimization Problem

\[
\max_{x_{ij}} \sum_{i=1}^{N} \sum_{j=1}^{K_l} (u_{ij} - \mu \delta_{ij}) x_{ij} \\
\text{subject to} \sum_{i=1}^{N} \sum_{j=1}^{K_l} (\epsilon r_{ij}) x_{ij} \leq B \\
\sum_{j=1}^{K_l} x_{ij} = 1, x_{ij} \in \{0, 1\} \forall i
\]

Utility Function

\[
u_{ij} = \prod_{l=1}^{a} \beta_{il} \cdot \log(r_{ij})
\]

Penalty Function

\[
\delta_{ij} = \begin{cases} 
|r_{ij} - r_{ic}|s_i + (m - \left\lfloor \frac{t_t}{k} \right\rfloor), & t < t_{thresh} \\
|r_{ij} - r_{ic}|s_i, & t \geq t_{thresh}
\end{cases}
\]
Implementation Challenges

- Extending SDN planes to enable controlling the data rate on the end device.

OVS & Open Flow protocol are extended to control and limit data rate through TCP flow control mechanism.

Open vSwitch flow table

- Forward the packet to an output port
- Forward to SDN controller
- Drop the packet
- Mirror the packet
- Modify packet headers.

OVS does not support data rate control
Implementation Challenges

• It is possible to bind OVS to Wi-Fi interface, but not to the cellular interface:
  • Uses different technologies and protocols to connect to its base station.
  • Moving the IP address of the cellular interface to OVS immediately breaks the connection with the base station.

• Typical solution?
  • Using a Wi-Fi access point as a mediator, but does not allow for direct experimentation.

• FlexStream?
  • Installing a number of rules to the OVS flow table to rewrite the source/destination IP and MAC addresses with OVS addresses to force all traffic to pass through OVS.
Evaluation
Evaluation

• **Quality Metrics:** Stability, fairness, stalls, and startup latency.

• **Scenarios:** Static Bandwidth and Dynamic Bandwidth

• **Experiments**
  - Basic: 3 real players in a real network.
  - Extended: 12 emulated players & server, real network.

• **Context:** User priority, screen size, link condition, background traffic, and surrounding luminance.

• **Overheads:** Computation and bandwidth.
Setup for Basic Experiments

FlexStream controller (Ubuntu 14.04)

Proxy server (Squid v3.1)
TC Linux to limit data rate

Public HTTP server

Cell tower

Big buck bunny
9:00 min

<table>
<thead>
<tr>
<th>Encoded Frame size</th>
<th>Frame Rate</th>
<th>Approx Bitrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>512x288</td>
<td>25</td>
<td>449kbps</td>
</tr>
<tr>
<td>704x396</td>
<td>25</td>
<td>843kbps</td>
</tr>
<tr>
<td>896x504</td>
<td>25</td>
<td>1416kbps</td>
</tr>
<tr>
<td>1280x720</td>
<td>25</td>
<td>2656kbps</td>
</tr>
</tbody>
</table>

Internet

Nexus 7 (7”)

Nexus 4 (4.8”)

WiFi AP (Ubuntu 12.04)

OVS (v1.9)
OpenFlow (v1.2)
OVS-VSCTL(v1.9)
OVS-OFCTL(v1.9)
GPAC(v0.6.2-DEV)
Device Agent
Setup for Extended Experiments

Dummy video segments equivalent in size and distribution to those used in the basic experiment

<table>
<thead>
<tr>
<th>Encoded Frame size</th>
<th>Frame Rate</th>
<th>Approx Bitrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>512x288</td>
<td>25</td>
<td>449 kbps</td>
</tr>
<tr>
<td>704x396</td>
<td>25</td>
<td>843 kbps</td>
</tr>
<tr>
<td>896x504</td>
<td>25</td>
<td>1416 kbps</td>
</tr>
<tr>
<td>1280x720</td>
<td>25</td>
<td>2656 kbps</td>
</tr>
</tbody>
</table>
Basic Experiments

Experiments with different network capacities, starting from 2500 Kbps to 8500 Kbps with an increase of 1500 Kbps.

Average bitrate switches per device

<table>
<thead>
<tr>
<th>Device</th>
<th>No control</th>
<th>FlexStream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone1</td>
<td>60</td>
<td>10</td>
</tr>
<tr>
<td>Phone2</td>
<td>80</td>
<td>15</td>
</tr>
<tr>
<td>Tablet</td>
<td>70</td>
<td>12</td>
</tr>
</tbody>
</table>

Balanced QoE for phones and tablet

<table>
<thead>
<tr>
<th>Device</th>
<th>No control</th>
<th>FlexStream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phones</td>
<td>1500</td>
<td>2000</td>
</tr>
<tr>
<td>Tablet</td>
<td>1000</td>
<td>1500</td>
</tr>
</tbody>
</table>
Basic Experiments

FlexStream ability to consider user priority and screen size

Impact of background traffic on stability with no control

Throughput
Requested bitrate
Phone-high priority
Tablet-regular priority
Phone-regular priority

No Control
FlexStream
Basic Experiments – Cellular

Instability and unfairness with no control

Improved stability and fairness with FlexStream

Total throughput measured by all video players
Extended Experiments

![Extended Experiments Diagram]

- **Stability**: Comparison of No control and FlexStream for different AP capacities (7, 10, 13, 16, 19 Mbps).
- **Stall Duration**: Graph showing the comparison between No control and FlexStream for varying AP capacities.
- **Startup Delay**: Graph illustrating the performance difference between No control and FlexStream under different AP capacities.
- **JFI (Fairness)**: Comparison of JFI values for No control and FlexStream across different AP capacities (7, 10, 13, 16, 19 Mbps).
FlexStream Overheads

• GPAC player streams 1.4 Mbps video while DA is running in the background:
  
  • CPU utilization Overhead?
    
    • The CPU usage is around 1%
  
  • Bandwidth Overhead?
    
    • The total number of bytes sent and received while streaming the whole video is measured with and without enabling FlexStream.
    
    • FlexStream feedback and control messages found to incur less than 0.00004% of the total bandwidth needed to stream the whole video.
Conclusion and Future work

• We introduced:
  • **SDN-based framework** that extends SDN functionality to mobile end devices.
  • **An optimization method** to improve video QoE considering various context information, and validate it using real experiments.
  • **The first working implementation of the SDN extension** to commodity mobile devices that runs over WiFi and cellular without support from the network infrastructure.

• **Future work:** Integrating FlexStream into the existing network policy functions and obtaining link capacity and other state from the network directly.
Thank You!

QUESTIONS

tnadeem@vcu.edu
https://music.lab.vcu.edu/