INTRODUCTION TO VIDEO STREAMING

Vaneet Aggarwal

Parts of work joint with:
A. Elgabli, A. Alabassi, A. Ghosh, M. Felemban (Purdue University)
S. Sen, S. Hao (AT&T Labs-Research)
F. Qian (IU)
MOBILE VIDEO STREAMING IS GETTING SO POPULAR

- By 2021: 82% of Internet Traffic (3 ZB = $3 \times 10^9$ TB) will be video [CISCO]

- Q4 2016: mobile video surpassed desktop videos in terms of online viewing time [IAB]
MOBILE VIDEO STREAMING IS GETTING SO POPULAR

- By 2021: 82% of Internet Traffic (3 ZB = 3 \times 10^9 TB) will be video [CISCO]
- Q4 2016: mobile video surpassed desktop videos in terms of online viewing time [IAB]
RESEARCH GOAL

Improve Quality of Experience (QoE) and Resource Efficiency for Video Streaming

Less Stall  
Higher Quality  
Less Energy/Data Usage
• Video Streaming Algorithms
  – Single Path
  – Multiple Paths
  – Giant Client

• 360-degree Video Streaming

• Video Streaming over Cloud
• Video Streaming Algorithms
  – Single Path
  – Multiple Paths
  – Giant Client

• 360-degree Video Streaming

• Video Streaming over Cloud
• Each chunk is encoded into multiple bit-rates

• The user can fetch each chunk at one of these rates.
• **H.264/MPEG-4 AVC (Advanced Video Coding), Standardized 2003.** One out of L chunks must be selected.

• **SVC (Scalable Video Coding), Standardized in 2007 as a extension to H.264.** Has one base layer and multiple enhancement layers.
In SVC, rate adaptation can increase, and decrease the quality on the fly, but SVC encoding requires more overhead.
• LTE data collection over 30 different weekdays
• What quality should each of the video chunk be fetched at?

• Objective: Maximize the Quality of Experience of the end users.

• Some key metrics:
  – Stall Duration (Rebuffering)
  – Average quality
  – Quality Variations
Decide the quality of next chunk based on the buffer occupancy

Higher video rate with higher buffer occupancy

Gap: Buffer occupancy accounts for buffer and thus implicitly bandwidth, but not explicitly. Slow in learning bandwidth change.
• An optimization approach for deciding the quality levels

• Weighted combination of different QoE metrics is chosen
  – Stall Duration
  – Average Quality
  – Quality variation between chunks

• The optimization problem has discrete constraints (quality levels are discrete)
- Weighted combination of different QoE metrics is chosen
  - Stall Duration
  - Average Quality
  - Quality variation between chunks

- The optimization problem has discrete constraints (quality levels are discrete)

- A lookup table for nearby points is constructed, and used in real-time.
As a first step, we assume SVC encoding, and will generalize it later.

We also assume a skip-based formulation, where chunks if not received by deadline are skipped. The results will be extended to the stall-based version too.

Notations
- C chunks
- N+1 layers – 0, 1, ..., N
- Length of chunk, L
- Startup delay, s
- Size of Layer n, Y_n
- z_n(i,j) – Amount of layer n of chunk i fetched in time j
- Quality level at which chunk at layer n can be fetched Z_n ∈ {0, Y_n}
- Bandwidth limitation at every time j, B(j)
- Buffer capacity in time, B_m
OUR PROPOSED FORMULATION

- **Constraints:**

  **Layer Size Constraints**
  \[
  \sum_{j=1}^{(i-1)L+s} z_n(i,j) = Z_{n,i}, \quad \forall i, n
  \]

  **Decoder Constraints**
  \[
  Z_{n,i} \leq \frac{Y_n}{Y_{n-1}} Z_{n-1,i}, \quad \forall i, n > 0
  \]

  **Bandwidth Constraints**
  \[
  \sum_{n=0}^{N} \sum_{i=1}^{C} z_n(i,j) \leq B(j) \quad \forall j = 1, \ldots, (C-1)L + s,
  \]

  **Buffer Constraints**
  \[
  \sum_{i,(i-1)L+s>t} I \left( \sum_{j=1}^{t} \left( \sum_{n=0}^{N} z_n(i,j) \right) > 0 \right) L \leq B_m \quad \forall t
  \]

  **Positivity, deadline, and feasibility constraints**
  \[
  z_n(i,j) \geq 0 \quad \forall i = 1, \ldots, C
  \]
  \[
  z_n(i,j) = 0 \quad \forall i, (i-1)L+s > j
  \]
  \[
  Z_{n,i} \in \mathcal{Z}_n \triangleq \{0, Y_n\} \quad \forall i, n
  \]
OUR PROPOSED FORMULATION

- What could be a good metric (objective)?

- The increase in QoE with chunk quality has diminishing returns

- Minimizing Skips = Maximizing number of chunks for which base layer is fetched.
OUR PROPOSED FORMULATION

- What could be a good metric (objective)?

- The increase in QoE with chunk quality has diminishing returns

- Minimizing Skips = Maximizing number of chunks for which base layer is fetched.

- So, one approach seems like \((\gamma << 1)\)

  \((\text{Number of chunks in BL}) + \gamma \cdot \text{(Number of chunks in EL1)} + \gamma^2 \cdot \text{(Number of chunks in EL2)} + \ldots\)

- This prefers going higher layers when there is no possibility where more lower layers can be obtained, thus matching the diminishing returns objective.
WHICH STRATEGY IS BETTER?

(Number of chunks in BL) + $\gamma$ (Number of chunks in EL1) + $\gamma^2$ (Number of chunks in EL2) + ...
WHICH STRATEGY IS BETTER?

(Number of chunks in BL) + $\gamma$ (Number of chunks in EL1) + $\gamma^2$ (Number of chunks in EL2) + ...
OTHER CONSIDERATION?

(Number of chunks in BL) + γ (Number of chunks in EL1) + γ² (Number of chunks in EL2) + ... 

- The objective does not account for quality variations.
OTHER CONSIDERATION?

(Number of chunks in BL) + \( \gamma \) (Number of chunks in EL1) + \( \gamma^2 \) (Number of chunks in EL2) + ...

- The objective does not account for quality variations.

- Approach: Favor later chunks.

Maximize: \( \left( \gamma^0 \sum_{i=1}^{C} \beta^i Z_{0,i} + \gamma^1 \sum_{i=1}^{C} \beta^i Z_{1,i} + \ldots + \gamma^n \sum_{i=1}^{C} \beta^i Z_{n,i} \right) \)
For every layer $a$, all higher layers achieve less objective than fetching a chunk at layer $a$

This allows for layer prioritization.

Can independent decision at layers be chosen with such constraints?
For every layer \( a \), all higher layers achieve less objective than fetching a chunk at layer \( a \).

This allows for layer prioritization.

Can independent decision at layers be chosen with such constraints? No

- This is because when the chunk is fetched impacts the available bandwidth for future.
- The chunks at all times have to satisfy buffer capacity constraints.
The problem has discrete constraints, since each layer of each chunk is either fetched fully or not at all.

Also, there is a non-convex buffer constraint.

### Layer Size Constraints
\[
\sum_{j=1}^{(i-1)L+s} z_{n}(i, j) = Z_{n,i}, \quad \forall i, n
\]

### Decoder Constraints
\[
Z_{n,i} \leq \frac{Y_{n}}{Y_{n-1}} Z_{n-1,i}, \quad \forall i, n > 0
\]

### Bandwidth Constraints
\[
\sum_{n=0}^{N} \sum_{i=1}^{C} z_{n}(i, j) \leq B(j) \quad \forall j = 1, \ldots, (C-1)L + s,
\]

### Buffer Constraints
\[
\sum_{i,(i-1)L+s>t} \mathbf{1}\left(\sum_{j=1}^{t} \sum_{n=0}^{N} z_{n}(i, j) > 0\right) L \leq B_m \quad \forall t
\]

### Positivity, deadline, and feasibility constraints
\[
z_{n}(i, j) \geq 0 \quad \forall i = 1, \ldots, C
\]
\[
z_{n}(i, j) = 0 \quad \forall i, (i-1)L+s > j
\]
\[
Z_{n,i} \in Z_n \triangleq \{0, Y_n\} \quad \forall i, n
\]
Many problems are.
- Knapsack Problem
- Cutting Stock Problem
- Bin Packing Problem
- Travelling Salesman Problem
- -- and many more.

- All these are discrete optimizations, and NP hard.
• In contrast, some problems are not
  – Shortest Path Trees
  – Flows and Circulations
  – Spanning Trees
  – Matching Problem
  – Matroid Problem
The problem can be seen as bin-extension to the Knapsack problem.

Every item $i$ has a weight $w_i$ and value $v_i$.

3 components of item 1.

Every component $n$ of item $i$ has weight $w_i^n$ and value $v_i^n$. 
BIN-EXTENDED KNAPSACK PROBLEM

maximize: \( \sum_{n=0}^{N-1} \sum_{i=1}^{C} \sum_{t=1}^{C} v^n_i x_{i,n,t} \)

s.t

\( \sum_{n=0}^{N-1} \sum_{i=1}^{C} \sum_{j=1}^{t} x_{i,n,j} w^n_i \leq \sum_{j=1}^{t} B_j, \forall t \)

\( \sum_{t=1}^{C} x_{i,n,t} \leq \sum_{t=1}^{C} x_{i,n-1,t}, \forall i, n \)

\( x_{i,n,t} > 0, \forall i, n, t \)

\( \sum_{j=i+1}^{C} I(\sum_{t=1}^{N-1} \sum_{n=0}^{N-1} x_{j,n,t} > 0) \leq B_{max}, \forall i \)

\( x_{i,n,t} = 0, \forall i, n, t > i \)

\( \sum_{t=1}^{C} x_{i,n,t} \in \{0, 1\}, \forall i, n \)
Knapsack problem is NP hard, so is this extension.

However, we consider a special case

1. $w_i^n = w^n \forall i$

2. $v_i^n = v^n \forall i$

3. $C \sum_{k=a+1}^{N-1} v^k < v^a$, for all $a = 0, \ldots, N - 2$

We will see that the problem is polynomial-time solvable in this special case (indeed, linear time solvable in N and C).
Example Setup:
- 10 chunks, 1 s each
- Startup delay = 3s
- Maximum Buffer size = 4s
- 1 BL, 1 EL
BANDWIDTH PROFILE AND BUFFER CAPACITY

BW

Buffer
STEP 1: BACKWARD ALGORITHM FOR BL DECISIONS
STEP 2: FORWARD ALGORITHM FOR BL DECISIONS
BEFORE AND AFTER STEP 2
STEP 3: BACKWARD ALGORITHM FOR EL1 DECISIONS
OPTIMALITY

• Step 1: Without loss of generality, can consider in-order fetching.

• Step 2: Fetching algorithm maximizes number of fetched chunks at BL.
  – There are skips only due to two reasons – bandwidth insufficient or buffer violations. In both cases, it can be shown that the skips cannot be decreased as compared to the proposed algorithm

• Step 3: Forward algorithm maximizes the available bandwidth for higher layers among all decisions with same decisions at lower layers

• Recursively using the results give the optimality of the proposed algorithm

• Complexity: $O(NC)$ – linear in number of layers and number of chunks.
ONLINE ALGORITHM

• The predicted bandwidth may only be available for short time ahead

• Bandwidth prediction is inaccurate

• We Compute the scheduling decision only for the next W chunks ahead.

• Repeat after downloading every chunk to adjust to prediction errors and compute quality decisions for more chunks ahead.

• Our formulation works with any prediction technique. For evaluation, we consider harmonic mean and crowd-sourcing based predictions
The new objective function is:

\[
\text{Maximize: } \sum_{n=1}^{N} \gamma^n \sum_{i=1}^{C} \beta^i Z_{n,i} - \lambda d(C)
\]

Subject to all skip version constraints plus the following constraint

No Base Layer Skips

\[
\sum_{j=1}^{(i-1)L+s+d(i)} z_0(i, j) = Y_0 \forall i
\]

Stall duration up to the ith chunk
The Buffer is full, $d(i) = d(i) + 1$, for $i = 8, 9, \text{ and } 10$
NO-SKIP BASED STREAMING FORMULATION

There must be a stall after the 3\textsuperscript{rd} chunk
Because of the buffer violation

The rest is equivalent to skip version, backward-forward scan starting from the BL.
No-skip is a special case of skip in which chunk deadlines are chosen such that there are no skips.
Simulation and TCP/IP-based emulation testbed in C++.

For Performance measure, we compare our skip version with 3 Baselines

- Baseline 1: Conservative Algo, Fetch BL of all chunks before moving to E1 and so on. (Horizontal)
- Baseline 2: Optimistic Algo, Fetch all layers of current chunk and if there is still BW, fetch all possible layers of next chunk. (Vertical)
- Baseline 3: In between, Fetch all layers of current chunk and if there is still BW, fetch BL of later chunks

We compare No-Skip version with

- Netflix BBA-0,
- Naive port of Microsoft Streamer to SVC (NMS).
- Slope based algorithm

**TABLE I: SVC encoding bitrates used in our evaluation**

<table>
<thead>
<tr>
<th>playback layer</th>
<th>BL</th>
<th>EL1</th>
<th>EL2</th>
<th>EL3</th>
</tr>
</thead>
<tbody>
<tr>
<td>nominal Cumulative rate (Mbps)</td>
<td>0.6</td>
<td>0.99</td>
<td>1.5</td>
<td>2.075</td>
</tr>
</tbody>
</table>

**Fig. 4:** Statistics of the bandwidth traces: (a) mean and standard deviation of each trace’s throughput, and (b) trace length, across the 50 traces.
EVALUATION RESULTS FOR SVC - SKIP

<table>
<thead>
<tr>
<th></th>
<th>Base 1</th>
<th>Base 2</th>
<th>Base 3</th>
<th>HM (10,25)</th>
<th>(20,50)</th>
<th>offline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer %</td>
<td>0.4</td>
<td>0.6</td>
<td>0.8</td>
<td>1.0</td>
<td>1.2</td>
<td>1.4</td>
</tr>
<tr>
<td>S</td>
<td>26.5</td>
<td>14</td>
<td>14</td>
<td>16.9</td>
<td>14.5</td>
<td>15.7</td>
</tr>
<tr>
<td>BL</td>
<td>0.05</td>
<td>1.19</td>
<td>1.18</td>
<td>1.3</td>
<td>1.31</td>
<td>1.31</td>
</tr>
<tr>
<td>EL1</td>
<td>0.1</td>
<td>0.15</td>
<td>0.15</td>
<td>0.14</td>
<td>0.13</td>
<td>0.13</td>
</tr>
<tr>
<td>EL2</td>
<td>0.05</td>
<td>0.1</td>
<td>0.1</td>
<td>0.12</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>EL3</td>
<td>0.05</td>
<td>0.1</td>
<td>0.1</td>
<td>0.12</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>%Skips</td>
<td>6.5</td>
<td>19.7</td>
<td>19.2</td>
<td>9</td>
<td>7</td>
<td>5.7</td>
</tr>
<tr>
<td>APBR (Mbps)</td>
<td>1.05</td>
<td>1.19</td>
<td>1.18</td>
<td>1.3</td>
<td>1.31</td>
<td>1.31</td>
</tr>
<tr>
<td>Layer Switching Rate, LSR, in Mbps/chunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDF(LSR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.1</td>
<td>0.15</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.1</td>
<td>0.15</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Average Playback Rate, APBR, in Mbps

Layer Switching Rate, LSR, in Mbps/chunk
## EVALUATION RESULTS FOR SVC - STALL

<table>
<thead>
<tr>
<th>%BL</th>
<th>APBR (Mbps)</th>
<th>Stall time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>1.2</td>
<td>15</td>
</tr>
<tr>
<td>23</td>
<td>1.25</td>
<td>23</td>
</tr>
<tr>
<td>26</td>
<td>1.2</td>
<td>12</td>
</tr>
<tr>
<td>27.5</td>
<td>1.43</td>
<td>53</td>
</tr>
<tr>
<td>19</td>
<td>1.35</td>
<td>19</td>
</tr>
<tr>
<td>16</td>
<td>1.33</td>
<td>15</td>
</tr>
<tr>
<td>14</td>
<td>1.32</td>
<td>10</td>
</tr>
<tr>
<td>10.5</td>
<td>1.37</td>
<td>7</td>
</tr>
</tbody>
</table>

(a) Layer composition by percentage:

- Layer BL: [28, 23, 26, 27.5, 19, 16, 14, 10.5]
- Layer SB1: [1.2, 1.25, 1.2, 1.43, 1.35, 1.33, 1.32, 1.37]
- Layer SB2: [1.2, 1.25, 1.2, 1.43, 1.35, 1.33, 1.32, 1.37]
- Layer NMS: [1.2, 1.25, 1.2, 1.43, 1.35, 1.33, 1.32, 1.37]
- Layer HM: [1.2, 1.25, 1.2, 1.43, 1.35, 1.33, 1.32, 1.37]
- Layer (20,50%): [1.2, 1.25, 1.2, 1.43, 1.35, 1.33, 1.32, 1.37]
- Layer (60,100): [1.2, 1.25, 1.2, 1.43, 1.35, 1.33, 1.32, 1.37]
- Layer offline: [1.2, 1.25, 1.2, 1.43, 1.35, 1.33, 1.32, 1.37]

(b) CDF of Average Playback Rate (APBR) in Mbps:

- SB1: Continuous line
- (20,50%): Dashed line
- offline: Dotted line

(c) CDF of re-buffering time in seconds:

- SB1: Continuous line
- (20,50%): Dashed line
- offline: Dotted line

(d) CDF of Layer Switching Rate (LSR):

- BL: Continuous line
- EL1: Dashed line
- EL2: Dotted line
- EL3: Dashed-dotted line

---

**Layer Switching Rate (LSR):**

- **%BL:** 28, 23, 26, 27.5, 19, 16, 14, 10.5
- **APBR (Mbps):** 1.2, 1.25, 1.2, 1.43, 1.35, 1.33, 1.32, 1.37
- **Stall time (minutes):** 15, 23, 12, 53, 19, 15, 10, 7
Real implementation in dash.js open source video platform.
Comparison with FastMPC, Festive, BBA, RB, and dash.js reference
We use real video "Envivo", chunk duration is 4s.
EVALUATION RESULTS FOR AVC

Figure 3: Comparison with other approaches: Video Quality distribution
SUMMARY

• Novel formulation for video streaming

• Novel algorithms for video streaming

• Low complexity, and improved performance as compared to the baselines

• Some discrete optimization problems can be solved with low complexity.
• Video Streaming Algorithms
  – Single Path
  – Multiple Paths
  – Giant Client

• 360-degree Video Streaming

• Video Streaming over Cloud
MULTIPATH STREAMING

- Use multiple interfaces (e.g. WiFi and LTE) to download video chunks
MULTIPATH STREAMING

- MPTCP may or may not be allowed for using two links

- A vast amount of commercial content providers do not support MPTCP because it requires OS kernel update at both the client and server side.

- MPTCP uses special TCP extensions that are often blocked by middle-boxes of the commercial content providers (e.g. MPTCP over Port 80/443 is blocked by most U.S. cellular carriers).

- Implementing MPTCP with link preference further requires message exchange between the rate adaptation logic at the application layer and MPTCP in order to disable/enable parallel TCP connections.
PROBLEM FORMULATION WITH AVC USING MPTCP

Note that a chunk can be split into the two links since splitting a chunk among links is possible with MPTCP.
Example 1: MPTCP-AVC
In this example we consider a video consists of 10 chunks, 1s each
Startup delay = 1 s
The video is encoded into two quality levels
- 0-th Quality level, Bit rate is 1Mb/s, so chunk size is 2Mb
- 1-st Quality level, Bitrate 3Mb/s, so chunk size is 3Mb,
Therefore, the size difference between the two quality levels is 1Mb
FORWARD SCAN FOR 0-QUALITY LEVEL

<table>
<thead>
<tr>
<th>Total</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>11</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>WiFi</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4.5</td>
<td>5.5</td>
<td>6</td>
<td>7</td>
<td>7.5</td>
</tr>
<tr>
<td>LTE</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>1.5</td>
<td>2.5</td>
<td>3</td>
<td>3.5</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

WiFi BW

LTE BW

Time
Total stall duration is 1 second
Deadline(i)=(i-1)L+s+d(i), s=1, and d(i)=1 for all i
i.e deadline(1)=2

<table>
<thead>
<tr>
<th>Time</th>
<th>Total</th>
<th>WiFi</th>
<th>LTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>3.5</td>
<td>3</td>
<td>2.5</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>5</td>
<td>4.5</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>6</td>
<td>5.5</td>
</tr>
<tr>
<td>11</td>
<td>7.5</td>
<td>7</td>
<td>6.5</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>7.5</td>
<td>7</td>
</tr>
<tr>
<td>13</td>
<td>9</td>
<td>8</td>
<td>7.5</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>9</td>
<td>8</td>
</tr>
</tbody>
</table>

LTE BW
WiFi BW

Time
### Backward Scan for 0-Quality Level

<table>
<thead>
<tr>
<th>Time</th>
<th>WiFi BW</th>
<th>LTE BW</th>
<th>Total BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.5</td>
<td>3.5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2.5</td>
<td>4.5</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3.5</td>
<td>6.5</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>11</td>
<td>22</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>12</td>
<td>24</td>
</tr>
</tbody>
</table>

**Backward Scan Diagram:**

- WiFi BW: 2, 1, 4, 5, 6, 7
- LTE BW: 2, 3, 4, 5, 6, 7
- Total BW: 2, 1, 4, 5, 6, 7

**Time Axis:**

- Time Range: 1 to 14
DECISION FOR 1-QUALITY LEVEL

<table>
<thead>
<tr>
<th>Time</th>
<th>WiFi</th>
<th>LTE</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.5</td>
<td>2.5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1.5</td>
<td>3.5</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3.5</td>
<td>7.5</td>
</tr>
<tr>
<td>6</td>
<td>4.5</td>
<td>4</td>
<td>8.5</td>
</tr>
<tr>
<td>7</td>
<td>5.5</td>
<td>5</td>
<td>10.5</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>6</td>
<td>12.0</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>7</td>
<td>14.0</td>
</tr>
<tr>
<td>10</td>
<td>7.5</td>
<td>7</td>
<td>14.5</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>6</td>
<td>14.0</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>5</td>
<td>14.0</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>4</td>
<td>14.0</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>3</td>
<td>13.0</td>
</tr>
</tbody>
</table>

[Graph showing the decision process for quality levels.]
DECISION FOR 1-QUALITY LEVEL

<table>
<thead>
<tr>
<th>Time</th>
<th>WiFi</th>
<th>LTE</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.5</td>
<td>2.5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1.5</td>
<td>3.5</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2.5</td>
<td>4.5</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3.5</td>
<td>6.5</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>

LTE BW

WiFi BW

PURDUE UNIVERSITY
CONSIDERING PREFERENCES: EXCHANGE ALGORITHM

[Diagram showing WiFi and LTE bandwidth over time with exchange algorithm]
MULTIPATH STREAMING

- MPTCP may or may not be allowed for using two links.

- A vast amount of commercial content providers do not support MPTCP because it requires OS kernel update at both the client and server side.

- MPTCP uses special TCP extensions that are often blocked by middle-boxes of the commercial content providers (e.g., MPTCP over Port 80/443 is blocked by most U.S. cellular carriers).

- Implementing MPTCP with link preference further requires message exchange between the rate adaptation logic at the application layer and MPTCP in order to disable/enable parallel TCP connections.
Without MPTCP

- Without MPTCP, we use SVC since each layer can be obtained from different link.

- Forward scan: Finds stalls using greedy approach

- Other steps, similar to before, where we assume that LTE can only be used upto a layer and prefer WiFi to LTE.
**EVALUATIONS**

Fig. 1. System Architecture for Emulation of MP-SVC.

| TABLE I |
|------------------|---|---|---|---|---|
|                  | BL | EL1 | EL2 | EL3 |
| nominal Cumulative rate (Mbps) | 0.6 | 0.99 | 1.5 | 2.075 |

Table 1: AVC encoding chunk sizes in MB of the Envívio video used in our evaluation

<table>
<thead>
<tr>
<th>Quality level</th>
<th>level-0</th>
<th>level-1</th>
<th>level-2</th>
<th>level-3</th>
<th>level-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.0433</td>
<td>0.0786</td>
<td>0.1265</td>
<td>0.2213</td>
<td>0.3205</td>
</tr>
<tr>
<td>Mean</td>
<td>0.1693</td>
<td>0.2916</td>
<td>0.4795</td>
<td>0.949</td>
<td>1.403</td>
</tr>
<tr>
<td>Max</td>
<td>0.2342</td>
<td>0.3855</td>
<td>0.6217</td>
<td>1.286</td>
<td>1.918</td>
</tr>
</tbody>
</table>

(a) CDF (Average BW) vs. Average BW in Mbps

(b) CDF (SD) vs. Standard Deviation (SD) in Mbps
EVALUATION RESULTS FOR SVC

Fig. 3. Skip based Streaming without link preference: (a) layer distribution, and (b) CDF of layer switching rate.

Fig. 4. Skip based streaming with Link Preference: (a) layer distribution, and (b) CDF of link 1 usage.
EVALUATION RESULTS FOR AVC

No Preference Results

<table>
<thead>
<tr>
<th></th>
<th>Avg(Mbps)</th>
<th>Stall(seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP-DASH-BBA</td>
<td>2.2</td>
<td>8</td>
</tr>
<tr>
<td>MSPlayer</td>
<td>1.65</td>
<td>213</td>
</tr>
<tr>
<td>MP-DASH-Festive</td>
<td>2.48</td>
<td>110</td>
</tr>
<tr>
<td>SmartStreamer</td>
<td>2.75</td>
<td>8</td>
</tr>
<tr>
<td>Off-SmartStreamer</td>
<td>2.87</td>
<td>0</td>
</tr>
</tbody>
</table>

![Diagram a](chart1.png)

![Diagram b](chart2.png)

![Diagram c](chart3.png)
EVALUATION RESULTS FOR AVC

Preference Results

<table>
<thead>
<tr>
<th>Avg(Mbps)</th>
<th>1.6</th>
<th>1.4</th>
<th>1.67</th>
<th>2.18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stall(seconds)</td>
<td>8</td>
<td>110</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

(a)

(b)

(c)

**EVALUATION RESULTS FOR AVC**
• Novel formulation for multi-path video streaming

• Considers link preference in the design of multi-path streaming algorithms

• Novel algorithms with low complexity and improved performance as compared to the baselines
• Video Streaming Algorithms
  – Single Path
  – Multiple Paths
  – Giant Client

• 360-degree Video Streaming

• Video Streaming over Cloud
COOPERATIVE VIDEO STREAMING

• Group of users are gathered in a low speed internet place.

• They are interested in watching a TV show on a common screen

• Willing to cooperate, but:
  – Users may have different data plan limits
  – Devices are running into different energy levels

• Maximum contribution of every user should not be violated
• Multipath TCP cannot be used due to different users from potentially different carriers involved

• The approach for multipath can be used here for fetching chunks from servers.
TABLE I
SVC ENCODING BITRATES USED IN OUR EVALUATION

<table>
<thead>
<tr>
<th>playback layer</th>
<th>BL</th>
<th>EL1</th>
<th>EL2</th>
<th>EL3</th>
</tr>
</thead>
<tbody>
<tr>
<td>nominal Cumulative rate (Mbps)</td>
<td>1.5</td>
<td>2.75</td>
<td>4.8</td>
<td>7.8</td>
</tr>
</tbody>
</table>

TABLE II
MAX CONTRIBUTIONS USED IN OUR EVALUATION

<table>
<thead>
<tr>
<th>User No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Contribution (Mb)</td>
<td>700</td>
<td>500</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Fig. 2. System Setup.

Fig. 3. Statistics of the two bandwidth traces: (a) mean, and (b) standard deviation of each trace’s available bandwidth.
RESULTS

(a) Layer distribution for different layers (S, BL, EL1, EL2, EL3) across BB, PBB, GroupCast, and Off-GroupCast.

(b) CDF of skip duration in minutes for BB, PBB, GroupCast, and Off-GroupCast.

(c) CDF of Average Playback Rate (ABPR) in Mbps for BB, PBB, GroupCast, and Off-GroupCast.

(d) Average fetching % for user1, user2, user3, and user4 across BB, PBB, GroupCast, and Off-GroupCast.

(e) CDF of Layer Switching Rate (LSR) in Mbps/chunk for BB, PBB, GroupCast, and Off-GroupCast.

TABLE I
SVC encoding bitrates used in our evaluation
<table>
<thead>
<tr>
<th>playback layer</th>
<th>BL</th>
<th>EL1</th>
<th>EL2</th>
<th>EL3</th>
</tr>
</thead>
<tbody>
<tr>
<td>nominal rate (Mbps)</td>
<td>1.5</td>
<td>2.75</td>
<td>4.8</td>
<td>7.8</td>
</tr>
</tbody>
</table>

TABLE II
Max contributions used in our evaluation
<table>
<thead>
<tr>
<th>User No.</th>
<th>Max Contribution (Mb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>700</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
</tr>
</tbody>
</table>
• Natural extension of the work to cooperative streaming

• Cooperation helps fetching content from multiple users
• Video Streaming Algorithms
  – Single Path
  – Multiple Paths
  – Giant Client

• 360-degree Video Streaming

• Video Streaming over Cloud
360-DEGREE VIDEO STREAMING

- A key application of Virtual Reality (VR)
- Provide users with panoramic views
HOW ARE 360 VIDEOS CAPTURED & PLAYED?

Omni-directional Camera

Projected view in the viewport

Viewing direction (lat/lon) & Field-of-view (FoV) size

Raw Panoramic Frame

Projected view in the viewport

Purdue University
Equirectangular (YouTube)

CubeMap (Facebook)
360 videos inherit almost everything from regular videos

- Same encoding scheme
- Same rate adaptation algorithm
- Same client/server infrastructure
- Viewport agnostic from networking perspective: client downloads the entire raw panoramic frame regardless of user’s viewport

360/VR videos  \(=\)  Regular video streaming techniques  \(+\)  Projection
Tremendous bandwidth waste

- User only consumes ~15% of the downloaded contents

Under the same perceived quality, 360 videos are 4~6 times larger than regular videos.

- Roller coaster, 1’57”, 4K, 250MB+
360-DEGREE VIDEO STREAMING

High-level idea: FoV-aware streaming!
- Predict user’s head movement and only fetch portions that the user is about to see

Target commodity mobile devices & low-end VR headsets

Significantly reduce bandwidth utilization, or improve perceived quality

Bare Smartphone
Google Cardboard, $10
Samsung Gear VR, $80
• A chunk is divided into tiles

• Efficient prediction of FoV can be obtained for some time-ahead

• Statistics of viewing each tile can be known

• How to account for FoV to obtain new streaming algorithm?
360-DEGREE VIDEO STREAMING

• What are the metrics?
  – Minimum rate in FoV
  – Average Rate in FoV
  – Stalls
  – Quality variations in FoV

• What if prediction is wrong – how about stalls?

• We propose minimum quality for all tiles, so that we do not have to stall due to incorrect prediction
What are the metrics?
- Minimum rate in FoV
- Average Rate in FoV
- Stalls
- Quality variations in FoV

One metric: Choose 99th percentile area, and fetch higher quality in that, BL in rest

Flexible: What quality for this bigger area?

Akin to VBR – and efficient streaming strategies have been proposed.
• Improves Playback rate and reduce stalls and quality variations as compared to approaches getting all tiles at same quality.
SUMMARY

- 360-degree adds a challenge of FoV prediction
- Exploiting the FoV prediction with tile-based streaming
- A new formulation is provided, and algorithms are proposed
• Video Streaming Algorithms
  – Single Path
  – Multiple Paths
  – Giant Client

• 360-degree Video Streaming

• Video Streaming over Cloud
• Video streaming applications represents 62% of the Internet traffic in US

• More than 50% of over-the-top video traffic is now delivered through CDNs

• What are the key challenges?
RESEARCH PROBLEMS

- Network Optimization: Determination of Caching, Bandwidths, and Server access for Optimized User QoE
- Computation Optimization: Multiple MapReduce Jobs for Video Processing and Data Analytics, optimized completions.
- Video Streaming Algorithms: Adaptive bit-rate algorithms for optimized user QoE
• Part of each video is cached at cache servers.

• The content from cache is downloaded from one of $e_j$ streams and that from the central cloud though $d_j$.

• Each video request chooses one of the distributed server, and correspondingly one of $d_j$ and one of $e_j$ streams.
• Part of each video is cached at cache servers.
**DECISION VARIABLES**

- Choice of Distributed Server
- Amount of each video in cache
- Choice of $e_j$ streams
- Choice of $d_j$ streams
- Bandwidth allocation among the different streams
OBJECTIVE

- Wish to Minimize Stalls

- One of the key metrics is to minimize the probability that the stalls are above a threshold.

- Metric is called Stall Duration Tail Probability
CHALLENGES

• Computation of objective is hard

• The key reason is that the choice of server, and the streams is a decision process, and the embedded computation of Markov Chain is challenging.

• Approach: Have the choice be independent of the queue state. [Proposed earlier for erasure-coded cloud storage by Aggarwal et al., Sigmetrics 2014, TON 2016.]

• Probabilistic Scheduling: Make the choice randomly with certain probabilities. However, the probabilities become design variables.

• Approach can give an upper bound for the objective
CHALLENGES

• Stall duration does not only depend on download of a chunk

• We need to compute download time of all chunks and stalls between chunks.

• Makes the metric harder to compute since download times are stochastic and the stall times are non-convex function of download times.

• Approach: Use efficient bounds.
For a file request for video $i$, assume that the servers, streams, and the cache contents at the server are known.

**Step 1:** Download Time for Chunks from the Cache Servers:

- Download time of first chunk only depends on the waiting time for previous videos, and the service time of 1st chunk.
- Download time of later chunks can be found by adding service times.

\[
D_{i,j,\beta_j,\nu_j}^{(g)} = W_{j,\nu_j}^{(e)} + \sum_{v=1}^{n} Y_{i,j,\nu_j}^{(e,v)}
\]
For a file request for video i, assume that the servers, streams, and the cache contents at the server are known.

**Step 2:** Download Time for Chunks from the Cloud Storage:

This is challenging due to two queues. The chunk first has to come on top pipe, be downloaded, and then be enqueued in the bottom pipe.
• **Step 2:** Download Time for Chunks from the Cloud Storage:

• This is challenging due to two queues. The chunk first has to come on top pipe, be downloaded, and then be enqueued in the bottom pipe.

\[
D^{(L_{j,i}+1)}_{i,j,\beta_i,\nu_i} = \max(W_{i,\beta_i}^{(c)} , E_{i,\beta_i}^{(L_{j,i}+1)} ) + Y^{(c,L_{j,i}+1)}_{i,\beta_i} \\
D^{(g)}_{i,j,\beta_j,\nu_j} = \max(D^{(g-1)}_{i,j,\beta_j,\nu_j} , E^{(g)}_{i,j,\beta_j} ) + Y^{(c,g)}_{j,\beta_j}
\]
• **Step 3:** Play Time for Chunks from the Cloud Storage:

Knowing download time of all chunks, the play times can be calculated.

\[
T^{(1)}_{i,j,\beta_j,\nu_j} = \max \left( d_s, D^{(1)}_{i,j,\beta_j,\nu_j} \right)
\]

\[
T^{(q)}_{i,j,\beta_j,\nu_j} = \max \left( T^{(q-1)}_{i,j,\beta_j,\nu_j} + \tau, D^{(q)}_{i,j,\beta_j,\nu_j} \right)
\]

• Thus, the statistics of Play times can be calculated.
• **Step 4: Stall Duration:**

- Stall duration can be calculated from the play time of the last chunk.
- Ideally, last chunk must be played at Startup Delay + (C-1) Chunk Duration
- The delay between the actual play time and the ideal time is the stall duration.

\[ \Gamma^{(i,j,\beta_j,\nu_j)} = T^{(L_i)}_{i,j,\beta_j,\nu_j} - d_s - (L_i - 1) \tau \]
• **Step 5:** Stall Duration Tail Probability:

Knowing the moment generating function of the stall duration, bounds on tail probability can be calculated, over the statistics of the probabilistic scheduling.

\[
\Pr \left( \Gamma(i,j,\beta_j,\nu_j) \geq \sigma \right) \overset{(a)}{=} \Pr \left( T_{i,j,\beta_j,\nu_j} \geq \sigma + d_s + (L_i - 1) \tau \right)
\]
• We wish to optimize the stall duration tail probability, weighted for different video requests, to determine the choice of bandwidth allocation, caching, and the probabilistic scheduling.

• The problem is non-convex, has integer constraints.

• We use an alternating minimization based algorithm over different variable groups, and solve each variable group using iNner cOnVex Approximation (NOVA) algorithm.

• This guarantees convergence to a local optima.
EVALUATIONS

• Comparisons:
  – Projected Equal Server-PSs Scheduling, Optimized Auxiliary variables, Caching, and Bandwidth Weights (PEA)
  – Projected Equal Bandwidth, Optimized Access Servers and PS scheduling Probabilities, Auxiliary variables and cache placement (PEB)
  – Projected Equal Caching, Optimized Scheduling Probabilities, Auxiliary variables and Bandwidth Allocation Weights (PEC)
  – Fixed-t: Fixed Auxiliary Variables.
  – Projected Proportional Service-Rate, Optimized Auxiliary variables, Bandwidth Weights, and Cache Placement (PSP)
EVALUATIONS

• Comparisons:
  - Projected Equal Server-PSs Scheduling, Optimized Auxiliary variables, Caching, and Bandwidth Weights (PEA)
  - Projected Equal Bandwidth, Optimized Access Servers and PS scheduling Probabilities, Auxiliary variables and cache placement (PEB)
  - Projected Equal Caching, Optimized Scheduling Probabilities, Auxiliary variables and Bandwidth Allocation Weights (PEC)
  - Fixed-t: Fixed Auxiliary Variables.
  - Projected Proportional Service-Rate, Optimized Auxiliary variables, Bandwidth Wights, and Cache Placement (PSP)

Better than Baselines
EVALUATIONS

• Comparisons:
  - Projected Equal Server-PSs Scheduling, Optimized Auxiliary variables, Caching, and Bandwidth Weights (PEA)
  - Projected Equal Bandwidth, Optimized Access Servers and PS scheduling Probabilities, Auxiliary variables and cache placement (PEB)
  - Projected Equal Caching, Optimized Scheduling Probabilities, Auxiliary variables and Bandwidth Allocation Weights (PEC)
  - Fixed-t: Fixed Auxiliary Variables.
  - Projected Proportional Service-Rate, Optimized Auxiliary variables, Bandwidth Wights, and Cache Placement (PSP)

Benefit increases with load
The proposed algorithm was implemented on real servers.
The proposed algorithm was implemented on real servers.
• Edge Router also has limited cache

• What is good caching policy at the edge router?
• Edge Router also has limited cache

• What is good caching policy at the edge router?

• LRU (Least Recently Used): Place last requested file in cache, remove the least recently used file.

• Edge cache can allow for multicast since the same content can be sent to another user even while being downloaded.
• Edge Router also has limited cache

• What is good caching policy at the edge router?

• LRU (Least Recently Used): Place last requested file in cache, remove the least recently used file.

• Edge cache can allow for multicast since the same content can be sent to another user even while being downloaded.

• LRU does not account for file priority, and is known to be bad for asymmetric file sizes. Thus, we consider TTL (time-to-live) where each file is removed after certain time (parameter) if not accessed.
We consider TTL (time-to-live) where each file is removed after certain time (parameter) if not accessed.

The cache has limited size, so parameters can be found to keep probability of cache over-size very small. Online adaptation can be done to manage such low probability violation.

With this edge cache strategy and multicast, stall duration statistics can be calculated. The rest of the procedure is similar as described earlier.
IMPLEMENTATION RESULTS
• CDN based video delivery system is considered

• Stall duration statistics are analyzed

• An optimized system is developed with knobs of scheduling, resource allocation, and caching
Thank You!