Delay-Aware Coded Caching for Mobile Users

Emre Ozfatura*, Thomas Rarris*, Deniz Gündüz*, and Ozgur Ercetin†
*Information Processing and Communications Lab
Department of Electrical and Electronic Engineering
Imperial College London
†Sabanci University

11 September 2018

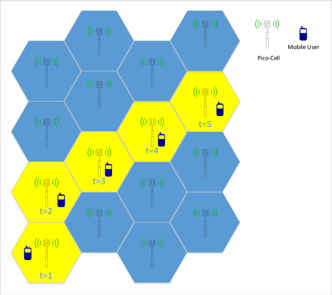
This work was supported in part by the European Union's Horizon 2020 research and innovation programme under the Marie Sktodowska-Curie grant agreement No 690893.

Motivation

- Video dominates Internet traffic:
  - In 2016, Youtube was responsible for %21 of mobile Internet traffic in North America (Sandvine 2016).
  - By 2021 size of Internet video traffic will be 4 times larger (Cisco 2017).
- Small number of viral video files are viewed by many users.
- Decreasing cost of high capacity storage.
- Densification of small-cells: mobility-aware coded storage

E. Ozfatura and D. Gunduz, Mobility and popularity-aware coded small-cell caching, IEEE Communications Letters, vol. 22, no. 2, pp. 288 - 291, Feb. 2018.

Network Model



- One macro base station (MBS)
- N small-cell base stations (SBSs) with a cache memory of size C bits.
- Library of K files, $\mathbb{V} = \{v_1, \ldots, v_K\}$, each of size F bits.
- v_k is the kth most popular file with request probability p_k.
- Within one time slot, MU can download B bits from a SBS.
- Duration of one streaming session: $T = F/B$ slots.

Video Streaming

- Mobility path: Sequence of SBSs visited within one streaming session.
- High mobility scenario: A mobile user (MU) connects to each SBS at most one time slot; that is, MU connects to T SBSs within one streaming session.
- Video display rate $\lambda = B$, i.e., it takes T time slots to play the downloaded file.
- Display can start before downloading all the segments of a file.
- Buffer starvation: MU buffer is empty.
- Rebuffering delay: When buffer starvation occurrs video display is frozen until next segment is downloaded.

Maximum Distance Separable (MDS) Codes

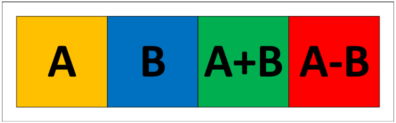


Figure: (2,4) MDS code

- Consider 4 SBSs and a user that can connect to any 2 over time ($T = 2$).
- Each file is divided into 2 fragments. Fragments are encoded into 4 fragments through a $(2, 4)$ MDS code.
- Each SBS caches a different coded fragment.
- User can recover the file by connecting to any 2 SBSs.

Conventional Coded Storage

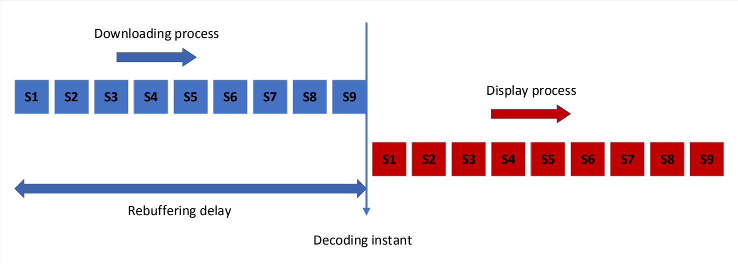


Figure: Video streaming process with conventional coded storage for $T = 9$

- Each file divided into T segments, and coded with (T, N) MDS code.
- Each SBS stores one coded segment (F/T bits) for each file.
- Rebuffering delay is T time slots

Delay-Aware Encoding

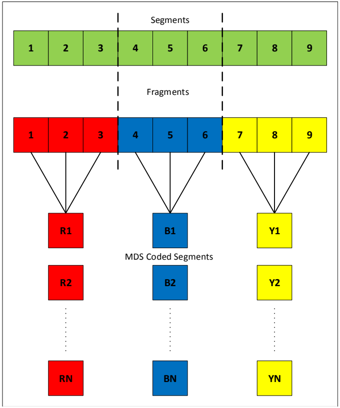


Figure: Video encoding for $T = 9$ and $M = 3$

- Segments are grouped into M disjoint fragments.
- Segments in each fragment are encoded with $(T/M, N)$ MDS code.

Delay-aware Coded Storage

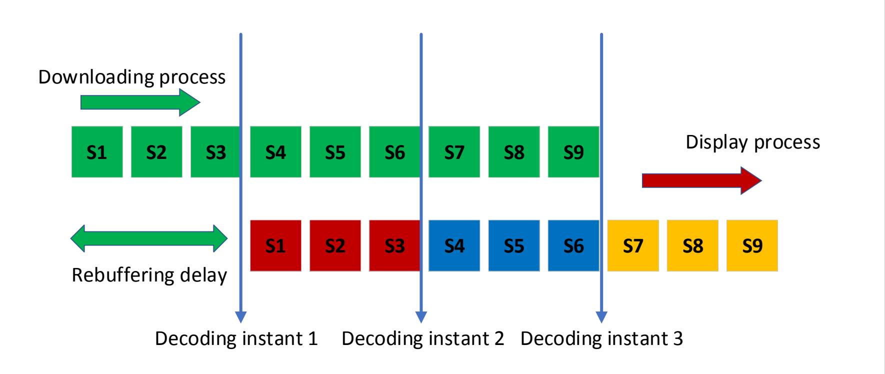


Figure: Video streaming with delay-aware coded storage for $T = 9$ and $M = 3$

- Each SBS stores $MF/T = MB$ bits for each file, and rebuffering delay is $\lceil T/M \rceil$ time slots

Delay-Memory Trade-off

- Delay-cache capacity function $\Omega(M) \triangleq \lceil T/M \rceil$:maps the number of fragments M to the rebuffering delay $D \in \mathcal{Z}^{+}$ (slots).
- $\Omega(M)$ is a monotonically decreasing step function.
- Delay levels, $D^{(l)}$: Possible values taken by $\Omega(M)$.
- Decrement points, $m^{(l)}$: minimum M that satisfies $\Omega(M) = D^{(l)}$

Delay-Memory trade-off

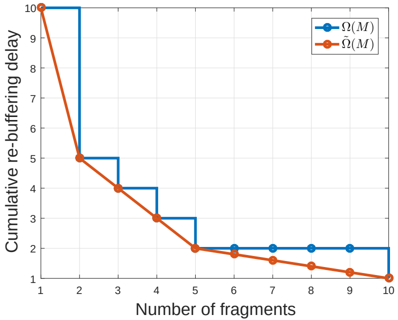


Figure: Delay-cache capacity function and its piece-wise linear approximation for $T = 10$

- $D^{(1)} = 10,\ D^{(2)} = 5,\ D^{(3)} = 4,\ D^{(4)} = 3,\ D^{(5)} = 2,\ D^{(6)} = 1$
- $m^{(1)} = 1,\ m^{(2)} = 2,\ m^{(3)} = 3,\ m^{(4)} = 4,\ m^{(5)} = 5,\ m^{(6)} = 10$

Problem Formulation

- For file v_k, expected delay-cache capacity function:

$$\Omega_k(M_k) \triangleq p_k \lceil T/M_k \rceil$$

- Average rebuffering delay, $\mathbf{M} = (M_1, \ldots, M_K)$

$$D_{avg}(\mathbf{M}) = \sum_{k=1}^{K} \Omega_k(M_k)$$

- D_{max}: Maximum allowable delay for a video file.

$$\begin{aligned} \textbf{P1:} \quad & \min_{\mathbf{M}} \; D_{avg}(\mathbf{M}) \\ \text{subject to:} \quad & D_k(M_k) \leq D_{max}, \; \forall k, \\ & \sum_{k=1}^{K} M_K B \leq C. \end{aligned}$$

Solution Approach

Observation

- Replacing $\Omega_k(M_k)$ with its linear approximation $\tilde{\Omega}_k(M_k)$, **P1** becomes a convex optimization problem
- Let $\gamma_{k,l}$ be the slope of $\tilde{\Omega}_k(M_K)$ in interval $(m^{(l)}, m^{(l+1)}]$, and an approximately optimal solution found in polynomial time by sorting $\gamma_{k,l}$

Lemma

- Approximate solution is equivalent to the optimal if M_k is equal to some decrement point $m^{(l_k)}$ for each k.
- Otherwise, the optimal solution can obtained by increasing cache size C by at most $\epsilon \leq F/2$.

Cost-Aware Delay-Constrained Caching

Problem Definition

- It may not be possible to satisfy maximum delay constraint D_{max} for all files.
- We can also impose a QoS constraint $\bar{D}_{max}$ on $D_{avg}(\mathbf{M})$.
- Some files not cached at SBSs and served directly by MBS with an additional cost.
- Objective: Minimize cost while satisfying constraints $\bar{D}_{max}$ and D_{max}

Solution Approach

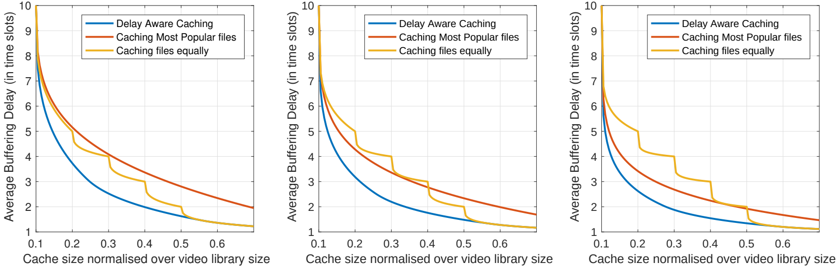
- At each iteration remove the least popular file and apply the delay-aware coded caching method. Continue until constraints D_{max} and $\bar{D}_{max}$ are met.

Simulation Results: Benchmarks

- Cache most popular files: First, all files are cached to meet D_{max}, then starting from the most popular file, rebuffering delays are sequentially reduced to minimum.

- Cache files equally: All files are cached with same delay.

Minimizing Average Rebuffering Delay

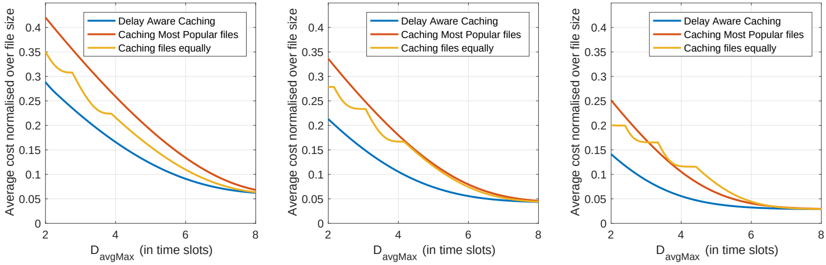


(a) $w = 0.75$ (b) $w = 0.85$ (c) $w = 0.95$

Figure: Average buffering delay versus cache size for $D_{max} = 10$ slots

- Video library of 10000 files: popularity is modeled using a Zipf distribution with coefficient $w \in \{0.75, 0.85, 0.95\}$.
- Video download duration $T = 10$ slots, and $D_{max} = 10$ time slots.
- Consider cache sizes $\hat{C} \in [0.1, 0.7]$.

Minimizing Average Cost



(a) $w = 0.75$ (b) $w = 0.85$ (c) $w = 0.95$

Figure: Average cost versus maximum average delay constraint for $T = 10$ slots

- Set cache size $\hat{C} = 0.08$ and $D_{max} = 10$.
- Consider $\bar{D}_{max} \in [2, 8]$.

Conclusions

- Analyzed storage-delay trade-off focusing on continuous video streaming.
- Introduced fragment-based coded caching to reduce rebuffering delay
- Future directions:
  - Consider more general user mobility models.
  - Consider a video display rate higher than the download rate from SBSs.