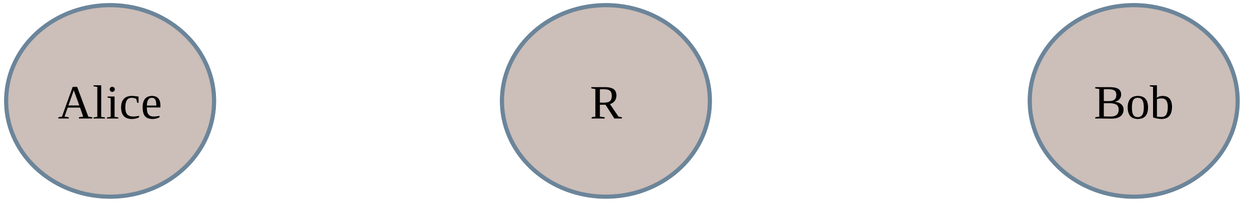
CACHE CONTENT PLACEMENT USING TRIANGULAR NETWORK CODING

Pouya Ostovari, Abdallah Khreishah, and Jie Wu
Computer & Information Sciences Department,
Temple University,
USA

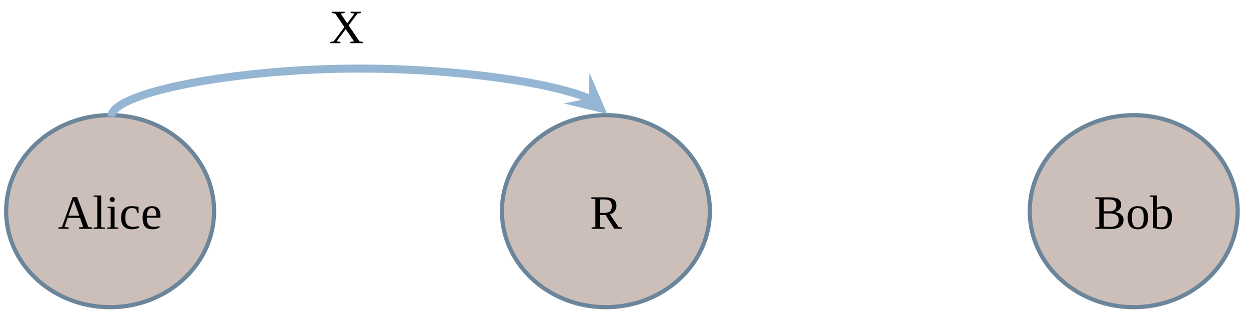
Agenda

- Introduction
- Motivation
- Content placement algorithm
- Simulation
- Conclusion

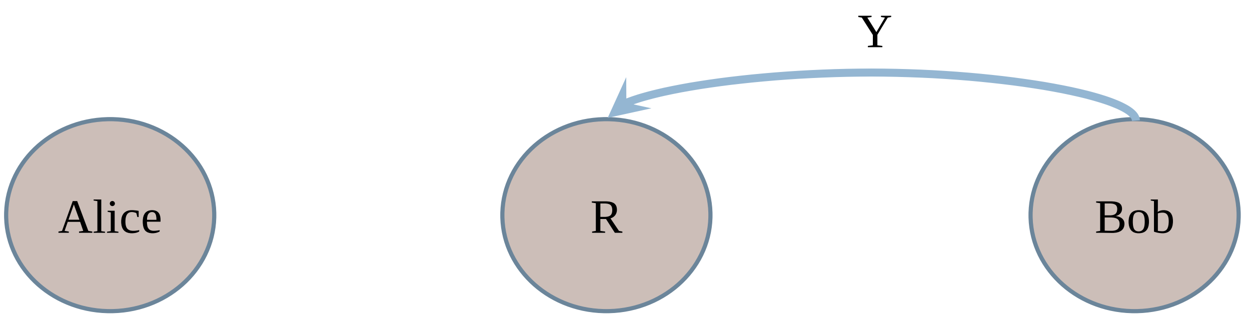
Alice and Bob (No coding)



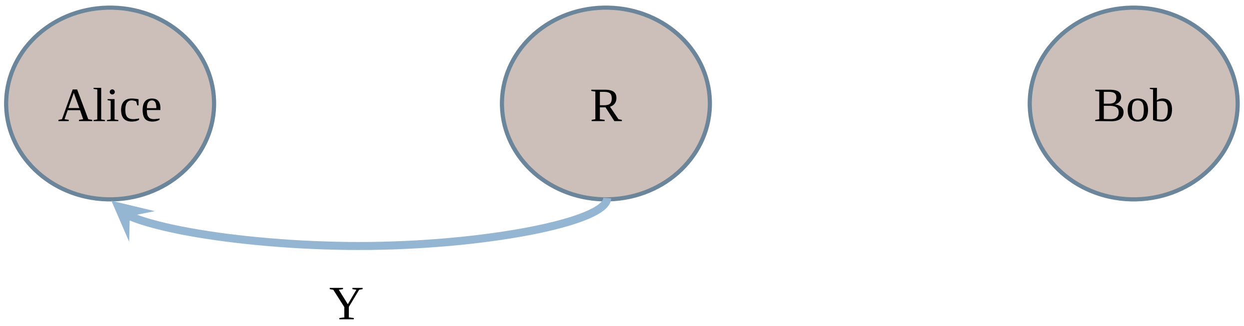
Alice and Bob (No coding)



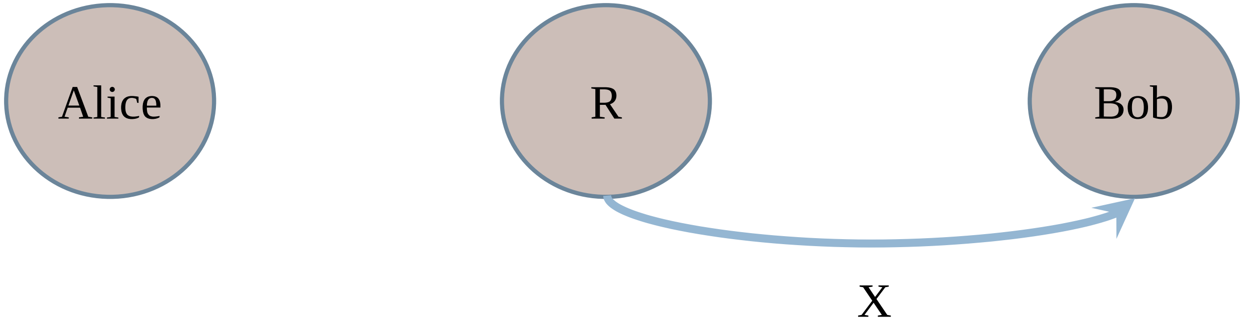
Alice and Bob (No coding)



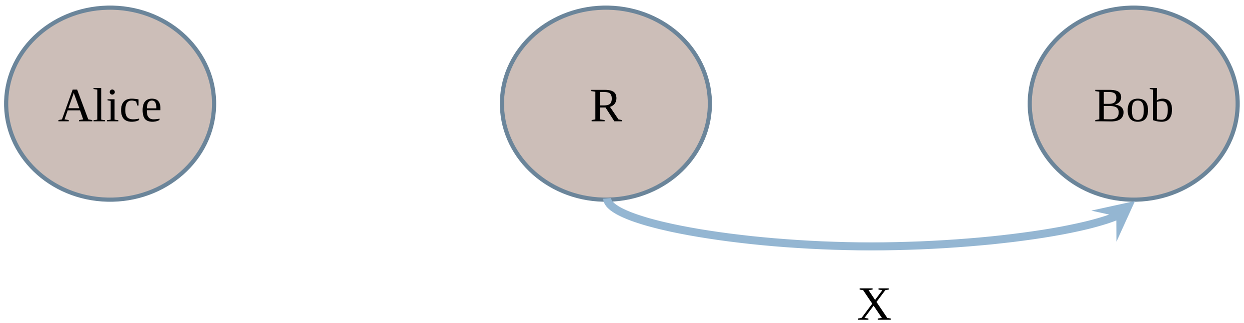
Alice and Bob (No coding)



Alice and Bob (No coding)

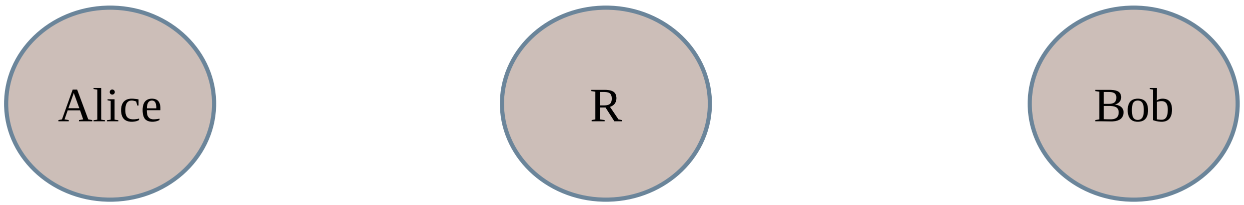


Alice and Bob (No coding)

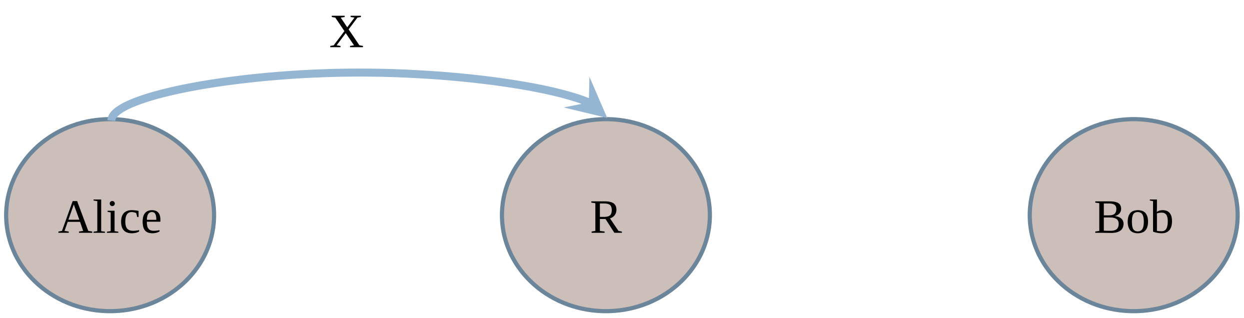


4 transmissions

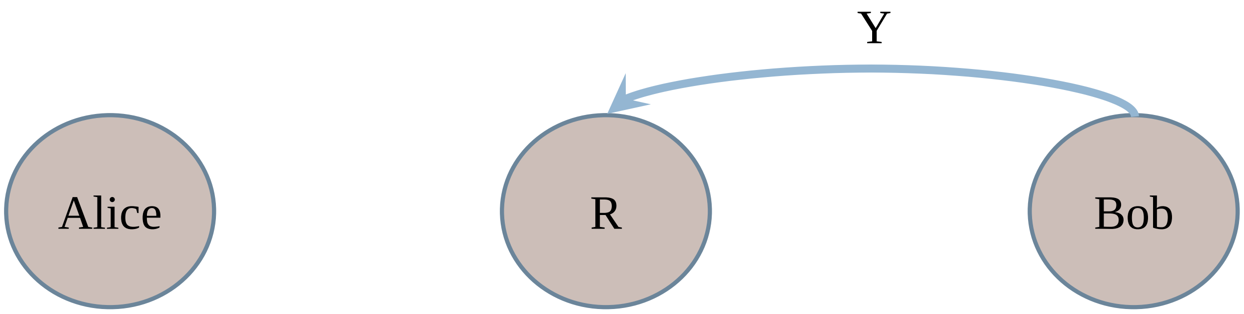
Alice and Bob (Coding)



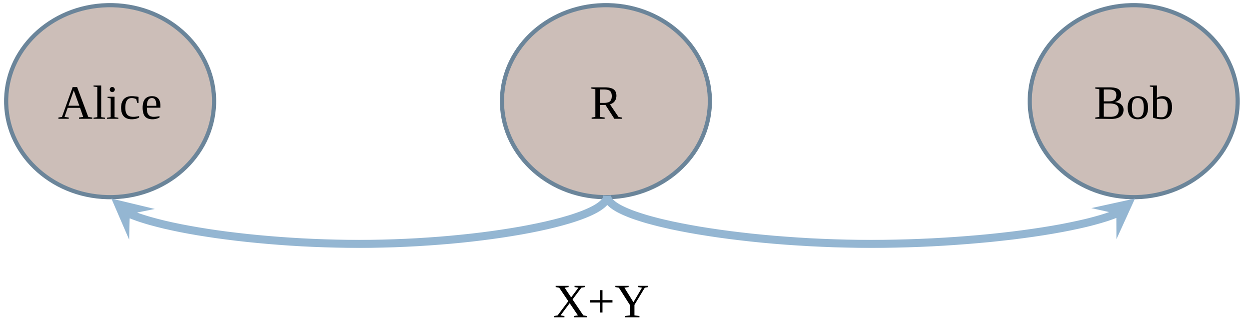
Alice and Bob (Coding)



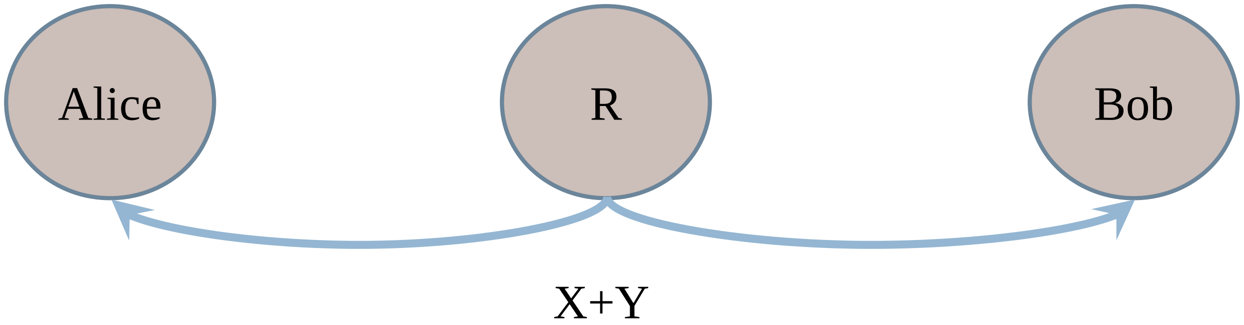
Alice and Bob (Coding)



Alice and Bob (Coding)



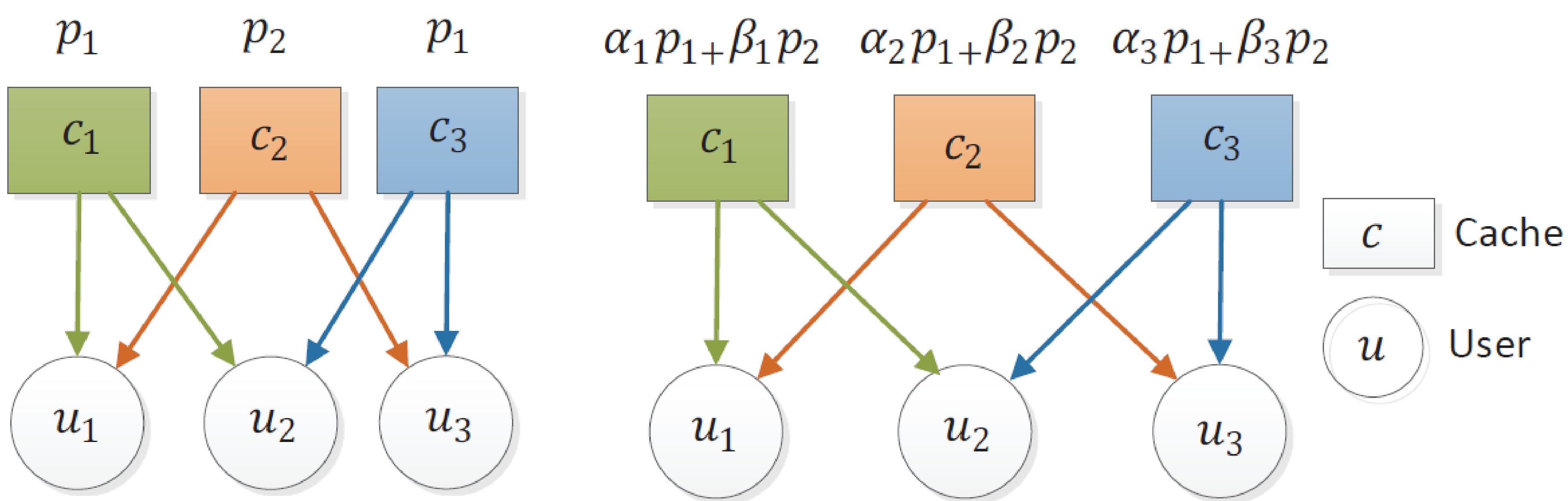
Alice and Bob (Coding)



3 transmissions

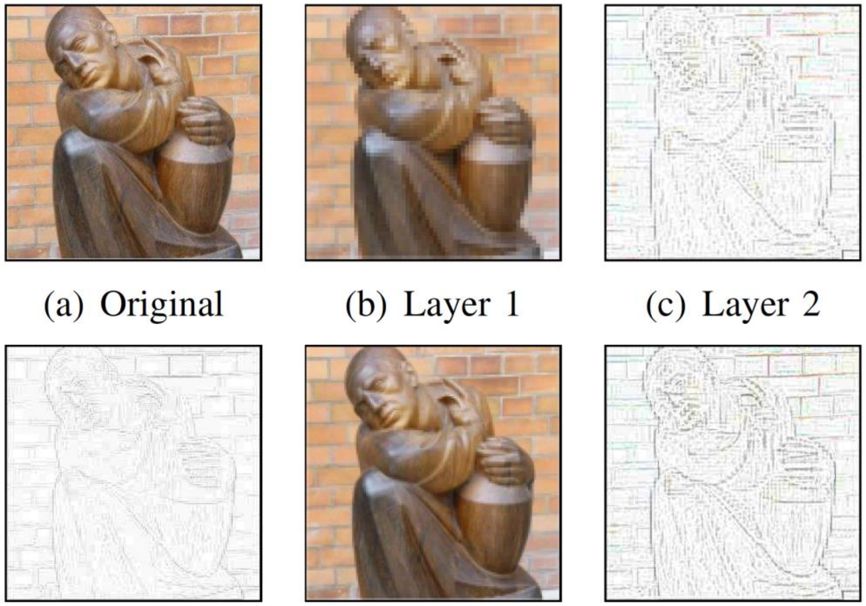
Motivation

- Providing more amount of data to the users.



Setting

- h video layers on the server: $p_1, .., p_h$
- Layer p_i is not useful without the layers with a smaller index.



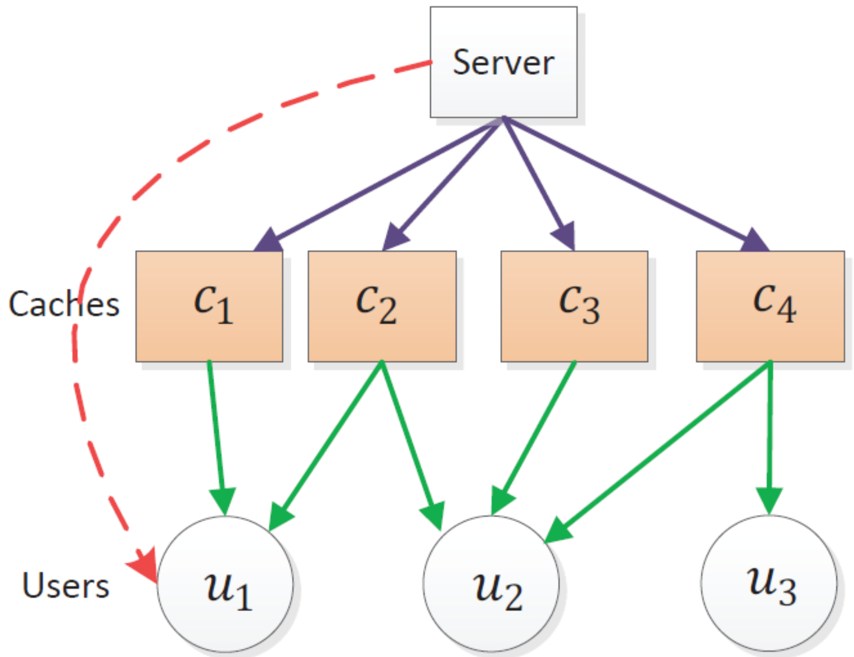
(a) Original (b) Layer 1 (c) Layer 2

(d) Layer 3 (e) Layers 1 & 2 (f) Layers 2 & 3

Setting

- Capacity=size of the video layers
- Objective: maximizing the total number of available layers.

$$\max \sum_{i=1}^{h} \sum_{j=1}^{n} z_{ij}$$



Triangular Coding

- Linear Coding
  - $2^h - 1$ ways to code *h* layers.
  - $(2^h - 1)^n$ different possible placements for *n* caches.
- Triangular network coding
  - The encoded video layers are in the form $\sum_{j=1}^{k} \alpha_j p_j$.

Original packets

$$\begin{cases} p_1 \\ p_2 \\ p_3 \end{cases}$$

Linear coding

$$\begin{cases} p_1,\ p_2,\ p_3 \\ p_1 + p_2,\ \ p_1 + p_3,\ \ p_2 + p_3 \\ p_1 + p_2 + p_3 \end{cases}$$

Triangular coding

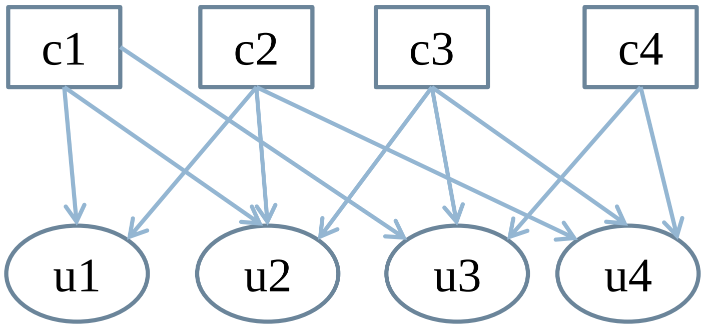
$$\begin{cases} p_1 \\ p_1 + p_2 \\ p_1 + p_2 + p_3 \end{cases}$$

Content Placement Algorithm

- The problem of efficient content placement on the caches is an NP-complete problem.
- The greedy algorithm fills-up the caches in rounds.
- In each round, we select a user and fill-up its adjacent caches.
- Selection rules
  - **Rule 1**: the user with the minimum degree.
  - **Rule 2**: the user with a larger number of filled-up caches.
  - **Rule 3**: the user whose adjacent caches have less cumulative ranks.
- The algorithm fills-up the empty adjacent caches to user u_i with a random linear combination of the first $d_i - v_i + r_i$ video layers.

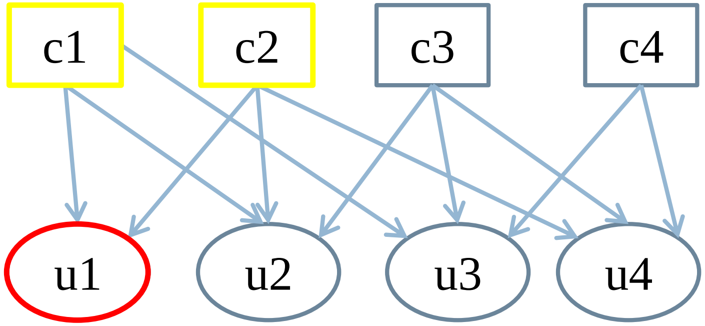
Example

- Step 1: user u_1 has the minimum degree.
  - 2-0+0=2
- Step 2: user u_2 has 2 filled adjacent caches.
  - 3-2+2=3
- Step 3: select u_3 or u_4 randomly (assume u_3).
  - 3-2+2=3



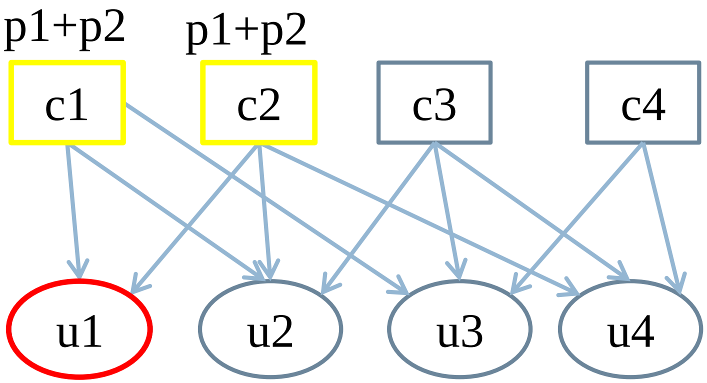
Example

- Step 1: user u_1 has the minimum degree.
  - 2-0+0=2
- Step 2: user u_2 has 2 filled adjacent caches.
  - 3-2+2=3
- Step 3: select u_3 or u_4 randomly (assume u_3).
  - 3-2+2=3



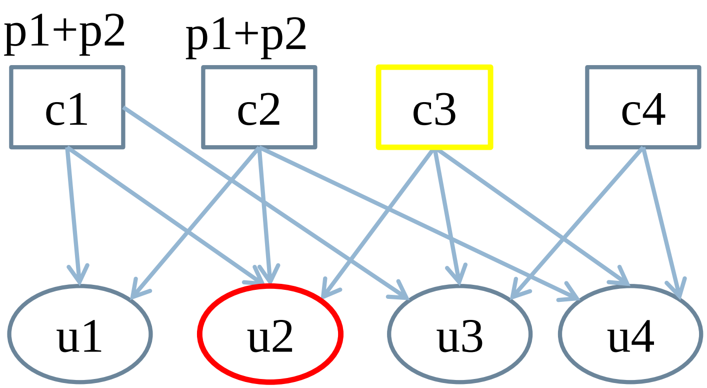
Example

- Step 1: user u_1 has the minimum degree.
  - 2-0+0=2
- Step 2: user u_2 has 2 filled adjacent caches.
  - 3-2+2=3
- Step 3: select u_3 or u_4 randomly (assume u_3).
  - 3-2+2=3



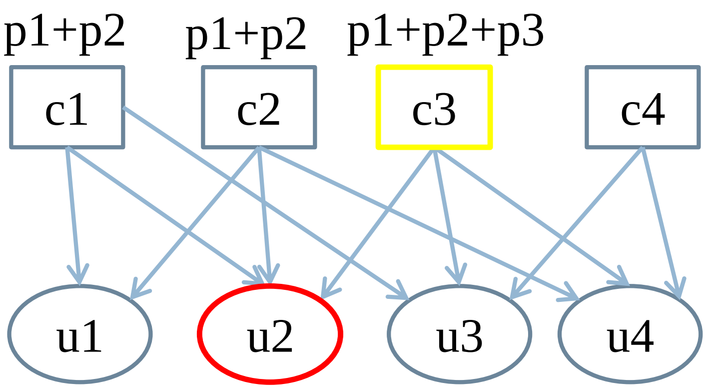
Example

- Step 1: user u_1 has the minimum degree.
  - 2-0+0=2
- Step 2: user u_2 has 2 filled adjacent caches.
  - 3-2+2=3
- Step 3: select u_3 or u_4 randomly (assume u_3).
  - 3-2+2=3



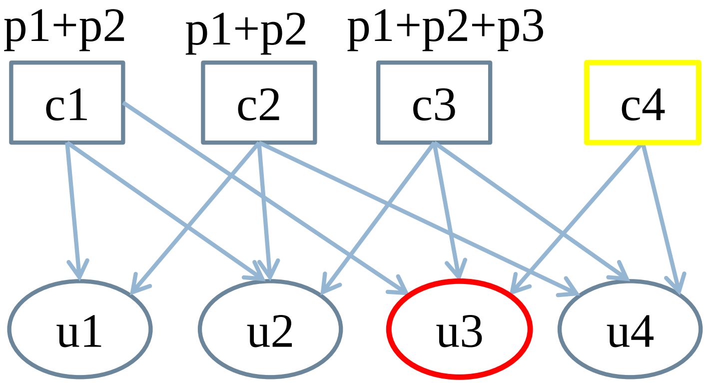
Example

- Step 1: user u_1 has the minimum degree.
  - 2-0+0=2
- Step 2: user u_2 has 2 filled adjacent caches.
  - 3-2+2=3
- Step 3: select u_3 or u_4 randomly (assume u_3).
  - 3-2+2=3



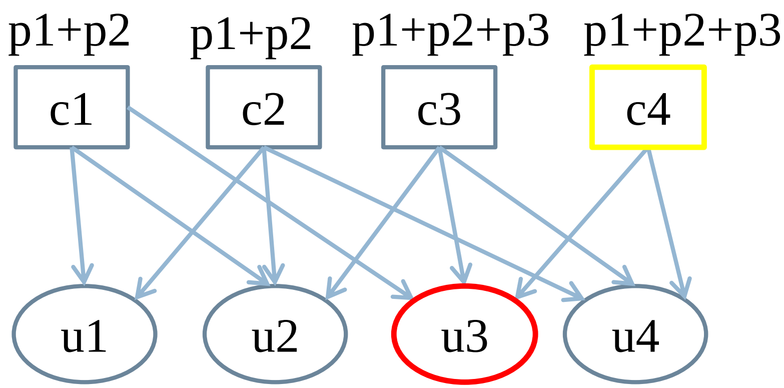
Example

- Step 1: user u_1 has the minimum degree.
  - 2-0+0=2
- Step 2: user u_2 has 2 filled adjacent caches.
  - 3-2+2=3
- Step 3: select u_3 or u_4 randomly (assume u_3).
  - 3-2+2=3



Example

- Step 1: user u_1 has the minimum degree.
  - 2-0+0=2
- Step 2: user u_2 has 2 filled adjacent caches.
  - 3-2+2=3
- Step 3: select u_3 or u_4 randomly (assume u_3).
  - 3-2+2=3



Simulation Setting

- Simulator in the MATLAB environment.
- Comparison
  - Number of available layers to the users.
  - Average utility: the number of available layers to a user divided by its degree.
  - Fairness: we define unfairness as the average difference between the number of available layers to each user and the average number of available layers to the users.

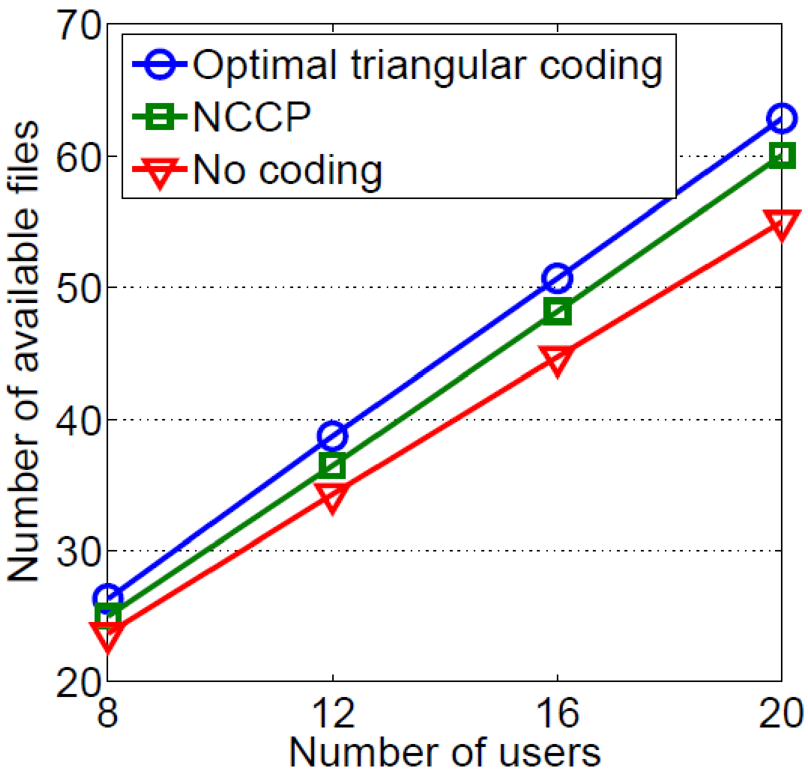
$$f' = \frac{\sum_{i=1}^{m} |q_i - e|}{m}$$

$$e = \frac{\sum_{i=1}^{m} q_i}{m}$$

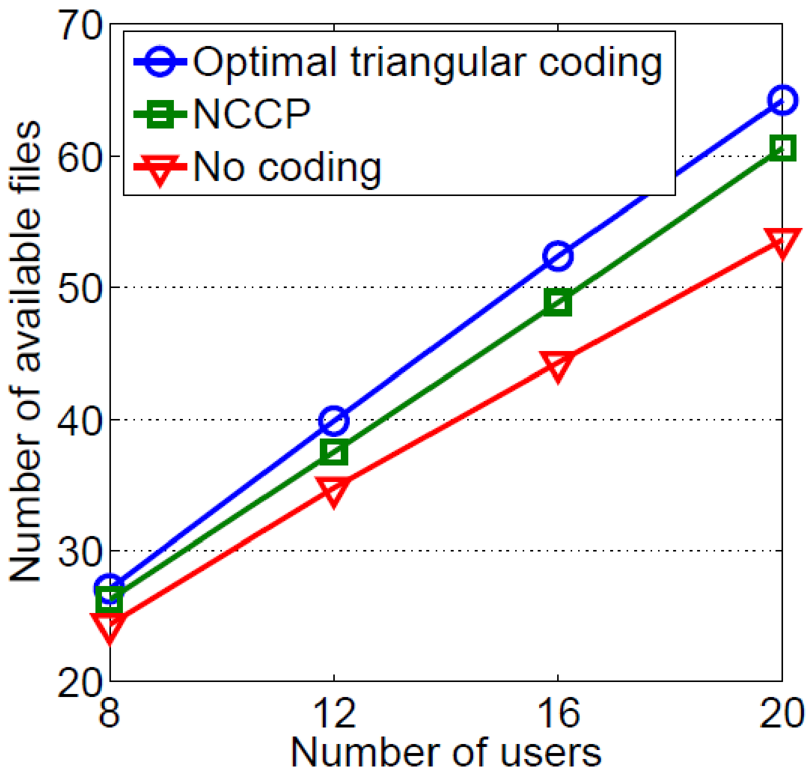
$$f = \frac{1}{f'}$$

Simulations

- Number of caches: 5
- Number of layers: 4

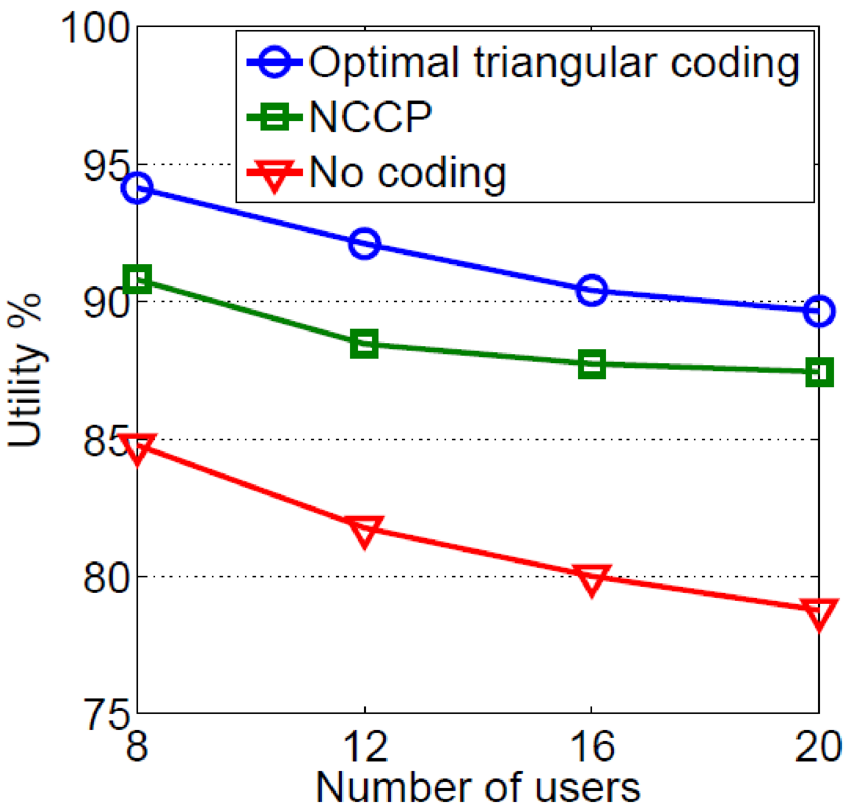


- Number of caches: 5
- Number of layers: 4

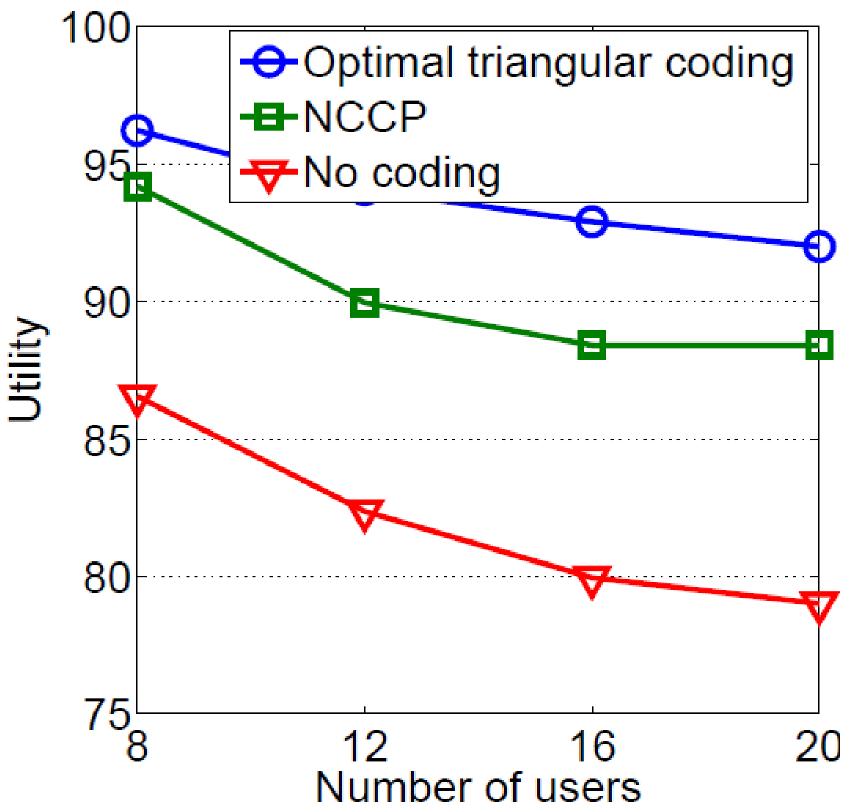


Simulations

- Number of caches: 5
- Number of layers: 4

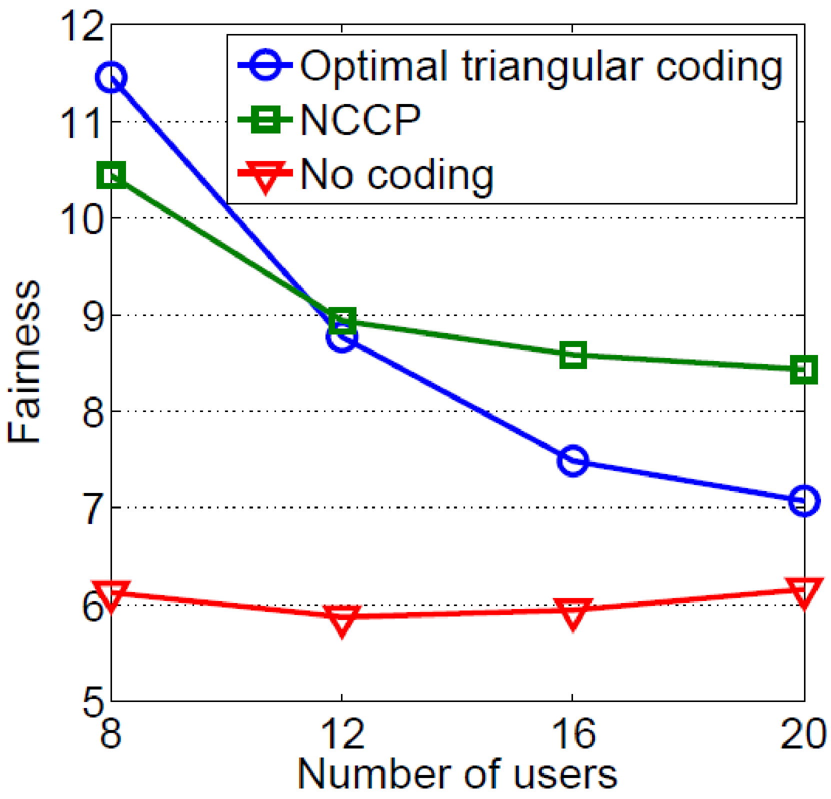


- Number of caches: 5
- Number of layers: 4

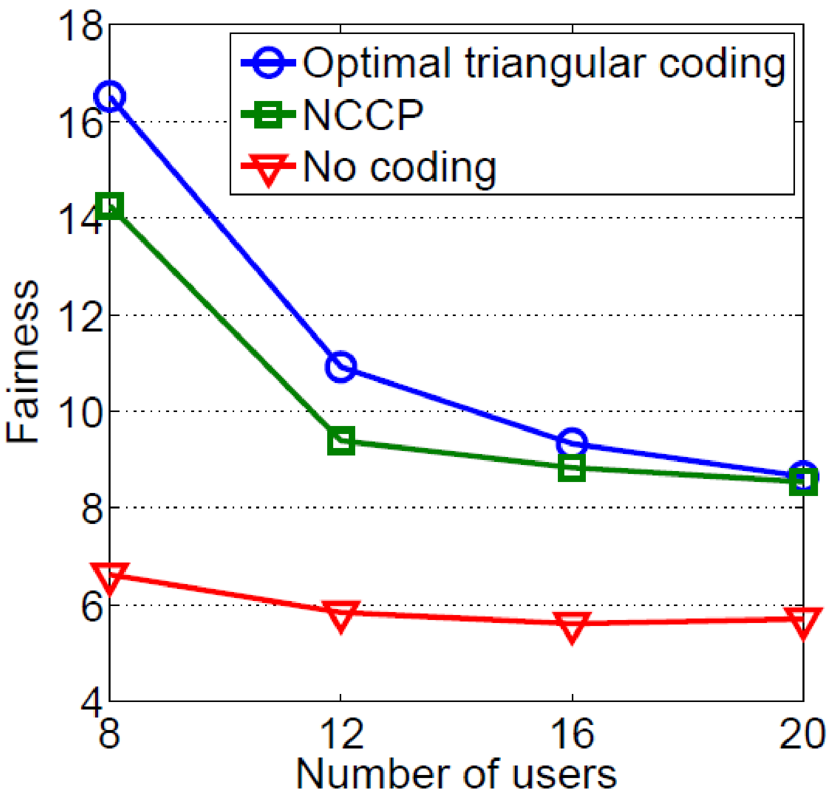


Simulations

- Number of caches: 5
- Number of layers: 4



- Number of caches: 5
- Number of layers: 4



Summary

- The problem of efficient content placement.

on the caches is known as an NP-complete problem.

- Triangular network coding can reduce the complexity of content placement compared to the general form of coding.
- We propose a heuristic algorithm to solve the problem.

Questions