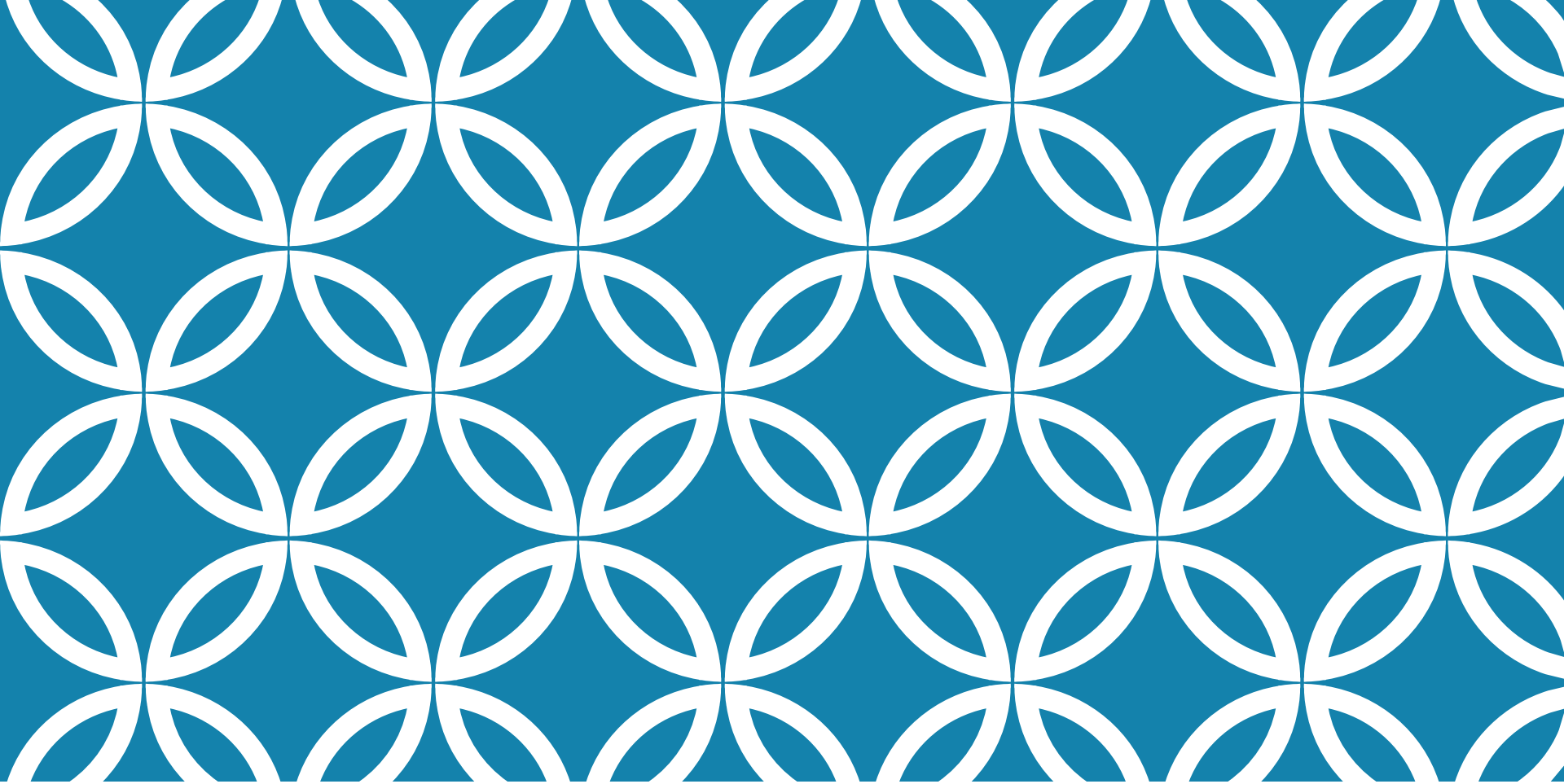


Enhancing Software-Defined RAN with Collaborative Caching and Scalable Video Coding

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

Agenda

Introduction

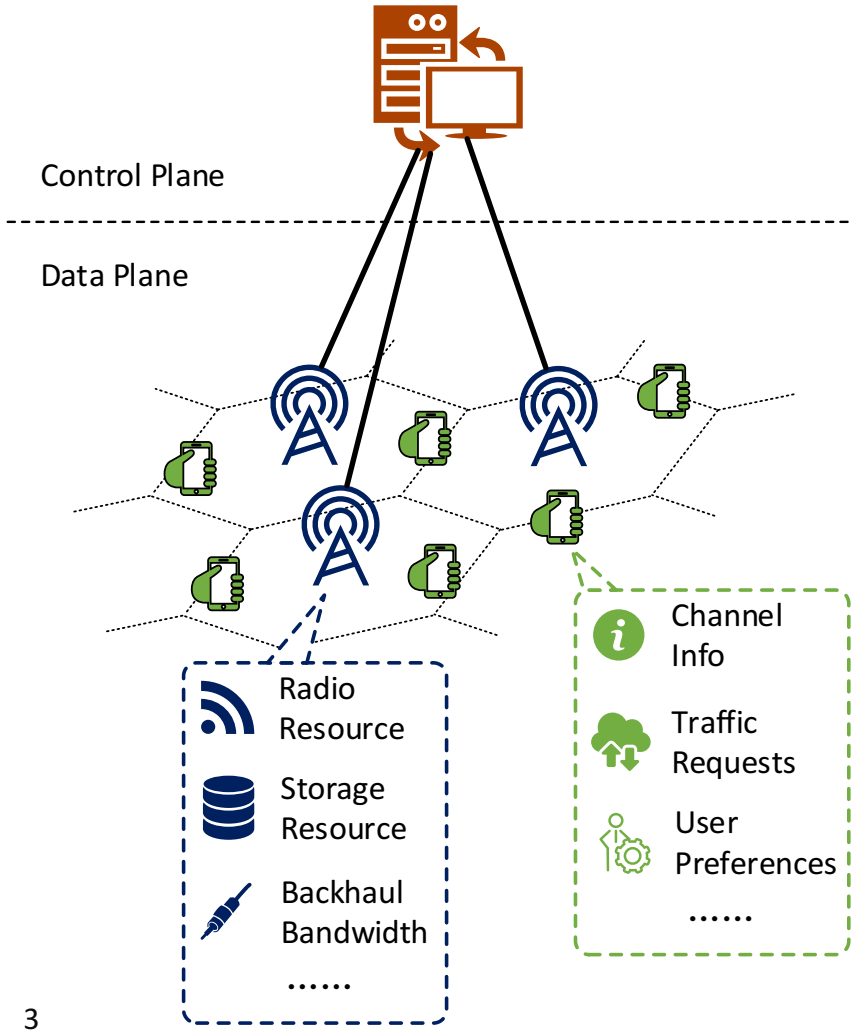
Problem Formulation

Solution Design

Performance Evaluation

Conclusions

Software-Defined RAN

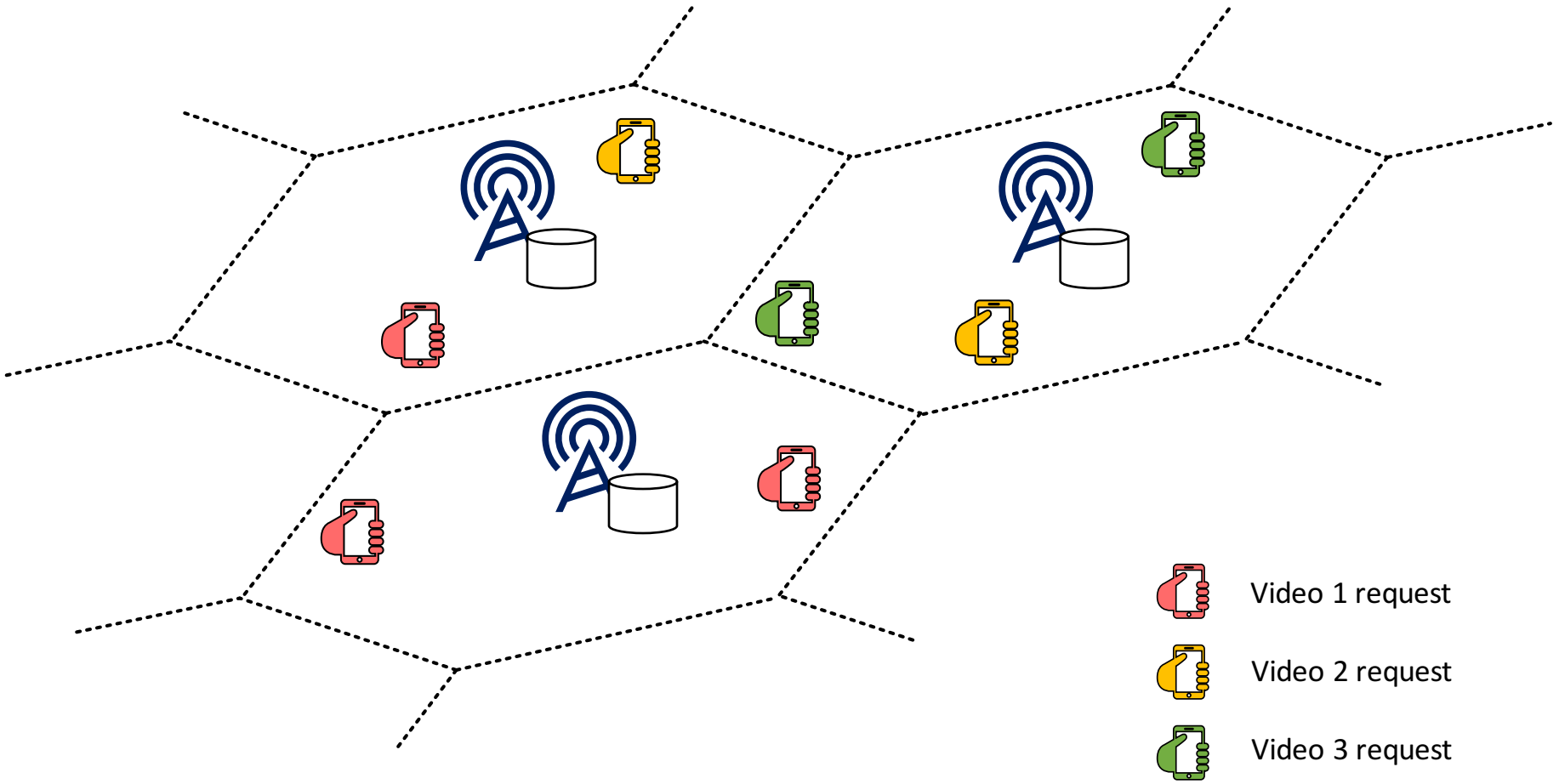


Decoupled control & data planes.

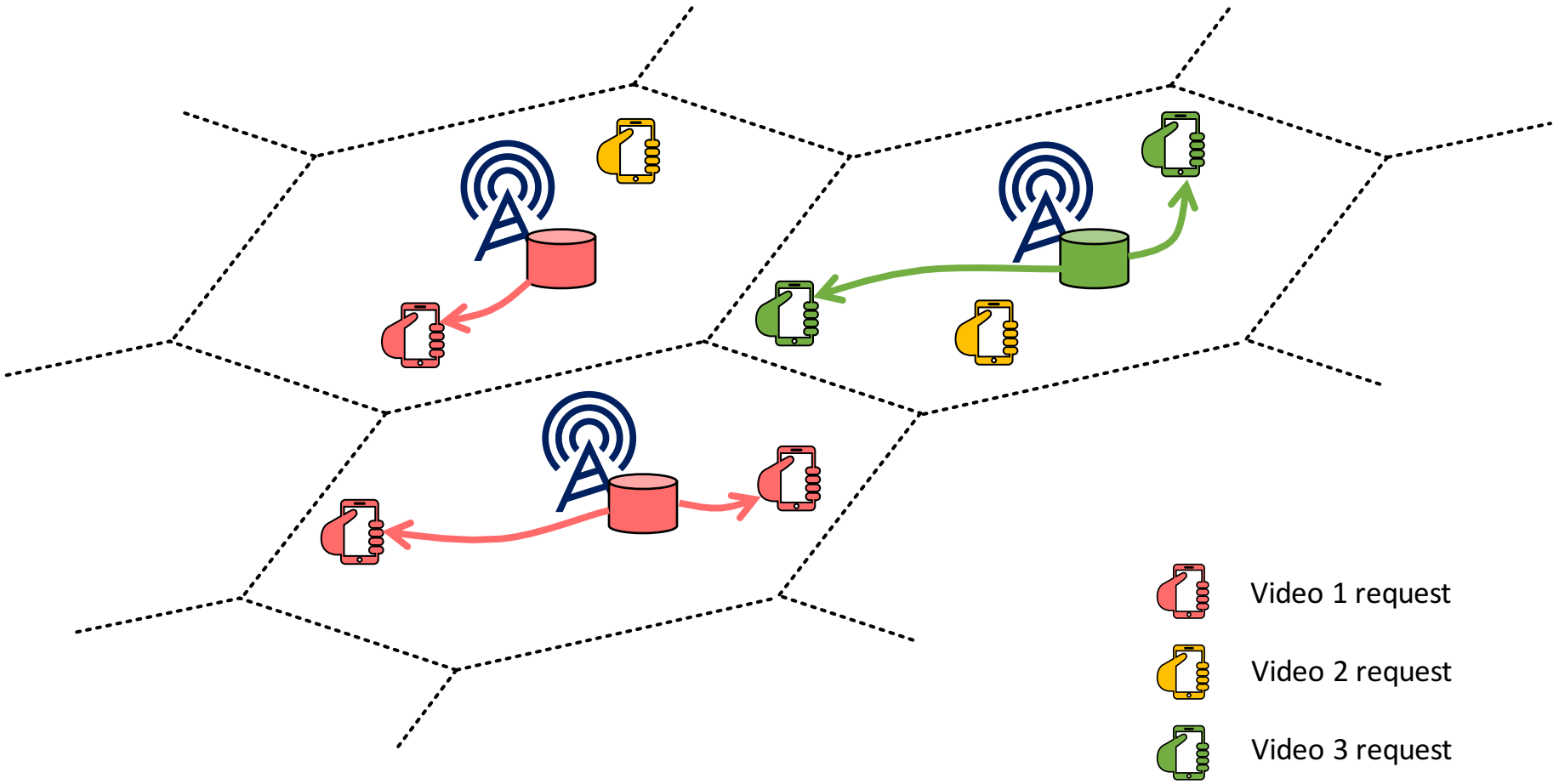
Centrally managed resources and info.

- Opportunity for inter-BS collaboration.
- Global optimization for content delivery

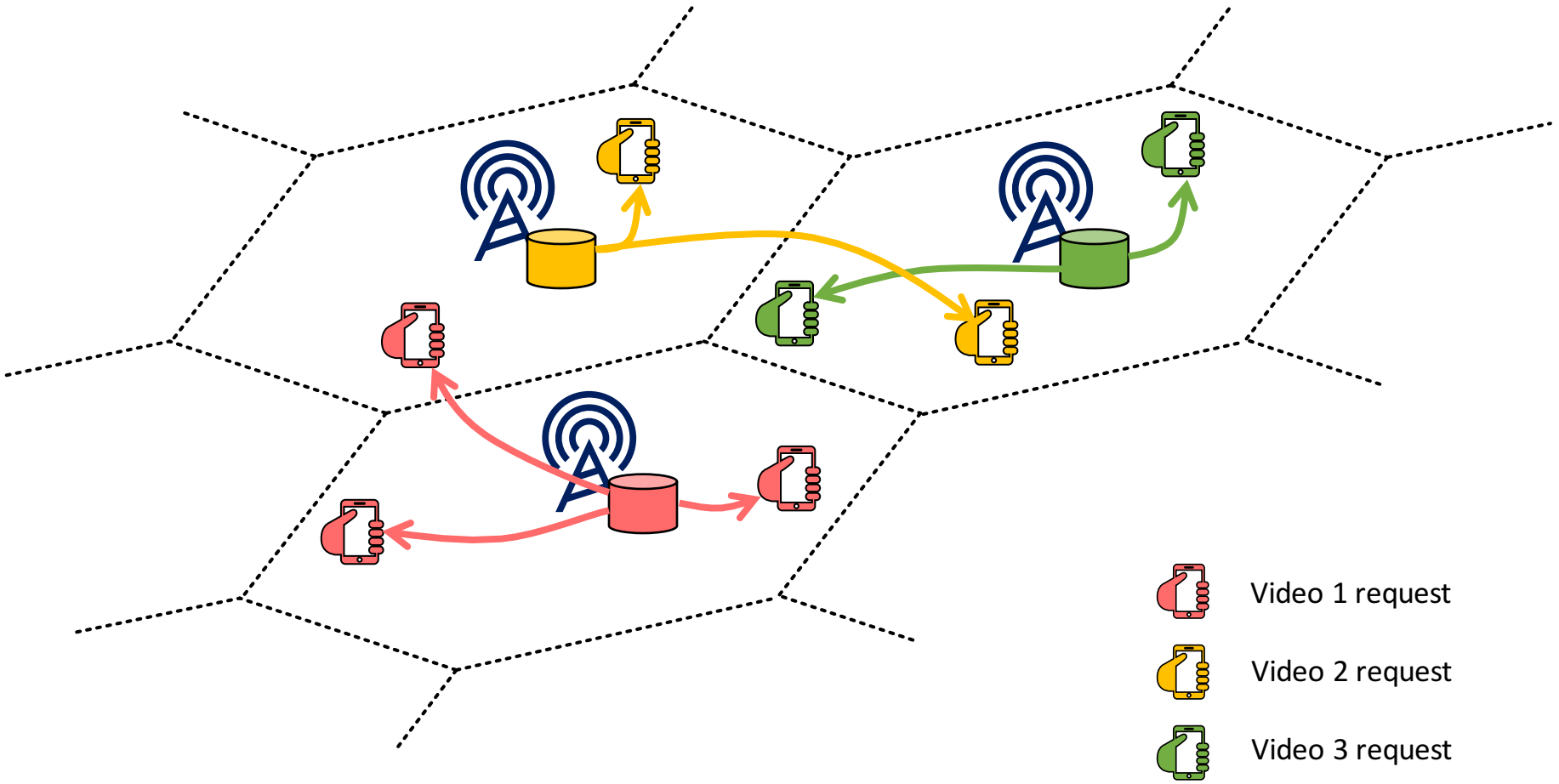
Video Caching in RAN



Video Caching in RAN



Collaborative Caching

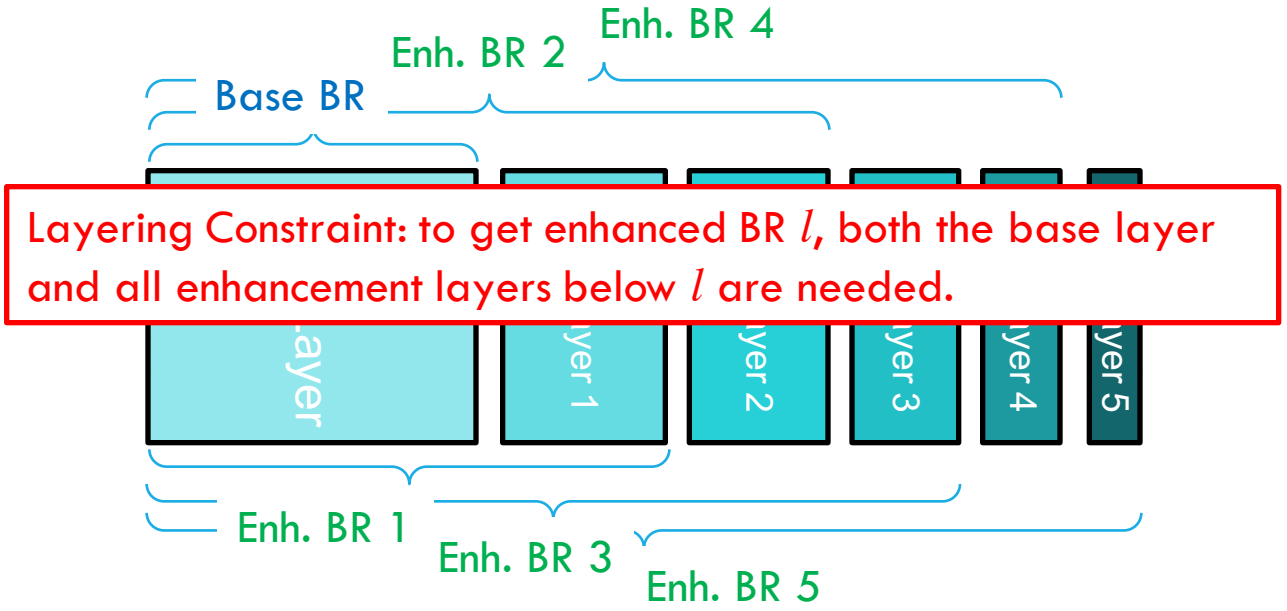


Scalable Video Coding (SVC)

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



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 $d_{i,\iota}$

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 β_j^l

$\psi_\iota^{j,l}$: number of users at BS B_ι, 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

$z_{i,\iota}^{j,l}$: number of video V_j's layer l requested by users from BS B_ι 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:

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

- d_ι^{j}: aggregated delay received by video V_j's user(s) at BS B_ι

Constraints

- **Capacity constraints:**

$$\sum_{j=1}^{N}\sum_{l=1}^{L_j} x_i^{j,l} \cdot s_j^l \leq c_i,\ \forall B_i \in \mathcal{B}$$

$$\sum_{\iota|\iota\neq i}\sum_{j=1}^{N}\sum_{l=1}^{L_j} z_{i,\iota}^{j,l} \cdot \beta_j^l \leq b_i^u,\ \forall B_i \in \mathcal{B}$$

$$\sum_{i|i\neq \iota}\sum_{j=1}^{N}\sum_{l=1}^{L_j} z_{i,\iota}^{j,l} \cdot \beta_j^l \leq b_\iota^d,\ \forall B_\iota \in \mathcal{B}$$

Constraints cont.

- **Caching & scheduling constraints:**

$$z_{i,\iota}^{j,l} \leq x_i^{j,l} \cdot \psi_\iota^{j,l},\ \forall B_i \in \mathcal{B}, B_\iota \in \mathcal{B}, V_j \in \mathcal{V}, l \in [1, L_j]$$

$$\sum_{i=0}^{M} z_{i,\iota}^{j,l} \leq \psi_\iota^{j,l},\ \forall B_\iota \in \mathcal{B}, V_j \in \mathcal{V}, l \in [1, L_j]$$

$$\sum_{i=0}^{M} z_{i,\iota}^{j,l} \leq \sum_{i=0}^{M} z_{i,\iota}^{j,l-1},\ \forall B_\iota \in \mathcal{B}, V_j \in \mathcal{V}, l \in [2, L_j]$$

- **Delay constraints:**

$$\lambda_{i_p^\iota,\iota}^{j} = \max_{l \in \mathcal{L}_j} \left\{ \sum_{q=0}^{p} z_{i_q^\iota,\iota}^{j,l} \right\} - \sum_{q=0}^{p-1} \lambda_{i_q^\iota,\iota}^{j},$$

$$\forall B_\iota \in \mathcal{B}, V_j \in \mathcal{V}, p \in \{0, \cdots, M\}$$

Linearized as inequalities

$$d_\iota^j = \sum_{i=0}^{M} \lambda_{i,\iota}^{j} \cdot d_{i,\iota},\ \forall B_\iota \in \mathcal{B}, V_j \in \mathcal{V}$$

Agenda

Introduction

Problem Formulation

Solution Design

Performance Evaluation

Conclusions

Algorithm Overview

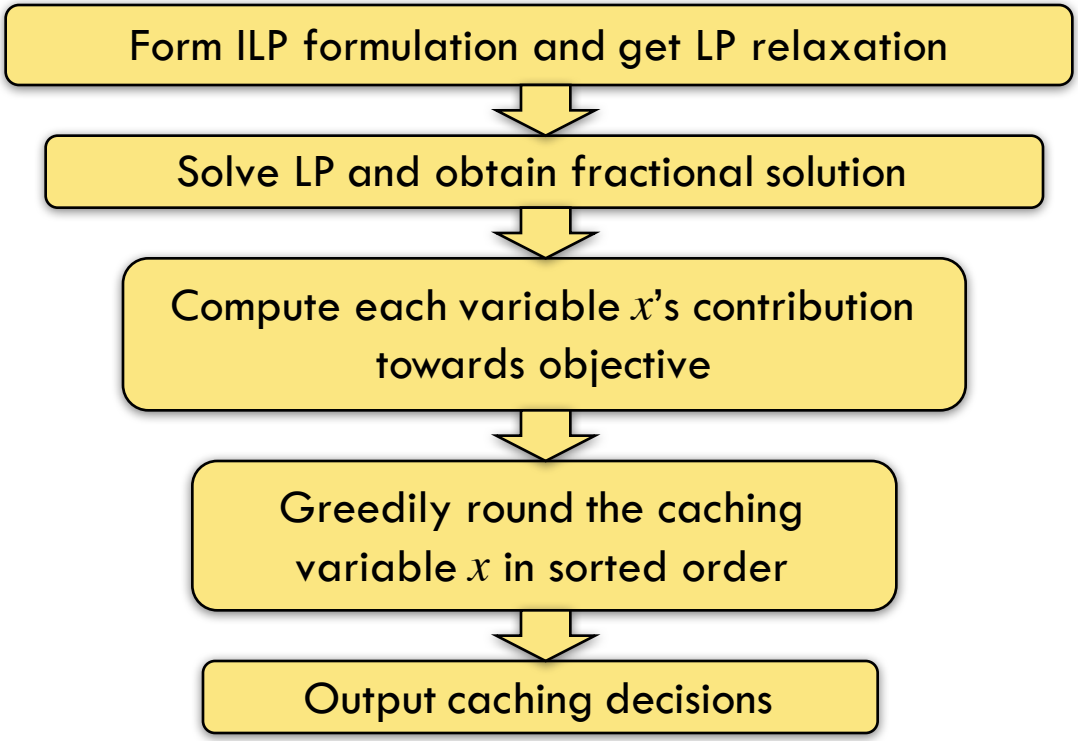
Two stages:

- 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.

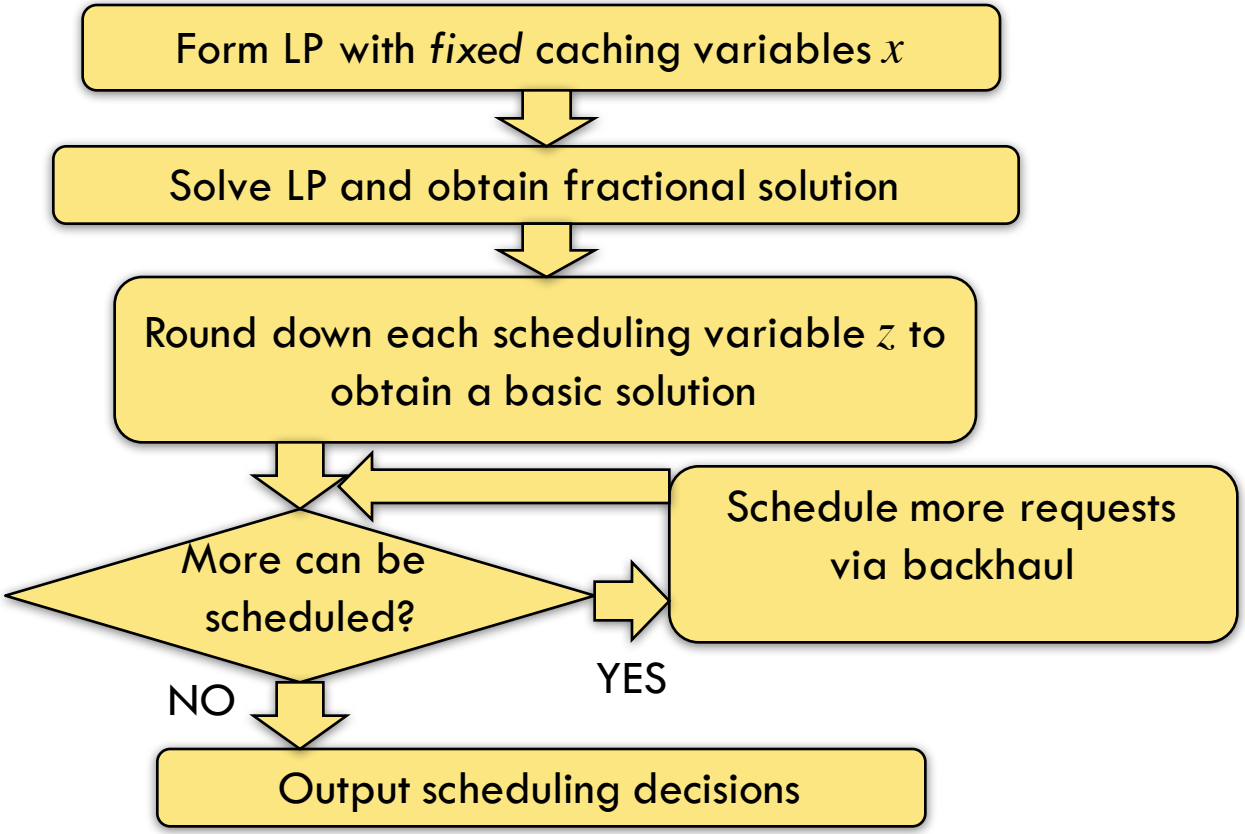
Rounding-based algorithm:

- Relax the ILP formulation to LP;
- Solve for a (fractional) solution;
- Use deterministic rounding technique to obtain an integral solution.

Stage 1: Video Caching



Stage 2: Video Scheduling



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: γ=0.95
- 5 layers per video: 10% coding overhead introduced
- Randomly generated cache, video size and bandwidth capacity/demands

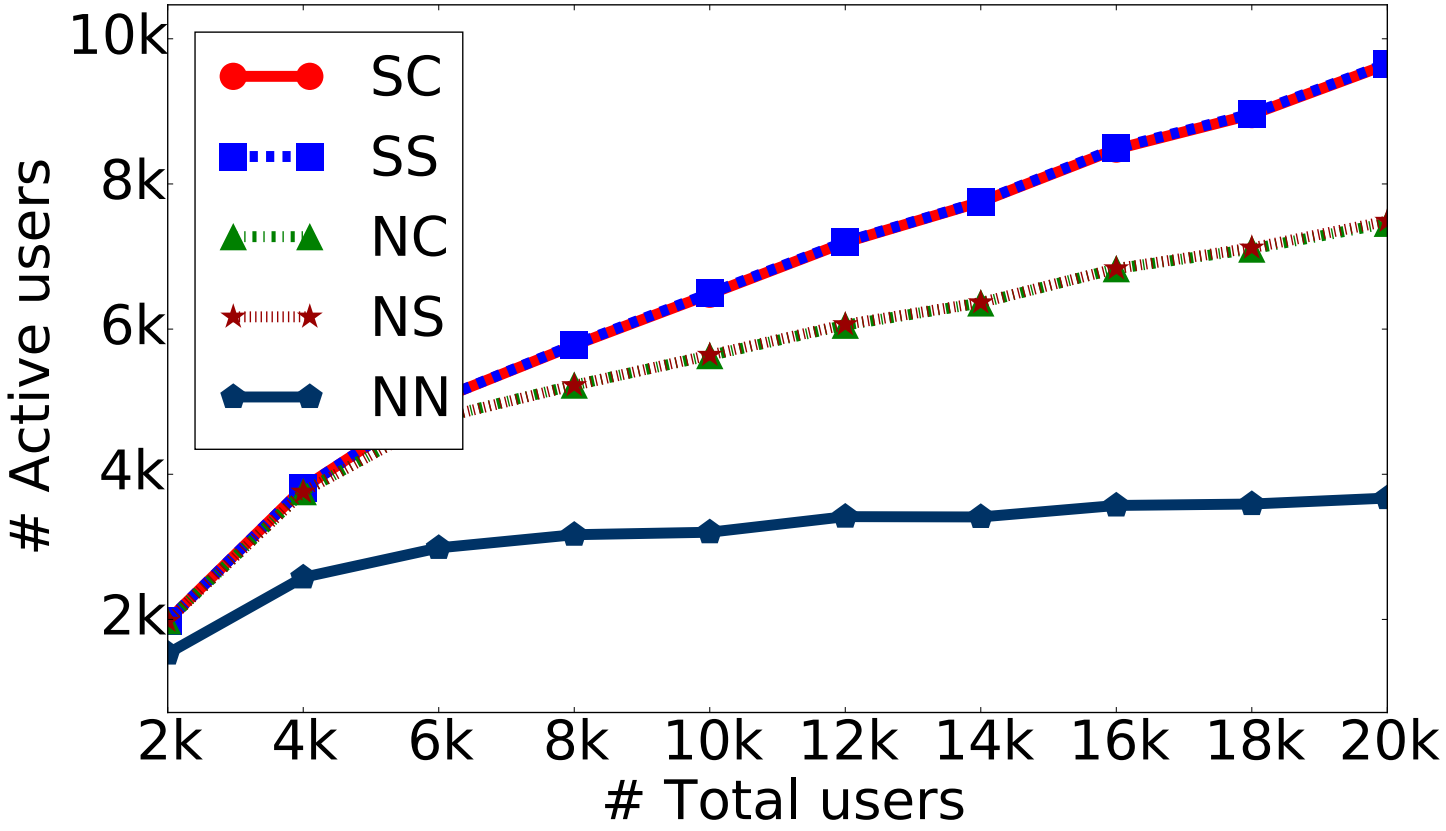
Five schemes for comparison:

- 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

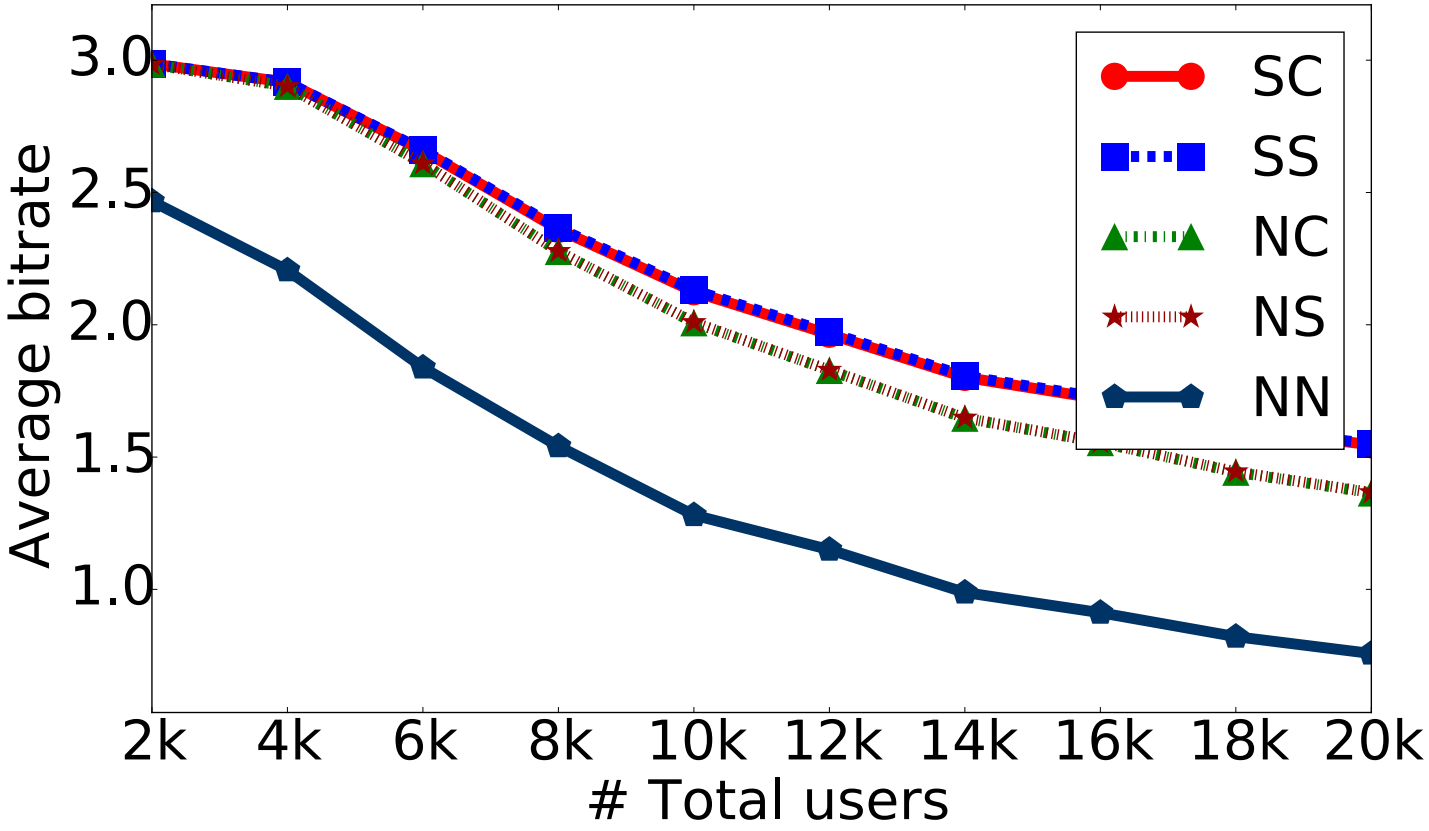
Exp. With Default Parameters

Combination	Act users	Avg layers	Avg delay
SVC + Collaborative (SC)	6480.9	2.1249	88.31
SVC + Non-collaborative (SS)	6492.1	2.1306	102.14
Non-SVC + Collaborative (NC)	5635.2	2.0077	103.80
Non-SVC + Non-collabor. (NS)	5642.4	2.0092	115.90
No caching (NN)	3201.1	1.2808	143.41

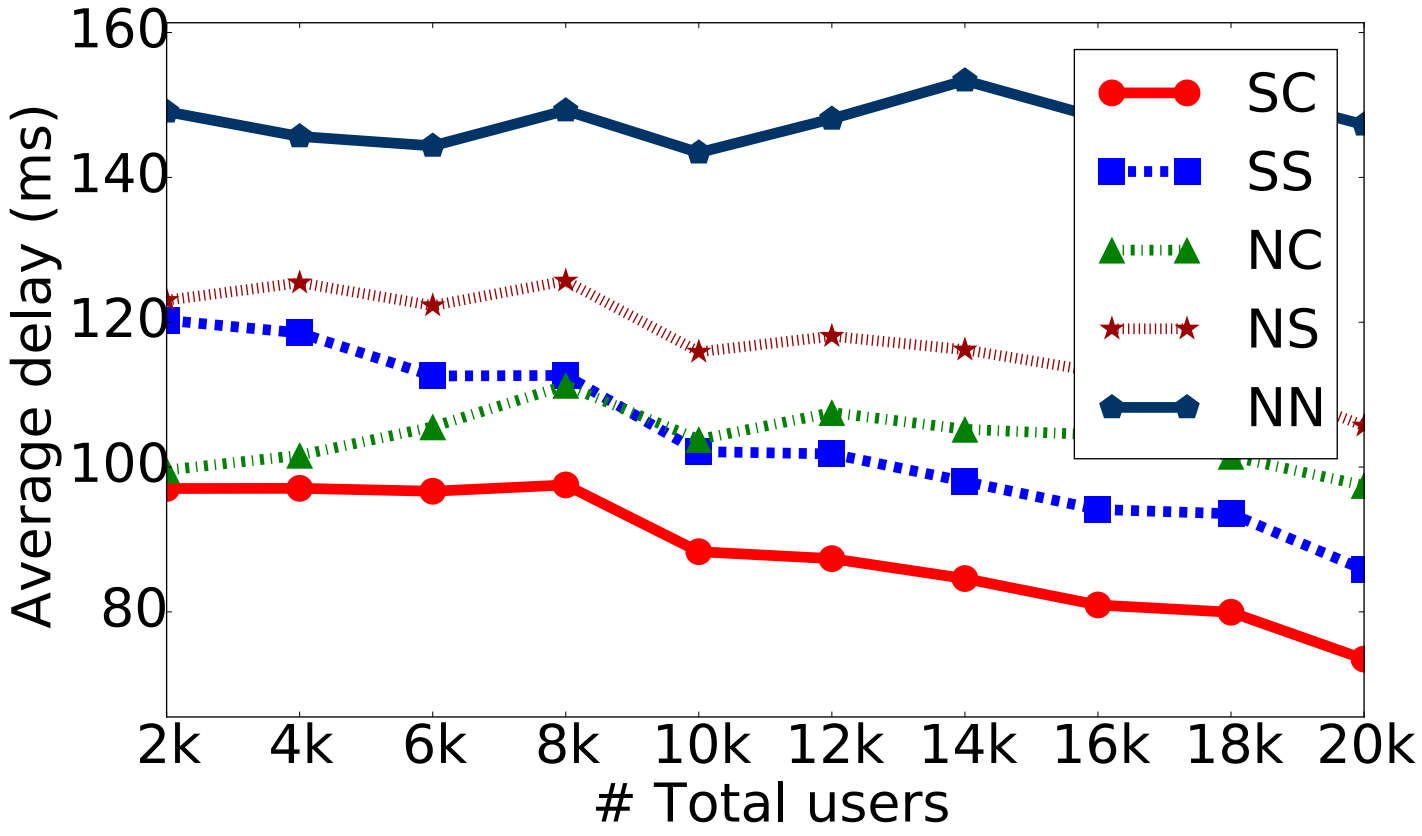
Active Users



Average Layers (Bitrates)



Average Delay



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.

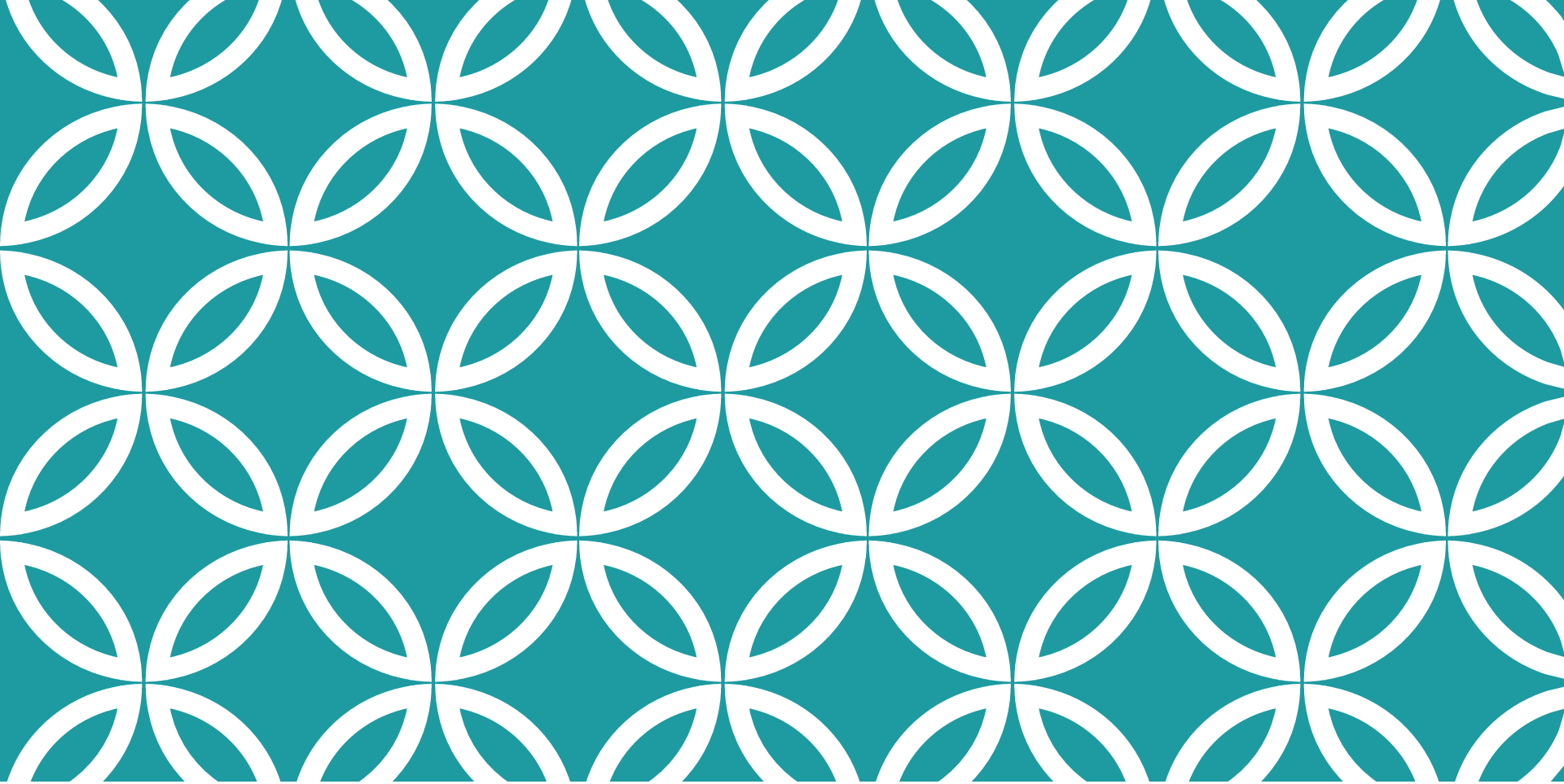
Maximizing rewards and minimizing delay: a joint problem.

- NP-hard

2-stage rounding-based algorithm.

- Decide caching first.
- Schedule videos based on caching.

Outperforms using either collaborative caching or SVC alone.



Thank You

Q&A