Adding transmitters dramatically boosts coded-caching gains for finite file sizes

(a.k.a. A powerful way for reducing the subpacketization bottleneck)

The original setting/model of in Coded caching

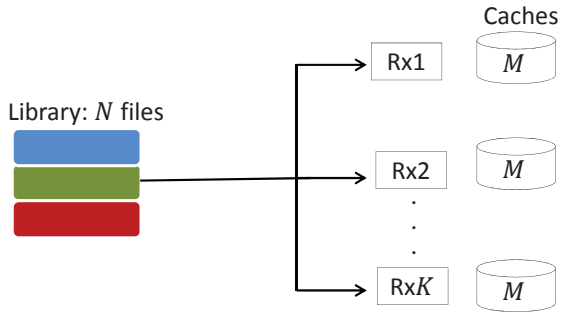


Figure 1: Cache-aided BC. One transmitter, serving K receivers via a single-shared link.

- Original setting in single-stream broadcast channel (BC)
- Single-antenna transmitter has access to a library of N files
- Serves K receivers, each with a cache of size equal to size of M files.

Normalized cache size per user

$$\gamma \triangleq M/N$$

Performance of Coded caching

- Worst-case completion time which is at most

$$T = K(1-\gamma)/(1+K\gamma)$$

- Sum-DoF (users served at a time)

$$d_1(\gamma) = K(1-\gamma)/T = 1 + K\gamma$$

THEORETICAL CACHING GAIN

$$G = d_1(\gamma) - d_1(\gamma = 0) = K\gamma$$

- Caching gain represents the number of extra users that could be served at a time, additionally, as a consequence of introducing caching.

CACHING GAINS ARE THEORETICALLY UNBOUNDED

Reason for massive theoretical gain

- Coded caching bypasses inherent inefficiency of traditional caching
- Traditional caching:
  - ⋆ Receiver uses the cached fraction of just the one single file that she requested
  - ⋆ Leaves all other cache information unused.
- Coded caching:
  - ⋆ Each receiver exploits the cached fraction of all K requested files
  - ⋆ The cached content of its own requested file: traditional local caching gain

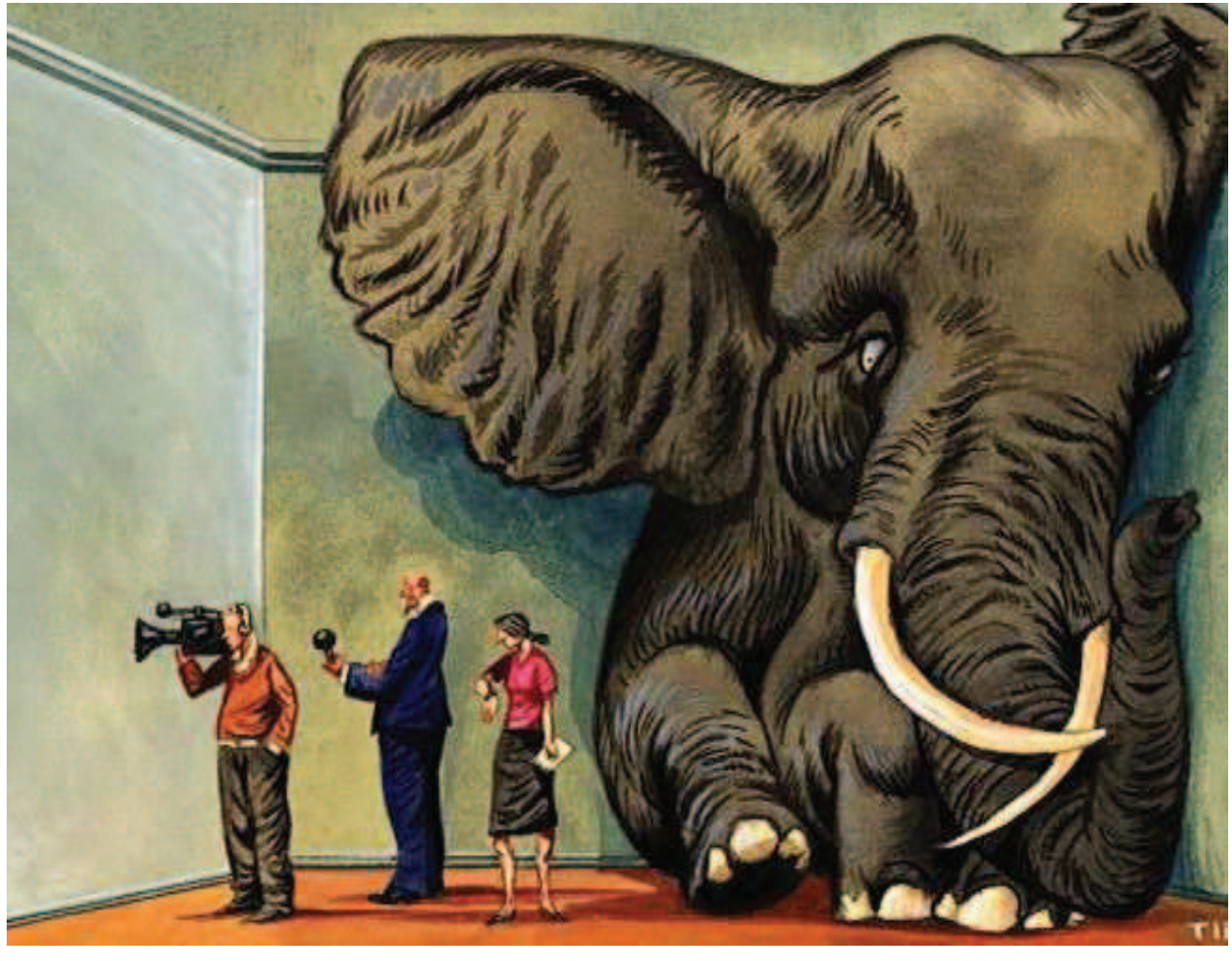
  - ⋆ The cached content of the $K-1$ files requested by others: used to cancel the interference caused by those same files.

Optimality of gain for a variety of settings

Theorem ([12, 13]): Caching gain $G = K\gamma$ proven to be optimal under uncoded cache placement.

- Uneven popularity distributions [2, 3, 4]
- Uneven topologies [5, 6]
- A variety of channels such as
  - ⋆ Erasure channels [7]
  - ⋆ MIMO broadcast channels with fading [8]
  - ⋆ Heterogeneous networks [9]
- D2D networks [10], etc.

The main crucial bottleneck of coded caching



Subpacketization bottleneck of coded caching

- In theory, caching gain $G = K\gamma$ increased indefinitely (as K increased)
- In practice the (realistic) gain remained hard-bounded by small constants

SUBPACKETIZATION PROBLEM
Coded caching algorithms required the splitting of finite-length files into an extremely large number of subpackets

- For MN algorithm, the gain $G = K\gamma$ required that each file be segmented at least into a total of

$$S_1 = \binom{K}{K\gamma} \tag{1}$$

subpackets.

Reason for large subpacketization

- Each file appears in each cache,
- Thus during delivery, a user must work together with all other users to get her file.
- In MN algorithm, must form cliques of $K\gamma + 1$ users, each requesting one subfile
  - ⋆ each user knows all subfiles requested from the clique
  - ⋆ except the one that she herself requests.
- There are a total of $\binom{K}{K\gamma}$ cliques in which a user must be part of
- All cliques must be used
- $\Rightarrow$ Must split each file into $\binom{K}{K\gamma}$ different subfiles.

Reduced effective caching gains

Communications under a maximum-allowable subpacketization S_{max}

- Can encode over a maximum number of users

$$\bar{K} = \arg \max_{K^o \leq K} \left\{ \binom{K^o}{K^o\gamma} \leq S_{max} \right\} \tag{2}$$

- thus substantially reduced *effective caching gain*

$$\bar{G}_1 = \bar{K}\gamma \ll K\gamma. \tag{3}$$

Effective caching gain

$$\bar{G}_1 = \bar{K}\gamma$$

Substantially reduced caching (and DoF) gains

- Since

$$\binom{\bar{K}}{\bar{K}\gamma} \in \left[\left(\frac{1}{\gamma}\right)^{\bar{K}\gamma}, \left(\frac{e}{\gamma}\right)^{\bar{K}\gamma}\right] = \left[\left(\frac{1}{\gamma}\right)^{\bar{G}_1}, \left(\frac{e}{\gamma}\right)^{\bar{G}_1}\right] \tag{4}$$

- then effective gain $\bar{G}_1$ is bounded as

$$\frac{\log S_{max}}{1+\log\frac{1}{\gamma}} \leq \bar{G}_1 \leq \frac{\log S_{max}}{\log\frac{1}{\gamma}}, \quad \bar{G}_1 \leq G \tag{5}$$

- Effective caching gain $\bar{G}_1$ and *effective sum-DoF* $\bar{d}_1 \triangleq 1+\bar{G}$
  - ⋆ under constant pressure from the generally small γ and small S_{max}

Why is there a big subpacketization problem

- Small cache memory (γ), thus need to encode over many users
  - ⋆ In wireless cellular settings, $\gamma < 10^{-2}$ (as argued in [14])
  - ⋆ Thus for target gain, we must encode over many users
  - ⋆ Thus subpacketization $\binom{\bar{K}}{\bar{K}\gamma}$ increases.

- Restrictions on maximum allowable subpacketization level S_{max}
  - ⋆ Modest file sizes (movies $\approx$ 1 Gigabyte)
  - ⋆ Video streaming, must break first into smaller 'sub-files' ($\approx 10^7$)
    - ∗ Reason: to avoid delay from asynchronous XOR decoding
  - ⋆ Atomic storage unit: 4096-byte sectors with zero-padding
  - ⋆ Minimum packet size
    - ∗ very small atomic packets cause communication delay overheads.

Diminished effective DoF from subpacketization

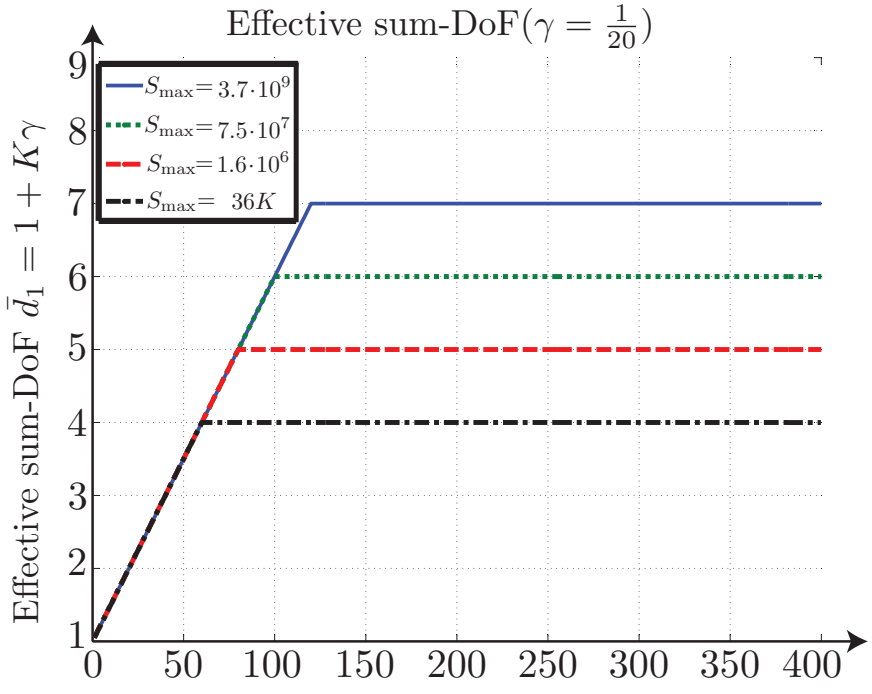


Figure 2: Maximum effective DoF $\bar{d}_1 = 1 + \bar{K}\gamma$ achieved by the original centralized algorithm (single antenna, $\gamma = 1/20$) in the presence of different subpacketization constraints S_{max}. The gain is hard-bounded irrespective of K (x-axis).

Diminished effective caching gains from subpacketization

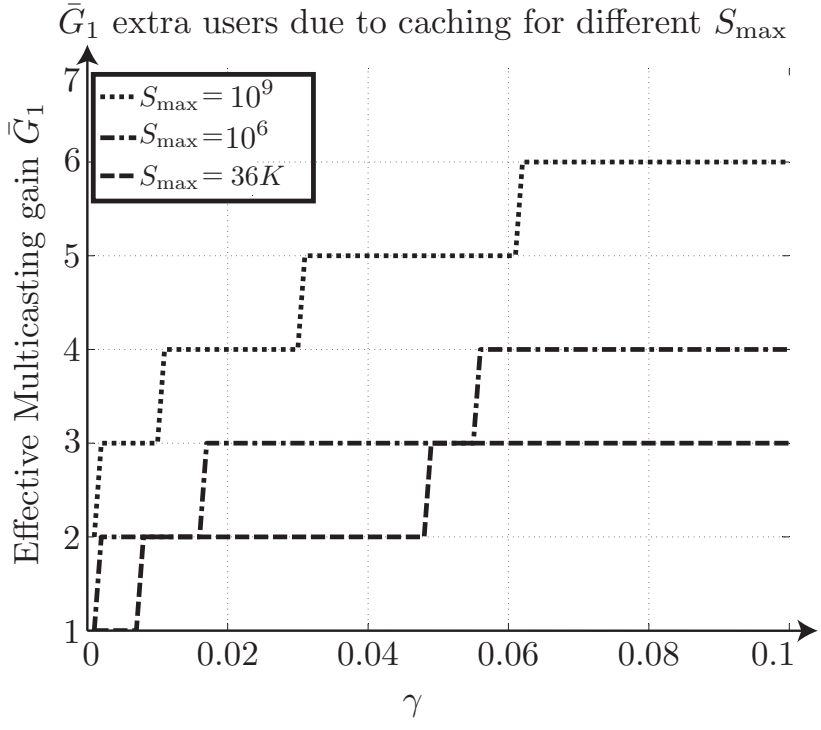


Figure 3: effective caching gain $\bar{G}_1 = \bar{d}_1 - 1$ (maximized over K) of the original algorithm for different S_{max}. Without subpacketization constraints, the theoretical gain is $G = K\gamma$ (unbounded as K increases).

Example - subpacketization problem

EXAMPLE 1:

- Subpacketization constraint $S_{max} \approx 10^6$
  - ⋆ 1 Gigabyte file size (movie, no streaming),
  - ⋆ 1 Kilobyte (KB) packet size

- Normalized cache size $\gamma < 1/20$
- Forced to encode over less than $\bar{K} = 80$ users

$$\binom{80}{4} > 10^6 = S_{max}$$

- Thus effective caching gain $\bar{G}_1 = \bar{K}\gamma < 4$

Example - subpacketization problem$_1$

EXAMPLE 2:

- Subpacketization constraint $S_{max} \approx 10^6$
- Normalized cache size $\gamma \geq 1/100$
- Forced to encode over less than $\bar{K} = 300$ users

$$\binom{300}{3} > 10^6$$

- Thus effective caching gain $\bar{G}_1 = \bar{K}\gamma < 3$

Example - subpacketization problem$_2$

EXAMPLE 3:

- Subpacketization constraint $S_{max} \approx 36K$ (low-latency video streaming applications)
- $\gamma \leq 1/20$
- Forced to encode over $\bar{K} \approx 60$ users

$$\binom{60}{3} \approx S_{max}$$

- Caching gain $\bar{G}_1 \approx 3$ ($\bar{d}_1 \approx 4$ users at a time)
- If $\gamma \leq 1/100$ then $\bar{G}_1 \approx 2$ ($\bar{d}_1 \approx 3$).

Coded caching algorithms with less subpacketization

- Subpacketization bottleneck sparked significant interest
- Effort to design new coded caching algorithms
  - ⋆ with reduced subpacketization costs

> The important aim:
>
> Increase effective gain $\bar{G}_1$ under subpacketization S_{max}

New coded caching algorithms for reducing subpacketization

Coded caching from placement delivery arrays

- *Placement-delivery array* codes (PD) ([18], see also [19])
  - ⋆ Reformulated coded caching into combinatorial design problem
  - ⋆ Exploited connections between coded caching and distributed storage
  - ⋆ Works for some operating parameters.
- Algorithm offers theoretical caching gain

$$G_{1,pd} = K\gamma - 1$$

  - ⋆ treating $d_{1,pd} = K\gamma$ users at a time (not $K\gamma + 1$)
- at reduced subpacketization

$$S_{1,pd} = \left(\frac{1}{\gamma}\right)^{K\gamma-1} = \left(\frac{1}{\gamma}\right)^{G_{1,pd}}$$

- Yields effective caching gain (under constraint S_{max})

$$\bar{G}_{1,pd} = \min\left\{\frac{\log S_{max}}{\log \frac{1}{\gamma}}, K\gamma - 1\right\}. \tag{6}$$

Coded caching from linear block codes

- Similar coded caching algorithm (LC) in [19]
- Used linear block codes over finite fields, to create set partitions
  - ⋆ partitions define how subpackets are cached and delivered.
- Allowed a tradeoff between an adjustable theoretical gain

$$G_{1,lc} \leq K\gamma - 1$$

  and the corresponding subpacketization $S \approx \left(\frac{1}{\gamma}\right)^{G_{1,lc}}$.

- Effective[1] gain

$$\bar{G}_{1,lc} \approx \frac{\log S_{max}}{\log \frac{1}{\gamma}}$$

[1] Recall for MN: $\frac{\log S_{max}}{1+\log \frac{1}{\gamma}} \leq \bar{G}_1 \leq \frac{\log S_{max}}{\log \frac{1}{\gamma}}, \quad \bar{G}_1 \leq G = K\gamma.$

Coded caching from hypergraph designs

- Hyper-graph codes from [20]

- *Theorem: There can be no caching algorithms that give a constant T (independent of K) with subpacketization that grows linearly with K*
  - ⋆ For γ independent of K
  - ⋆ Each file is divided into an identical number of subpackets
  - ⋆ Uncoded cache placement.

- Constructions nicely tradeoff performance with subpacketization
- Best design is Construction 6 (best known for several settings)
- Problem: Gain > 4 requires that that $K > 4/\gamma^2$ (approximately)
  - ⋆ K must be large because theoretical gain is small

$$G \approx K\gamma^2/4$$

  - ⋆ K must also be (essentially) a square integer (rare as $K \uparrow$)

Coded caching from Ruzsa-Szeméredi graphs

- Milestone of theoretical nature in recent work in [21].
- Employed the Ruzsa-Szeméredi graphs.
- Showing for first time that, under the assumption of (an unattainably) large K (trillions of trillions)

> *Theorem: One can get a (suboptimal) gain that scales with K, with subpacketization that scales with $K^{1+\delta}$ for some arbitrarily small positive δ.*

Subpacketization remained the crucial bottleneck

PROBLEM REMAINS:

- While new algorithms can exponentially reduce subpacketization
- Problem: very small improvement on the actual gain $\bar{G}$
  - ⋆ over the original (MN) algorithm in [1]
  - ⋆ for realistic values of γ and S_{max}.

For example, for $\gamma \leq 1/20$ and $S_{max} \leq 10^5$:
No known algorithm can improve over the MN algorithm effective gain (and effective DoF) by more than TWO EXTRA USERS

- ⋆ This best-known improvement is due to Construction 6 in [20]
  - ∗ $(a = b = 2, \lambda = 40)$ which encodes over $\bar{K} = 3160$ users
  - ∗ effective sum-DoF of 6, while the MN algorithm gives a DoF of 4 (with $\bar{K} = 60$). (2 additional users served at a time)

Coded caching with multiple antennas

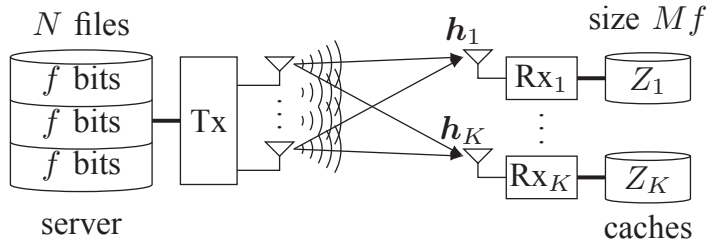


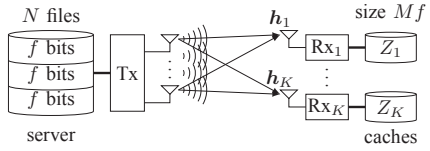
Figure 4: MISO BC with coded caching. Multiple transmit antennas.

- Motivated by coded caching limitations
- Many works considered coded caching with multiple antennas[2]

> Main aim of multiple antennas:
> Complement caching gains, with additional multiplexing gains

[2]E.g. [22, 23] and [24, 25, 26, 27, 28, 29, 30, 8, 31, 32] and others.

Multiserver (multi-antenna) coded caching



- K-user MISO BC with $L < K$ antennas ([22])

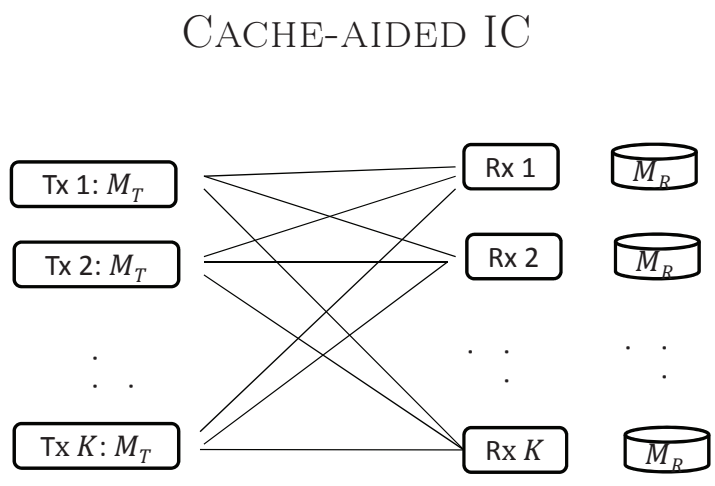
> *Theorem: Achievable theoretical sum-DoF of*
> $$d_L(\gamma) = L + K\gamma$$

  - ⋆ Multiplexing gain L (users served, per second per hertz)
  - ⋆ Extra theoretical caching gain $G = K\gamma$ (extra users, due to caching).

- Work nicely showed that:

> MULTIPLEXING AND CACHING GAINS CAN IN THEORY BE COMBINED ADDITIVELY

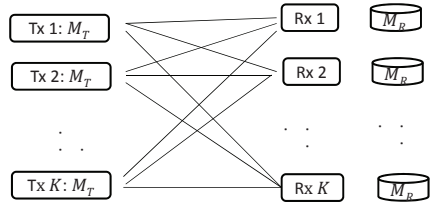
Multi-transmitter cache-aided interference scenario



- Similar conclusions in context of cache-aided interference scenario ([23])

- K_T transmitters, each with normalized cache size γ_T
  - ⋆ Each transmitter can store fraction γ_T of N-file library
  - ⋆ communicates to K receivers, each with normalized cache size γ.

Multi-transmitter cache-aided interference scenario



Theorem ([23]): Achievable theoretical sum-DoF of

$$d(\gamma) = K_T\gamma_T + K\gamma$$

At most a factor of 2 from the optimal (one-shot) linear-DoF.

- Work nicely showed that

Main conclusion:
Cooperative multiplexing gain $K_T\gamma_T$ (due to tx-cache redundancy) can be additively combined with theoretical caching gain $G = K\gamma$.

Severe subpacketization of multi-antenna coded caching

- Subpacketization in multiserver case ([22], MISO BC)

$$S = \binom{K}{K\gamma}\binom{K - K\gamma - 1}{L - 1} \tag{7}$$

- Subpacketization in multi-transmitter case ([23])

$$S = \binom{K}{K\gamma}\binom{K_T}{K_T\gamma_T} \tag{8}$$

- Adding extra transmitters
  - ⋆ Maintained the theoretical caching gains
  - ⋆ Added extra multiplexing gains

BUT kept high subpacketization, thus reducing effective caching gains.

Subpacketization bottleneck summary example

Under the generous assumptions that

$$S_{max} \leq 10^5,\ \gamma \leq 1/50,\ K \leq 10^5$$

Best known DoF boost due to caching is $\bar{G} = 5$ additional served users

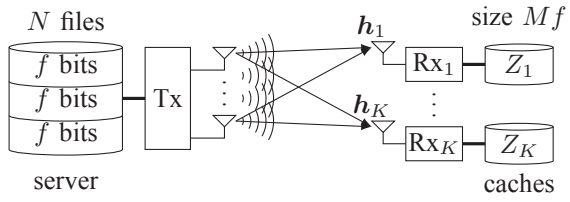
**** In any known single-antenna or multi-antenna setting****

3

[3]This corresponds to Hypergraph Construction 6 ($a = b = 2, \lambda = 100$), and it requires approximately 20000 users ([20]). Theoretical gain would have been $G = 400$ (reduced to $\bar{G} = 5$).

Summary of setting and parameters

System model

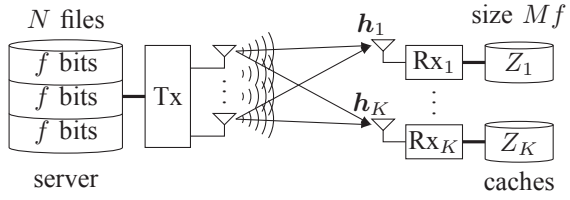


- K-user MISO BC
- L-antenna transmitter. Single-antenna receivers

- Tx library of N files $W_1, W_2, \ldots, W_N$
  - ⋆ File size $|W_n| = f$ bits
- User k cache Z_k. Size $|Z_k| = Mf$ bits (naturally $M \leq N$).

- Placement phase: pre-fill caches $Z_1, Z_2, \ldots, Z_K$
- Delivery phase
  - ⋆ Starts when each user k requests a file W_{R_k} from transmitter
  - ⋆ Transmitter aims to deliver the (remaining of the) requested files
  - ⋆ Aim to reduce (delivery phase) duration T.

Channel model



- During delivery phase, for each transmission, received signal at user k

$$y_k = \boldsymbol{h}_k^T \boldsymbol{x} + w_k, \quad k = 1, \ldots, K \tag{9}$$

  - ⋆ $\boldsymbol{x} \in \mathbb{C}^{L \times 1}$ is the transmitted vector across antennas
    - ∗ satisfying a power constraint $\mathbb{E}(||\boldsymbol{x}||^2) \leq P$
  - ⋆ $\boldsymbol{h}_k \in \mathbb{C}^{L \times 1}$ is (possibly random) channel from antennas to user k
  - ⋆ w_k is unit-power AWGN noise at receiver k
- We assume that P is high (high SNR)
- Assume perfect CSI throughout the (active) nodes as in [22, 23]
- Assume fading process that is statistically symmetric across users.

Understanding worst-case delay measure T

- As in [1], T is the number of time slots, per file served per user
  - ⋆ needed to complete the delivery process, *for any request*.
- The wireless link capabilities, and the time scale, are normalized
  - ⋆ 1 t-slot: duration to communicate a single file to one receiver
    - ∗ (had there been no caching and no interference.)
- As in the single-stream case in [1], T is the delay that allows
  - ⋆ in the information theoretic sense (large file sizes f)
  - ⋆ that each receiver k decodes (with probability 1) its message W_{R_k}.
- T is the maximum such delay, maximized over any request vector $\{W_{R_k}\}_{k=1}^{K}$.

Transition to cache-aided DoF

- High-SNR normalized delay T (cf. [31]; see also [24, 25])
  - ⋆ accounts for file sizes and high-SNR link capacity scaling log(SNR)
  - ⋆ identical to "Rate" used in [1] for single-stream error-free setting.
- Thus, for high SNR, we get cache-aided sum DoF

$$d_L(\gamma) = \frac{K(1-\gamma)}{T}$$

  - ⋆ Defined in [33] in the context of tx-caching
  - ⋆ Defined in [31] in the context of rx-caching (see also [24, 25]).

- The sum-DoF is the sum of multiplexing and theoretical caching gains
  - ⋆ describes the total amount of users served at a time.

Summary of Notation

- $d_1(\gamma) = 1 + K\gamma$: Theoretical DoF ($L = 1$)
- $d_L(\gamma) = L + K\gamma$: Theoretical DoF (multiple antennas)
- $d_L(\gamma = 0) = L$: Multiplexing gain
- G: Theoretical caching gain
  - $G = d_1(\gamma) - d_1(\gamma = 0) = d_L(\gamma) - d_L(\gamma = 0) = K\gamma$
  - G additional users served at a time, due to caching
- $S_1 = \binom{K}{K\gamma}$: Subpacketization needed for theoretical G ($L = 1$)
- S_{max}: Maximum allowable subpacketization
- S_L: Subpacketization needed for theoretical G (multiple antennas)
- $\bar{d}_1(\gamma)$: Effective (subpacketization-constrained) DoF ($L = 1$)
- $\bar{G}_1 = \bar{d}_1(\gamma) - 1$: Effective caching gain ($L = 1$)
- $\bar{d}_L(\gamma)$: Effective DoF (multiple antennas)
- $\bar{G}_L = \bar{d}_L(\gamma) - L$: Effective caching gain (multiple antennas)

Intuition on important terms

- $\bar{d}_L(\gamma = 0) = d_L(\gamma = 0) = L$ is the multiplexing gain.

- Effective caching gain $\bar{G}_L$ describes the actual number of additional users that can be served at a time
  - ⋆ As a result of introducing caching
  - ⋆ under a subpacketization constraint.

- The effective DoF $\bar{d}_L(\gamma) = L + \bar{G}_L$
  - ⋆ describes the actual (total) number of users that can be served at a time, under a subpacketization constraint.

Motivation/Justification for ADDITIVE caching gain

- We measure the caching gain as the DoF difference (not ratio)

$$G = d_1(\gamma) - d_1(\gamma = 0) = d_L(\gamma) - d_L(\gamma = 0) = K\gamma$$

- In theory, the two gains (multiplexing and caching) aggregate additively

$$d_L(\gamma) = L + K\gamma$$

$$d(\gamma) = K_T\gamma_T + K\gamma$$

Additive gain G better suited for multi-antenna settings because

- Cleanly removes multiplexing gain thus isolating effect of caching
- Caching gain that does not vanish with increasing L (unlike ratio)
- Caching gain that scales with cumulative cache size at rxs (as $K \uparrow$).

Back to subpacketization

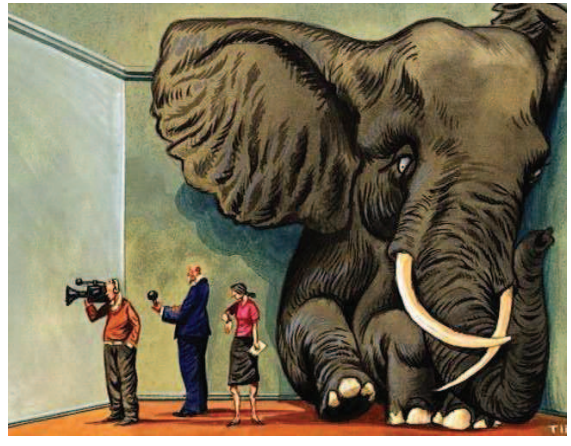
Reminder of Subpacketization bottleneck

Under the generous assumptions that

$$S_{max} \leq 10^5, \ \gamma \leq 1/50, \ K \leq 10^5$$

Best known DoF boost due to caching is $\bar{G} = 5$ additional served users

**** In any known single-antenna or multi-antenna setting****



Description of scheme
Via example

Description of the scheme: general steps

Basic steps of algorithm

1. User grouping

2. Subpacketization

3. Cache placement

4. Transmission

5. Decoding: 'Caching-out' out-of-group messages

6. Decoding: Nulling-out intra-group messages - completion of decoding.

Description of scheme via an example

- Design scheme for $K = 50$, $L = 5$ and $\gamma = M/N = 3/10$
  - ⋆ Theoretical caching gain $G = K\gamma = 15$
  - ⋆ Multiplexing gain $L = 5$
  - ⋆ Desired (maximum known) DoF

$$d_{\Sigma} = L + G = L + K\gamma = 5 + 15 = 20$$

- Previously needed subpacketization
  - ⋆ MN ($d = 1 + K\gamma$) would need

$$S = \binom{K}{K\gamma} \approx 2 \cdot 10^{12}$$

  - ⋆ Multi-server ($d = L + K\gamma$) would need

$$S = \binom{K}{K\gamma}\binom{K - K\gamma - 1}{L - 1} \approx 10^{17}.$$

Description of scheme: $K = 50$, $L = 5$ and $\gamma = M/N = 3/10$

- We will achieve desired sum-DoF of

$$d_{\Sigma} = L + K\gamma = 20$$

- With subpacketization

$$S_L = 120.$$

Description of scheme via example
$K = 50, L = 5, \gamma = 3/10$

- Step 1: User grouping
  - ⋆ Split the $K = 50$ users into $K' = K/L = 10$ groups of $L = 5$:

$$\mathcal{G}_1 = \{1, 11, 21, 31, 41\}, \ldots, \mathcal{G}_{10} = \{10, 20, 30, 40, 50\}.$$

$$\downarrow \text{Proceed as if } L = 1,\ K' = 10 \text{ users} \downarrow$$

- Step 2: Subpacketization
  - ⋆ Recall that $K'\gamma = 3$
  - ⋆ Split each file W_n into $|\mathcal{T}| = \binom{K'}{K'\gamma} = 120$ parts

$$W_n = \{W_n^{(1,2,3)}, W_n^{(1,2,4)}, \ldots, W_n^{(1,3,4)}, \ldots, W_n^{(8,9,10)}\}$$

Description of scheme via example
$K = 50, L = 5, \gamma = 3/10$

- STEP 3: CACHE PLACEMENT (as if user = group, K'-user, $L = 1$)

$$Z_{\mathcal{G}_1} = \{W_n^{(1,2,3)}, W_n^{(1,2,4)}, \dots W_n^{(1,3,4)}, \dots W_n^{(1,9,10)}\}_{n=1}^{N}$$
$$\vdots$$
$$Z_{\mathcal{G}_{10}} = \{W_n^{(1,2,10)}, W_n^{(1,3,10)}, \dots, W_n^{(2,3,10)}, \dots W_n^{(8,9,10)}\}_{n=1}^{N}$$

- STEP 4: TRANSMISSION
  - ⋆ We will serve $K'\gamma + 1 = 4$ groups at a time.
  - ⋆ First treat the group clique $\chi = (1, 2, 3, 4)$.
  - ⋆ Gather $L = 5$ subfiles for the 5 users in group 1

$$\boldsymbol{w}_1^{(2,3,4)} = [W_{R_1}^{(2,3,4)}, W_{R_{11}}^{(2,3,4)}, W_{R_{21}}^{(2,3,4)}, W_{R_{31}}^{(2,3,4)}, W_{R_{41}}^{(2,3,4)}]^T$$

  - ⋆ Similarly gather $\boldsymbol{w}_2^{(1,3,4)}, \boldsymbol{w}_3^{(1,2,4)}, \boldsymbol{w}_4^{(1,2,3)}$ for the other groups.

Scheme: $K = 50, L = 5, \gamma = 3/10$

- Then simply transmit

  $$\boldsymbol{x}_{(1,2,3,4)} = (\mathbf{H}^{\mathcal{G}_1})^{-1}\boldsymbol{w}_1^{(2,3,4)} + (\mathbf{H}^{\mathcal{G}_2})^{-1}\boldsymbol{w}_2^{(1,3,4)} + (\mathbf{H}^{\mathcal{G}_3})^{-1}\boldsymbol{w}_3^{(1,2,4)} + (\mathbf{H}^{\mathcal{G}_4})^{-1}\boldsymbol{w}_4^{(1,2,3)}$$

  - $(\mathbf{H}^{\mathcal{G}_g})^{-1}$ is the (normalized) inverse of the $L \times L$ channel to group $\mathcal{G}_g$.

- Steps 5-6: Decoding
  - caching out out-of-group interfering files
    - Receiver 1 can remove – using its cache – the last three summands
  - nulling-out intra-group interfering files
    - ZF can remove the unwanted $L - 1 = 4$ elements from $\boldsymbol{w}_1^{(2,3,4)}$.

- Performance
  - Serving 4 groups at a time. No data repeated.
  - Serving $4 \times 5 = d_L(\gamma) = 20$ users at a time
  - Caching gain is $G = 15$
  - Subpacketization is $S_L = \binom{K'}{K'\gamma} = \binom{K/L}{K\gamma/L} = \binom{10}{3} = 120$.

Intuition: Scheme $K = 50, L = 5, \gamma = 3/10$

$$L = 5, K = 50, K\gamma = 15 \qquad \text{vs.} \qquad L = 1, K' = 10, K'\gamma = 3$$

$$\begin{array}{lllll}
\boldsymbol{x}_{(1,2,3,4)} & = (\mathbf{H}^{\mathcal{G}_1})^{-1}\boldsymbol{w}_1^{(2,3,4)} & +(\mathbf{H}^{\mathcal{G}_2})^{-1}\boldsymbol{w}_2^{(1,3,4)} & +(\mathbf{H}^{\mathcal{G}_3})^{-1}\boldsymbol{w}_3^{(1,2,4)} & +(\mathbf{H}^{\mathcal{G}_4})^{-1}\boldsymbol{w}_4^{(1,2,3)} \\
\updownarrow & = \updownarrow & \updownarrow & \updownarrow & \updownarrow \\
\mathrm{XOR}_{(1,2,3,4)} & = W_1^{(2,3,4)} & \oplus W_2^{(1,3,4)} & \oplus W_3^{(1,2,4)} & \oplus W_4^{(1,2,3)}.
\end{array}$$

Theorem 1 *In the cache-aided MISO BC with L transmitting antennas and K receiving users, the delay of $T = \frac{K(1-\gamma)}{L+K\gamma}$ and the corresponding sum-DoF*[4]

$$d_L(\gamma) = L + K\gamma$$

can be achieved with subpacketization

$$S_L = \binom{K/L}{K\gamma/L}.$$

Main results - Similarly for Cache aided IC

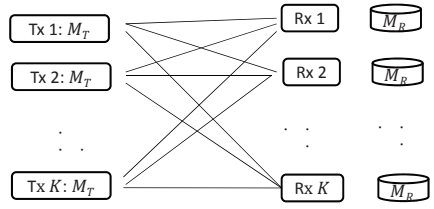
Theorem 2 *In the cache-aided IC with K_T transmitters and K receivers, with respective normalized cache sizes γ_T, γ, the sum-DoF*

$$d(\gamma) = K_T\gamma_T + K\gamma$$

can be achieved with subpacketization

$$S = \binom{K/(K_T\gamma_T)}{K\gamma/(K_T\gamma_T)}.$$

Adapting to the cache-aided interference scenario



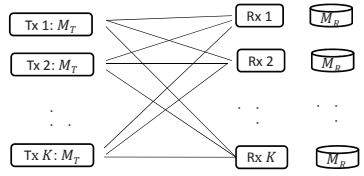
- TX-CACHE PLACEMENT: Consecutively cache whole files
  - ⋆ 1st tx caches the first M files, 2nd tx the next M files, and so on, modulo N.

$$Z_{\mathrm{Tx}_m} = \left\{ W_{1+(n-1)\mathrm{mod}N} : n \in \{1+(m-1)M, ..., Mm\} \right\}$$

- Each subfile in $K_T\gamma_T$ transmitters

- COOPERATIVE TRANSMISSION: Consider any given subfile
  - ⋆ $K_T\gamma_T$ transmitters will have this file
  - ⋆ Will play previous role of $L = K_T\gamma_T$ (use CSIT)
  - ⋆ Precoding subfile with same precoders as before
  - ⋆ Split of $L = K_T\gamma_T$ streams within group $\mathcal{G}_g$ of $L = K_T\gamma_T$ rxs.

Adapting to the cache-aided interference scenario$_1$

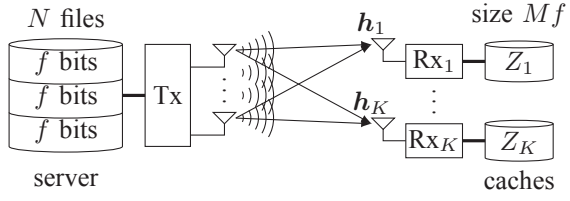


- PERFORMANCE: Treat $K'\gamma + 1$ groups $\Rightarrow K_T\gamma_T + K\gamma \leq K$ users at a time.
  - ⋆ Sum-DoF of $K_T L_T \gamma_T + K\gamma$
  - ⋆ Subpacketization $S_{K_T\gamma_T} = \binom{\frac{K}{K_T\gamma_T}}{\frac{K\gamma}{K_T\gamma_T}}$.

MULTI-ANTENNA SETTING

- K_T BS, with L_T tx-antennas per BS
- Same placement: each subfile 'seen by' $L = K_T L_T \gamma_T$ antennas
- Sum-DoF of $K_T L_T \gamma_T + K\gamma$
- Subpacketization $S_{K_T L_T \gamma_T} = \binom{\frac{K}{K_T L_T\gamma_T}}{\frac{K\gamma}{K_T L_T\gamma_T}}$.

Interpretation, examples and ramifications



Multiplicative boost of effective DoF

- Previously adding an antenna had a more modest, additive, impact
- Previously: without subpacketization constraints, extra tx antennas

$$d_1 = 1 + K\gamma \quad \longrightarrow \quad d_L = L + K\gamma$$

⋆ leaving the theoretical caching gain unaffected

⋆ adding $d_L(\gamma) - d_1(\gamma) = L - 1$ DoF.

For example: Adding one antenna, adds one DoF

But with subpacketization, we will see much more powerful impact of multiple antennas

Multiplicative boost of effective DoF_1

Corollary 1 *Under a maximum allowable subpacketization S_{max}, the multi-antenna effective caching gain and DoF take the form*

$$\bar{G}_L = \min\{L \cdot \bar{G}_1, G = K\gamma\} \tag{10}$$

$$\bar{d}_L = \min\{L \cdot \bar{d}_1, d_L = L + K\gamma\} \tag{11}$$

which means that with extra antennas, the DoF:
- is either increased by a multiplicative factor of L
- or it reaches the theoretical (unconstrained) DoF $d_L = L + K\gamma$.

Combining gains – ∞ file size

Combining gains – ∞ file size$_1$

The law of Tx-Addition

$1 + 28 = 29$

$2 + 28 = 30$

$3 + 28 = 31$

$4 + 28 = 32$

...

...

Effective combining of gains – Constrained

Effective combining of gains – Constrained$_1$

The effective law of Tx-Addition

$$1 + 7 = 8$$

$$2 + 7 = 16$$

$$3 + 7 = 24$$

$$4 + 7 = 32$$

$$\dots$$

$$\dots$$

Understanding multiplicative DoF boost

- Recall that for $L = 1$, then $S_1 = \binom{K}{K\gamma}$, thus we can encode over

$$\bar{K}_1 \triangleq \arg \max_{K^o \leq K} \left\{ \binom{K^o}{K^o\gamma} \leq S_{max} \right\} \text{ users}$$

- But if L antennas, then $S_L = \binom{\frac{K}{L}}{\frac{K\gamma}{L}}$, thus we can encode over

$$\bar{K}_L \triangleq \arg \max_{K^o \leq K} \left\{ \binom{\frac{K^o}{L}}{\frac{K^o\gamma}{L}} \leq S_{max} \right\} = \min\{L \cdot \bar{K}_1, K\} \qquad (12)$$

- Up to L times more (encoded) users, thus up to L times more caching gain

$$\bar{G}_L = \bar{K}_L\gamma = \min\{L \cdot \bar{G}_1, G\}$$

  - ⋆ up to the theoretical $G = K\gamma$.

Understanding multiplicative DoF boost$_1$

- Either achieve theoretical (unconstrained) gain $\bar{G}_L = G$
  - ⋆ When $\binom{\frac{K}{L}}{\frac{K\gamma}{L}} \leq S_{max}$
  - ⋆ Corresponding to multiplicative boost $\frac{\bar{G}_L}{\bar{G}_1} = \frac{G}{\bar{G}_1}$
- Or the effective gain and the effective sum-DoF both experience a multiplicative increase by a factor of exactly L.

THE L-FOLD MULTIPLICATIVE DOF BOOST STAYS INTO EFFECT AS LONG AS $\binom{\frac{K}{L}}{\frac{K\gamma}{L}} \geq S_{max}$.

IN ESSENCE: L-FOLD DOF BOOST STAYS INTO EFFECT AS LONG AS SUBPACKETIZATION REMAINS AN ISSUE.

Example: Multiplicative boost of effective DoF

EXAMPLE:

- L-antenna MISO BC, $\gamma = 1/20$ and $K = 1280$
- THEORETICAL caching gain of $G = K\gamma = 64$
  - ⋆ theoretical sum-DoF of $d_L = L + G = L + 64$.
  - ⋆ E.g. when $L = 1$ then $d_1 = 65$, when $L = 2$ then the sum DoF is 66, and so on.
- Consider a subpacketization limit $S_{max} = \binom{80}{4} \approx 1.5 \cdot 10^6$

Example: Multiplicative boost of effective DoF$_1$

- For $L = 1$, we could encode over only $\bar{K} = 80$ ($\binom{80}{4} \approx S_{max}$)
  - ⋆ effective caching gain $\bar{G}_1 = \bar{K}\gamma = 4$
  - ⋆ treating a total of $\bar{d}_1 = 1 + \bar{G}_1 = 5$ users at a time.

- $L = 2$ tx-antennas: can encode over $\bar{K}_L = L \cdot \bar{K}_1 = 2 \cdot 80 = 160$ users
  - ⋆ Since $\binom{\frac{\bar{K}_L}{L}}{\frac{\bar{K}_L \gamma}{L}} = \binom{80}{4} \leq S_{max}$
  - ⋆ Caching gain of $\bar{G}_L = \bar{K}_L \gamma = 160\frac{1}{20} = 8$
  - ⋆ treating a total of $\bar{d}_L = L + \bar{G}_L = L(1 + \bar{G}_1) = 10$ users at a time
  - ⋆ doubling the number of users served at a time, from 5 to 10.
- For $L = 4$, then encode over $\bar{K}_L = 320$ users
  - ⋆ Caching gain $\bar{G}_L = 16$ and DoF $\bar{d}_L = 20$
- For $L = 16$ antennas, encode over $\bar{K}_L = 16 \cdot \bar{K}_1 = 1280$ users
  - ⋆ reach theoretical optimal $\bar{G}_L = 64$, $\bar{d}_L = 80$
  - ⋆ 16-fold multiplicative DoF boost.

Intuition on effective gain $\bar{G}_L$

Corollary 2 *Given a maximum allowable subpacketization S_{max}, the effective caching gain of the presented scheme is bounded as*

$$\bar{G}_L \geq \min\{\ L \cdot \frac{\log S_{max}}{1+\log(\frac{1}{\gamma})}, K\gamma\ \}. \tag{13}$$

5

[5] *Proof: Sterling's approximation:* $S_L = \binom{K'}{K'\gamma} \leq \left(\frac{e}{\gamma}\right)^{K'\gamma} = \left(\frac{e}{\gamma}\right)^{\frac{G}{L}} \leq S_{max}$. *Thus* $\bar{G}_L \geq L \cdot \frac{\log S_{max}}{1+\log(\frac{1}{\gamma})}$ *(up to the theoretical gain* $G = K\gamma$*).*

Multiplicative boost of caching gains

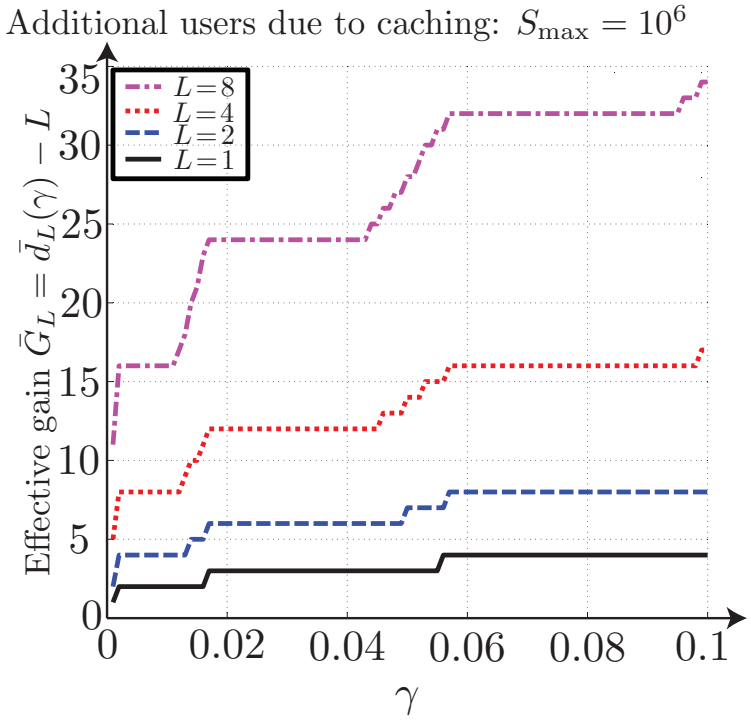


Figure 5: Maximum achievable effective caching gain $\bar{G}_L = d_L(\gamma) - L$ (maximized over all possible K), achieved by the new scheme for different L, under subpacketization constraint $S_{max} = 10^6$.

Multiplicative boost of caching gains[1]

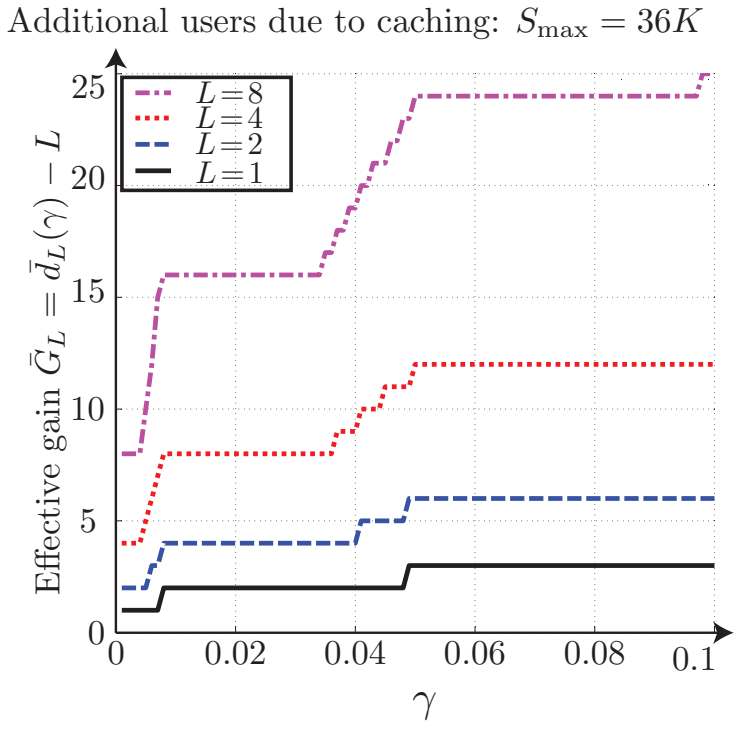


Figure 6: Maximum achievable effective caching gain $\bar{G}_L = d_L(\gamma) - L$ (maximized over all possible K), achieved by the new scheme for different L, under subpacketization constraint $S_{max} = 3.6 \cdot 10^4$.

Practical implications

Practical implication - Making small caches relevant

- Exponential increase in range of cache sizes that achieve a target gain.
- High gains with small γ needs encoding over many users
- But increased subpacketization stops us from encoding over many users
  - ⋆ When $L = 1$, then $S_1 \geq \left(\frac{1}{\gamma}\right)^G$, thus for target caching gain G, we would have needed

$$\gamma \geq (S_{\max})^{-1/G}. \tag{14}$$

Practical implication - Making small caches relevant$_1$

- But when $L > 1$ then the reduced $S_L \geq \left(\frac{1}{\gamma}\right)^{\frac{1}{L}G}$ can allow for the same caching gain G (given sufficiently many users to encode over) with only

$$\gamma \geq \left((S_{\max})^{-1/G}\right)^{L}. \tag{15}$$

Exponential reduction in the minimum applicable γ

Matches spirit of caches on the edge (network periphery)

- Relatively small caches
- Abundantly many caches.

Corollary 3 *In our L-antenna MISO BC setting, a subpacketization of*

$$S = \binom{1/\lambda}{x-1} = \binom{\frac{1}{\lambda}}{\frac{\gamma}{\lambda}}$$

can yield a DoF that is x times the multiplexing gain $L = \lambda K$.

Subpacketization cost determined by ratio $x = \frac{d_L(\gamma)}{d_L(\gamma=0)}$

- not by K or $L = \lambda K$, and not by desired caching gain G

- Same[6] S for treating $K'\gamma + 1$ users at a time in the single-antenna case, now guarantees treatment of $K'\gamma + 1$ groups $(K' = K/L)$.

[6] $d_L(\gamma) = L + K\gamma = x \cdot L$ implies that $K\gamma = L(x-1)$ and that $\gamma = \lambda(x-1)$. Thus $S_L = \binom{K/L}{K\gamma/L} = \binom{\frac{1}{\lambda}}{\frac{\gamma}{\lambda}} = \binom{1/\lambda}{x-1}$.

Example: cost of complementing multiplexing gains

- Can double DoF of MIMO system with $\gamma = \lambda(x-1) = \lambda$ and

$$S_L = 1/\lambda = K/L$$

- Can triple DoF of MIMO system ($L \to 3L$) with $\gamma = 2\lambda$ and

$$S_L = \binom{1/\lambda}{2} < \frac{1}{2\lambda^2}$$

- EXAMPLE: Assume multi-antenna BS gives $\lambda = 1/30$ DoF per user
- Can triple this with

$$\gamma = \frac{xL - L}{K} = \lambda(x-1) = 2\lambda = 2/30$$

$$S_{max} = \binom{1/\lambda}{x-1} = \binom{30}{2} = 435$$

Corollary 4 *In asymptotic terms, as long as L scales with the caching gain $K\gamma$, the entire sum-DoF $L + K\gamma$ is achievable with constant subpacketization.*

7

Corollary 5 *For $L = K\gamma$: can achieve DoF $L + K\gamma$ with subpacketization*

$$S_L = \frac{1}{\gamma} = \frac{K}{L}.$$

8

[7] For $L = \frac{1}{q}K\gamma$ for some fixed $q \in \mathbb{Z}^+$, then $S = \binom{1/\lambda}{q}$ (indep. of K, L).

[8] Recall in multi-server, this would have been approximately $(\binom{K}{K\gamma})^2$.

Example: utility of matching $K\gamma$ with L

- Consider BC with $\gamma = 1/100$ and $L = 1$
  - ⋆ Caching gains $G = K\gamma = 10$ would require

$$S_1 = \binom{1000}{10} > 10^{23}$$

  - ⋆ In practice coded caching could not offer such gains.

- If though $L = 10$ antennas:
  - ⋆ Same caching gain $G = 10$ comes[9] with only

$$S_L = K/L = 100.$$

[9] Multiserver: $S \approx 10^{40}$.

Tx-Cooperation for boosting coded caching

All apply to cache-aided IC (BENEFITS OF TX-CACHE REDUNDANCY)

1. As tx-cache redundancy $K_T\gamma_T$ increases
   - the effective DoF will boost by a multiplicative factor of $K_T\gamma_T$
   - or it will reach the theoretical (unconstrained) DoF $K_T\gamma_T + K\gamma$

2. With tx-redundancy $K_T\gamma_T$, then effective caching gain is bigger than $(K_T\gamma_T) \cdot \frac{\log S_{max}}{1+\log(\frac{1}{\gamma})}$

3. Increasing tx redundancy $K_T\gamma_T$, allows for $\gamma \geq \left((S_{\max})^{-1/G}\right)^{K_T\gamma_T}$ to offer a (receiver-side) caching gain of $G = K\gamma$.

Tx-Cooperation for boosting coded caching$_1$

4. Subpacketization $S = \binom{\frac{K}{K_T\gamma_T}}{x-1}$ can yield a sum DoF that is x times the cooperative multiplexing gain $K_T\gamma_T$.

5. In asymptotic terms, if tx-cache redundancy $K_T\gamma_T$ scales with rx-cache redundancy $K\gamma$, the entire sum-DoF $K_T\gamma_T + K\gamma$ is achievable with constant subpacketization.

6. When tx-cache and rx-cache redundancies match ($K_T\gamma_T = K\gamma$),
   - DoF $K_T\gamma_T + K\gamma$ is achieved with $S_{K_T\gamma_T} = \frac{K}{K_T\gamma_T}$.

Example: Base-station cooperation

- $K = 10000$ cell-phone users (urban setting)
- $N = 10000$ low-definition Netflix movies (1 Gigabyte per movie)
- $M = 20$ (20 Gigabyte memory per phone)
  - ⋆ $\gamma = M/N = 1/500$
  - ⋆ Theoretical caching gain $G = 20$.
- Base-station can store entire library (10 Terabytes)

- If $K_T = 1, L_T = 1$ (one single-antenna BS), caching gain $G = 20$ needs (MN) $S_1 = \binom{K}{K\gamma} = \binom{10000}{20} > 10^{61}$.
- If $K_T = 2$ BS with $L_T = 5$ antennas each, then gain needs $S_L = \binom{\frac{K}{K_T L_T}}{\frac{K\gamma}{K_T L_T}} = \binom{10000/10}{20/10} = \binom{1000}{2} \approx 5 \cdot 10^5$
  - ⋆ adding caching triples total number of users
- with $K_T = 4$ ($L_T = 5$) cooperating BS, gain can be achieved with $S = \binom{10000/20}{20/20} = 500$.

...Base-station cooperation: Example continues

- If library is pruned to $N = 1000$ most popular movies[10]
  - $\gamma = 1/50$
  - theoretical caching gain of $G = K\gamma = 200$ (additional users served per second per hertz).
- With one large-MIMO array with $L_T = 100$ antennas
  - DoF $d_L(\gamma) = 300$
  - caching would allow us to serve 200 additional users at a time
  - with $S_L = \binom{10000/100}{200/100} = \binom{100}{2} \approx 5000$.
- Same with $K_T = 5$ cooperating BS with $L_T = 20$ antennas each.

[10] No consideration of cost of cache-misses

Near-optimality of schemes

Corollary 6 *The described subpacketization* $S_L = \binom{\frac{K}{L}}{\frac{K\gamma}{L}}$ *and* $S_{K_T\gamma_T} = \binom{\frac{K}{K_T\gamma_T}}{\frac{K\gamma}{K_T\gamma_T}}$ *guarantees sum-DoF performance that is at most a factor of 2 from the theoretical optimal linear-DoF*

11

[11]Proof: Schemes have the 'one-shot, linear' property, thus are amenable to the analysis in [23].

Elevating other CC algorithms to the L antenna BC

- Use other coded caching algorithms to further reduce subpacketization
- Saw 'elevated' MN algorithm
  - ⋆ from $L = 1$ scenario with $K' = K/L$ users
  - ⋆ to the L-antenna case with K' groups of L-users per group.
- Same idea holds to other centralized CC algorithms (e.g. [18, 19, 21])

Steps for elevating other CC algorithms

1. Choose new CC algorithm for the single-stream K'-user scenario.
2. Split the K users into K' groups of L users each
   - Employ new algorithm to fill caches as in K'-user ($L = 1$) case
     - ⋆ As if each group is a user

SUCH THAT SAME-GROUP USERS HAVE IDENTICAL CACHES

3. Using new CC algorithm for the single-stream K'-user scenario
   - Generate the sequence of XORs as if $L = 1$ and K' users.
   - Each XOR consists of $d_1'(\gamma)$ summands
     - ⋆ $d_1'(\gamma)$ is sum-DoF of CC algorithm in the K'-user $L = 1$ case
   - Each element (summand) of the XOR, corresponds to a group of users.

Steps for elevating other CC algorithms$_1$

4. Each such XOR summand is replaced by a (precoded) L-length vector.

   Each such vector carries the L-requests of the associated group.

5. Add these $d_1^{'}$ vectors together (corresponding to XOR), to form composite tx vector.
   - Each composite vector treats a total of $d_1^{'}$ groups at a time, i.e., treats $L \cdot d_1^{'}(\gamma)$ users at a time.
6. Then continue with the rest of the XORs.

Example: Elevating PDA/LC algorithms

- Recall elevating the MN algorithm
  - ⋆ MN treats $d_1^{'}(\gamma) = K'\gamma + 1$ users at a time (K'-user case with $L = 1$)
  - ⋆ Thus elevated-MN treated $d_1^{'} = K'\gamma + 1$ groups at a time
  - ⋆ Thus treated a total of $d_L(\gamma) = L \cdot d_1^{'}(\gamma) = L + K\gamma$ users at a time

- Elevating the PDA and LC algorithms in [18, 19]
  - ⋆ Must first change cache-placement and XOR-generation
  - ⋆ For $L = 1$, PDA algor. treats $d_{1,pd}^{'} = K'\gamma$ users/time (not $K'\gamma + 1$)
  - ⋆ Thus for $L \geq 1$, it treats $d_{1,pd}^{'} = \frac{K}{L}\gamma$ groups at a time ($L \leq K\gamma$)
  - ⋆ Thus treats $d_{L,pd}(\gamma) = L \cdot d_{1,pd}^{'} = K\gamma$ users at a time (not $K\gamma + L$).

Elevating PDA/LC coded caching algorithms

Corollary 7 *Given a maximum allowable subpacketization S_{max}, the effective caching gain of the elevated PD and LC algorithms, takes the form*

$$\bar{G}_{L,pd} = \bar{G}_{L,lc} = \min\{\ L \cdot \frac{\log S_{max}}{\log(\frac{1}{\gamma})}\ , K\gamma - L\}. \tag{16}$$

PROOF:

- $d'_{1,pd} = K'\gamma \Rightarrow d_{L,pd} = K\gamma$
- Theoretical gain $G_{L,pd} = d_{L,pd}(\gamma) - d_{L,pd}(\gamma = 0) = K\gamma - L$
- Subpacketization $S_{L,pd} = \left(\frac{1}{\gamma}\right)^{K'\gamma - 1} = \left(\frac{1}{\gamma}\right)^{\frac{G_{L,pd}}{L}}$
- Thus effective gain $\bar{G}_{L,pd} = L \cdot \frac{\log S_{max}}{\log(\frac{1}{\gamma})}$
  - ⋆ naturally bounded by theoretical caching gain $K\gamma - L$.

L-fold increase in impact of alternate CC algorithms

- Consider any two coded caching algorithms in the single stream case
  - ⋆ each algorithm provides its own effective caching gain
- Elevate each to the L-antenna case
- Any difference in their effective gains is magnified up to L times
  - ⋆ because the underlying algorithms are used at level of groups of users.

Example: Elevated-MN vs. elevated PD and LC

$$\bar{G}_{L,pd} = \min\{L \cdot \frac{\log S_{max}}{\log(\frac{1}{\gamma})}, K\gamma - L\}$$

$$\bar{G}_L \geq \min\{L \cdot \frac{\log S_{max}}{1 + \log(\frac{1}{\gamma})},\ K\gamma\}$$

- Thus improvement in effective gains is bounded as

$$\bar{G}_{L,pd} - \bar{G}_L \leq L \cdot \frac{\log S_{max}}{(\log(\frac{1}{\gamma}))(1 + \log(\frac{1}{\gamma}))}.$$

Example: Elevated-MN vs. elevated PD and LC_1

- When $L = 1$, this improvement - for realistic γ, S_{max} - can be small

$$\bar{G}_{1,pd} - \bar{G}_1 \leq \frac{\log S_{max}}{(\log(\frac{1}{\gamma}))(1 + \log(\frac{1}{\gamma}))}$$

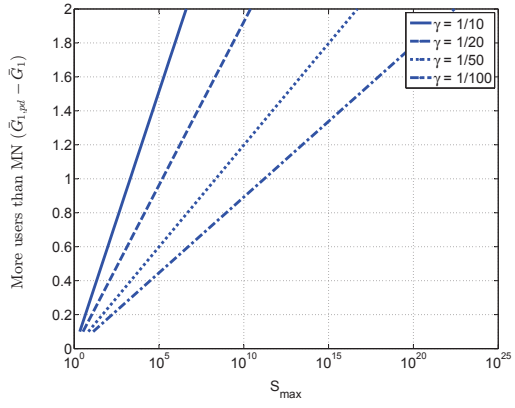


Figure 7: An upper bound on DoF improvement over MN ($L = 1$) by PDA. E.g. for $\gamma \leq 1/20$ and $S_{max} \leq 10^5$, the improvement is less than 1 extra user (served at a time).

Example: Elevated-MN vs. elevated PD and LC_2

$$\bar{G}_{L,pd} - \bar{G}_L \leq L \cdot \frac{\log S_{max}}{(\log(\frac{1}{\gamma}))(1 + \log(\frac{1}{\gamma}))}.$$

- When a new algorithm is elevated, its improvement increases as a multiple of L

Conclusion to draw

– Our method does not bypass the need for novel single-stream coded caching algorithms of reduced subpacketization.

– It in fact accentuates the importance of searching for such algorithms.

Some conclusions

Conclusions

- Presented simple scheme which exploits transmitter-side dimensionality
  - ⋆ Provides very substantial reductions in the required subpacketization
  - ⋆ Without any sacrifice on the caching gain.
- In theory, adding antennas gives *additive* DoF increase

$$d_1(\gamma) = 1 + G \longrightarrow d_L(\gamma) = L + G$$

  - ⋆ allowing the addition of $L - 1$ extra users served at a time

- In practice, adding antennas gives *multiplicative* effective-DoF increase

$$\bar{d}_1 = 1 + \bar{G}_1 \longrightarrow L + L \cdot \bar{G}_1$$

MAIN IMPACT OF MULTIPLE TX IS NOT THE MULTIPLEXING GAIN, BUT RATHER THE BOOST ON THE EFFECT OF RECEIVER-SIDE CODED CACHING

Intuition on design

- Design based on simple observation that multi-node (transmitter-side) precoding, reduces the need for content overlap.

> Usual source of problem: Subpacketization is high because we need many pairings between the different caches.

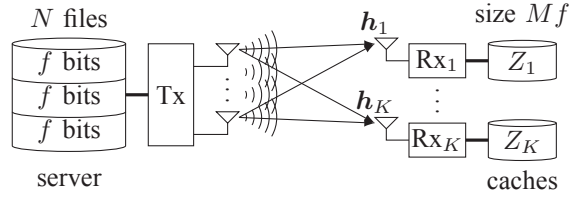
> Part of solution: Receivers of each group have identical caches.

  - ⋆ The number of different distinct caches is reduced
  - ⋆ Thus the number of pairings remains smaller.

- Subpacketization reduction from $\binom{K}{K\gamma}$ to $\binom{K/L}{K\gamma/L}$

Joint exposition of coded caching and PHY

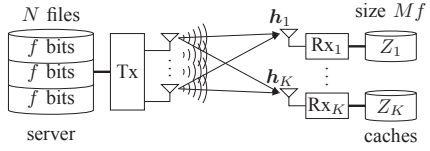
- Separable schemes separate PHY from coded caching
  - ⋆ Work obliviously of "network structure"
  - ⋆ Such schemes enjoy robustness against network-structure uncertainty
  - ⋆ Bounded gap to 'joint' (non-separable) optimal (cf. [34]).



- Our scheme is not separable (cache-placement needs to know L)
  - ⋆ Can provide unbounded improvement on effective gain compared to separable schemes[12].

[12] Compare our effective gain with any gain that has subpacketization S_1.

Joint exposition of coded caching and PHY$_1$



- Our result advocates for joint consideration between cache-placement and network structure.

- Joint consideration of coded caching and PHY yields very substantial improvements in effective DoF.

  - ⋆ Exploits network structure.
  - ⋆ Can still maintain some robustness to not knowing network structure for cache-placement.

- Revealed an instance where non-separated schemes have unboundedly better effective gains over separable schemes.

Practicality and timeliness of result

- Scheme consists of basic implementable ingredients
  - ⋆ Zero forcing
  - ⋆ Low-dimensional coded caching
- Works for all values of $K, L, \gamma, K_T, \gamma_T$.

HAVING EXTRA TRANSMITTING ANTENNAS (SERVERS) MAKES CODED CACHING EVEN MORE APPLICABLE IN PRACTICE

- At a time when subpacketization complexity is the clear major bottleneck of coded caching.
- At a time when multiple antennas and transmitter cooperation are standard ingredients in wireless communications.

Bibliography

Bibliography

[1] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[2] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1146–1158, Feb 2017.

[3] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," in *2015 Information Theory and Applications Workshop (ITA)*, Feb 2015, pp. 98–107.

[4] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "On the average performance of caching and coded multicasting with random demands," in *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, Aug 2014, pp. 922–926.

[5] S. S. Bidokhti, M. Wigger, and R. Timo, "Erasure broadcast networks with receiver caching," in *2016 IEEE International Symposium on Information Theory (ISIT)*, July 2016, pp. 1819–1823.

[6] J. Zhang and P. Elia, "Wireless coded caching: A topological perspective," *preprint arXiv:1606.08253*, 2016.

[7] A. Ghorbel, M. Kobayashi, and S. Yang, "Cache-enabled broadcast packet erasure channels with state feedback," *preprint arXiv:1509.02074*, 2015.

[8] J. Zhang and P. Elia, "Fundamental limits of cache-aided wireless bc: Interplay of coded-caching and csit feedback," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3142–3160, May 2017.

[9] J. Hachem, N. Karamchandani, and S. Diggavi, "Content caching and delivery over heterogeneous wireless networks," in *IEEE Conference on Computer Communications (INFOCOM)*, 2015.

[10] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb 2016.

[11] F. Engelmann and P. Elia, "A content-delivery protocol, exploiting the privacy benefits of coded caching," in *Proc. WiOpt*, May 2017, pp. 1–6.

[12] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," *preprint arXiv:1511.02256*, 2015.

[13] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *preprint arXiv:1609.07817*, 2016.

[14] S.-E. Elayoubi and J. Roberts, "Performance and cost effectiveness of caching in mobile access networks," in *Proc. of the 2nd International Conference on Information-Centric Networking*, 2015.

[15] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis., "Finite-length analysis of caching-aided coded multicasting," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5524–5537, Oct 2016.

[16] M. Ji, K. Shanmugam, G. Vettigli, J. Llorca, A. M. Tulino, and G. Caire, "An efficient multiple-groupcast coded multicasting scheme for finite fractional caching," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 3801–3806.

[17] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions on Networking*, 2015.

[18] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design in centralized Coded Caching scheme," *preprint arXiv:1510.05064*, 2015.

[19] L. Tang and A. Ramamoorthy, "Coded caching with low subpacketization levels," *preprint arXiv:1607.07920*, 2016.

[20] C. Shangguan, Y. Zhang, and G. Ge, "Centralized coded caching schemes: A hypergraph theoretical approach," *preprint arXiv:1608.03989*, 2016.

[21] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using Ruzsa-Szemeredi graphs," *preprint arXiv:1701.07115*, 2017.

[22] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7253–7271, Dec 2016.

[23] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, "Fundamental limits of cache-aided interference management," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3092–3107, May 2017.

[24] A. Sengupta, R. Tandon, and O. Simeone, "Cache aided wireless networks: Tradeoffs between storage and latency," *preprint arXiv:1512.07856*, 2015.

[25] Y. Cao, M. Tao, F. Xu, and K. Liu, "Fundamental storage-latency tradeoff in cache-aided MIMO interference networks," *preprint arXiv:1609.01826*, 2016.

[26] J. Hachem, U. Niesen, and S. N. Diggavi, "Degrees of Freedom of cache-aided wireless interference networks," *preprint arXiv:1606.03175*, 2016.

[27] J. S. P. Roig, F. Tosato, and D. Gündüz, "Interference networks with caches at both ends," *preprint arXiv:1703.04349*, 2017.

[28] S. Yang, K. H. Ngo, and M. Kobayashi, "Content delivery with coded caching and massive MIMO in 5G," in *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Sept 2016, pp. 370–374.

[29] S. P. Shariatpanahi, G. Caire, and B. H. Khalaj, "Multi-antenna coded caching," *preprint arXiv:1701.02979*, 2017.

[30] E. Piovano, H. Joudeh, and B. Clerckx, "On coded caching in the over-loaded MISO broadcast channel," *preprint arXiv:1702.01672*, 2017.

[31] J. Zhang, F. Engelmann, and P. Elia, "Coded caching for reducing CSIT-feedback in wireless communications," in *Proc. Allerton Conf. Communication, Control and Computing*, Sep. 2015.

Bibliography$_5$

[32] J. Zhang and P. Elia, "Feedback-aided coded caching for the MISO BC with small caches," *preprint arXiv:1606.05396*, 2016.

[33] M. A. Maddah-Ali and U. Niesen, "Cache-aided interference channels," in *2015 IEEE International Symposium on Information Theory (ISIT)*, June 2015, pp. 809–813.

[34] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, "On the optimality of separation between caching and delivery in general cache networks," *preprint arXiv:1701.05881*, 2017.

THANK YOU