Enhancing Software-Defined RAN with Collaborative Caching and Scalable Video Coding

Ruozhou Yu, Shuang Qin, Mehdi Bennis, Xianfu Chen, Gang Feng, Zhu Han, Guoliang Xue
Agenda

Introduction
Problem Formulation
Solution Design
Performance Evaluation
Conclusions
Software-Defined RAN

Decoupled control & data planes.

Centrally managed resources and info.
- Opportunity for inter-BS collaboration.
- Global optimization for content delivery
Video Caching in RAN
Video Caching in RAN

Video 1 request

Video 2 request

Video 3 request
Collaborative Caching
Scalable Video Coding (SVC)

Videos have different bitrate versions: 360p, 720p, 1080p, etc.

SVC slices video into different layers:
- **Base layer** guarantees the minimum bitrate playback
- Each **enhancement layer** increases bitrate by one level

Layering Constraint: to get enhanced BR $l$, both the base layer and all enhancement layers below $l$ are needed.
Agenda

Introduction

Problem Formulation

Solution Design

Performance Evaluation

Conclusions
Problem Overview

Generally, given the network status and some prediction on user’s video requests, we want to decide two things:

1) which base station’s cache stores which layers of which video, and
2) how to schedule (route) the video streams of each request,

such that we maximize the number and quality of videos served, meanwhile minimizing the delay received by users.
Network Model

Base stations $B=\{B_0, \ldots, B_M\}$
- Cache size $c_i$
- Upstream/downstream backhaul bandwidth $b_i^u/b_i^d$
- $B_0$ denotes the Internet with unlimited cache and bandwidth
- Distance between two BSs $d_{i,t}$

Videos $\mathcal{V} = \{V_1, \ldots, V_N\}$

Layers $\mathcal{L}_j = \{1, \ldots, L_j\}$ for each video $V_j$
- Layer size $s_j^l$
- Layer bandwidth requirement $\beta_j^l$

$\psi_{i,j,l}$: number of users at BS $B_\nu$, requesting the first $l$ layers of video $V_j$
Variables

$x_{i,j,l}^{}$: indicator of caching video $V_j$’s layer $l$ at BS $B_i$

$z_{i,j,l}^{}$: number of video $V_j$’s layer $l$ requested by users from BS $B_i$ and served by the cache of BS $B_i$
Objectives

\( r_{j,l}^b \): unit reward for serving video \( V_j \)'s layer \( l \) to user;

\( c_j^d \): unit cost for incurring delay for video \( V_j \)

Objective:

\[
\text{maximize} \quad R = \sum_{\iota=1}^{M} \sum_{j=1}^{N} \left( \sum_{l=1}^{L_j} r_{j,l}^b \cdot \sum_{i=0}^{M} z_{i,t}^{j,l} - c_j^d \cdot d_{i}^{j} \right)
\]

- \( d_{i}^{j} \): aggregated delay received by video \( V_j \)'s user(s) at BS \( B_t \)
Constraints

- Capacity constraints:

\[
\sum_{j=1}^{N} \sum_{l=1}^{L_{j}} x_{i,j}^{l} \cdot s_{j}^{l} \leq c_{i}, \quad \forall B_{i} \in B
\]

\[
\sum_{\iota \neq i} \sum_{j=1}^{N} \sum_{l=1}^{L_{j}} z_{i,j}^{l} \cdot \beta_{j}^{l} \leq b_{i}^{u}, \quad \forall B_{i} \in B
\]

\[
\sum_{i \neq \iota} \sum_{j=1}^{N} \sum_{l=1}^{L_{j}} z_{i,j}^{l} \cdot \beta_{j}^{l} \leq b_{i}^{d}, \quad \forall B_{i} \in B
\]
Constraints cont.

- **Caching & scheduling constraints:**
  \[
  z^j_{i, l} \leq x^j_{i, l} \cdot \psi^j_{i, l}, \quad \forall B_i \in \mathcal{B}, B_l \in \mathcal{B}, V_j \in \mathcal{V}, l \in [1, L_j]
  \]
  \[
  \sum_{i=0}^{M} z^j_{i, l} \leq \psi^j_{i, l}, \quad \forall B_l \in \mathcal{B}, V_j \in \mathcal{V}, l \in [1, L_j]
  \]
  \[
  \sum_{i=0}^{M} z^j_{i, l} \leq \sum_{i=0}^{M} z^j_{i, l-1}, \quad \forall B_l \in \mathcal{B}, V_j \in \mathcal{V}, l \in [2, L_j]
  \]

- **Delay constraints:**
  \[
  \lambda^j_{i_p, l} = \max_{l \in \mathcal{L}_j} \left\{ \sum_{q=0}^{p} z^j_{i_{q}, l} \right\} - \sum_{q=0}^{p-1} \lambda^j_{i_{q}, l}, \quad \forall B_l \in \mathcal{B}, V_j \in \mathcal{V}, p \in \{0, \cdots, M\}
  \]
  \[
  d^j_l = \sum_{i=0}^{M} \lambda^j_{i, l} \cdot d_{i, l}, \quad \forall B_l \in \mathcal{B}, V_j \in \mathcal{V}
  \]
  Linearized as inequalities
Agenda

Introduction
Problem Formulation
Solution Design
Performance Evaluation
Conclusions
Algorithm Overview

Two stages:
- Stage 1: decide the caching of videos (layers) at each base station;
- Stage 2: decide which base station serves each layer of each user’s request, based on the Stage 1 results.

Rounding-based algorithm:
- Relax the ILP formulation to LP;
- Solve for a (fractional) solution;
- Use deterministic rounding technique to obtain an integral solution.
Stage 1: Video Caching

1. Form ILP formulation and get LP relaxation
2. Solve LP and obtain fractional solution
3. Compute each variable $x$'s contribution towards objective
4. Greedily round the caching variable $x$ in sorted order
5. Output caching decisions
Stage 2: Video Scheduling

1. Form LP with *fixed* caching variables $x$
2. Solve LP and obtain fractional solution
3. Round down each scheduling variable $z$ to obtain a basic solution
4. More can be scheduled?
   - NO: Schedule more requests via backhaul
   - YES: Output scheduling decisions
Agenda

Introduction
Problem Formulation
Solution Design
Performance Evaluation
Conclusions
Setup

Randomly generated RAN environment:
- 15 BSs
- 10000 users uniformly distributed
- 5000 videos with Zipf popularity distribution: $\gamma=0.95$
- 5 layers per video: 10% coding overhead introduced
- Randomly generated cache, video size and bandwidth capacity/demands

Five schemes for comparison:
- SC: SVC + Collaborative caching
- SS: SVC + Single BS caching
- NC: Non-SVC + Collaborative caching
- NS: Non-SVC + Single BS caching
- NN: Non-SVC + No caching
## Exp. With Default Parameters

<table>
<thead>
<tr>
<th>Combination</th>
<th>Act users</th>
<th>Avg layers</th>
<th>Avg delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC + Collaborative (SC)</td>
<td>6480.9</td>
<td>2.1249</td>
<td>88.31</td>
</tr>
<tr>
<td>SVC + Non-collaborative (SS)</td>
<td>6492.1</td>
<td>2.1306</td>
<td>102.14</td>
</tr>
<tr>
<td>Non-SVC + Collaborative (NC)</td>
<td>5635.2</td>
<td>2.0077</td>
<td>103.80</td>
</tr>
<tr>
<td>Non-SVC + Non-collabor. (NS)</td>
<td>5642.4</td>
<td>2.0092</td>
<td>115.90</td>
</tr>
<tr>
<td>No caching (NN)</td>
<td>3201.1</td>
<td>1.2808</td>
<td>143.41</td>
</tr>
</tbody>
</table>
Active Users

![Graph showing the number of active users vs. total users for different categories: SC, SS, NC, NS, NN. The graph includes a legend and labels for the x-axis (# Total users) and y-axis (# Active users).]
Average Layers (Bitrates)
Average Delay

Average delay (ms)

# Total users

2k 4k 6k 8k 10k 12k 14k 16k 18k 20k

SC
SS
NC
NS
NN
Agenda

Introduction
Problem Formulation
Solution Design
Performance Evaluation
Conclusions
Conclusions

Enhancing video delivery in software-defined RAN with collaborative caching and SVC.
- Collaborative caching to reduce user delay;
- SVC to increase cache reuse and serve more users.

Maximizing rewards and minimizing delay: a joint problem.
- NP-hard

2-stage rounding-based algorithm.
- Decide caching first.
- Schedule videos based on caching.

Outperforms using either collaborative caching or SVC alone.