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

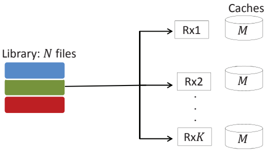
Eleftherios Lampiris and Petros Elia

EURECOM, Sophia Antipolis

February 11, 2018

EURECOM
Sophia Antipolis

Coded caching

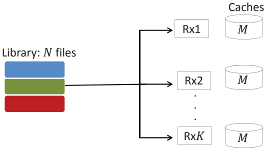


- Coded caching in single-stream BC[1]
- N files, K receivers, cache size M.
- Normalized cache size per user

$$\gamma \triangleq M/N$$

[1] Maddah-Ali & Niesen

Coded caching



- Coded caching in single-stream BC[1]
- N files, K receivers, cache size M.
- Normalized cache size per user

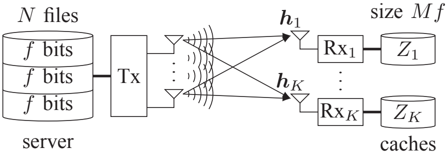
$$\gamma \triangleq M/N$$

Sum-DoF and caching gain

$$d_1(\gamma) = 1 + K\gamma = 1 + G$$

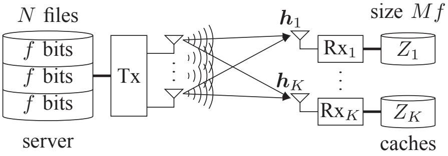
[1] Maddah-Ali & Niesen

Coded caching with multiple antennas



- Same as before, but with <u>L transmit antennas</u>
- Aim: Complement caching gains, with additional multiplexing gains

Coded caching with multiple antennas



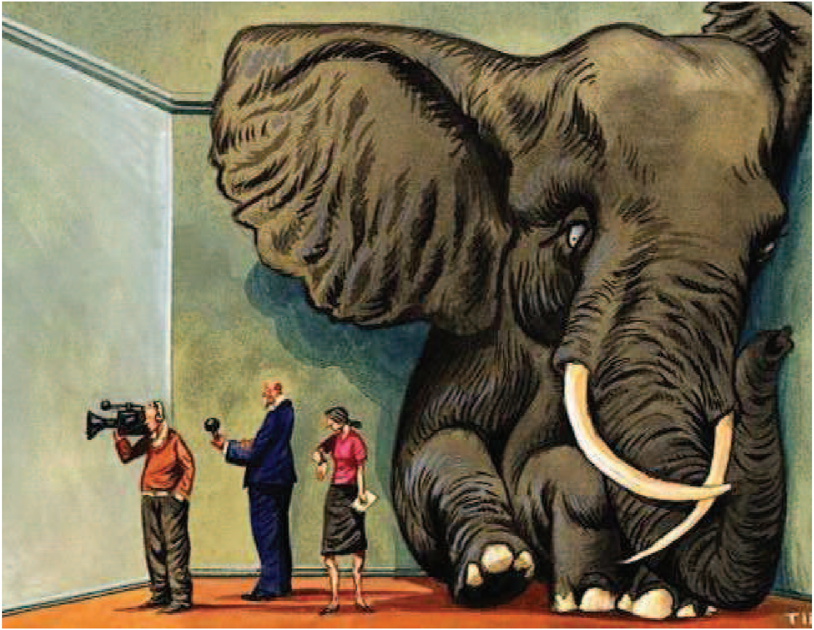
- Same as before, but with <u>L transmit antennas</u>
- Aim: Complement caching gains, with additional multiplexing gains

Sum-DoF of

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

MULTIPLEXING AND CACHING GAINS CAN BE COMBINED ADDITIVELY
(Shariatpanahi et al. — Naderializadeh et al.)

Subpacketization bottleneck of coded caching



Subpacketization bottleneck of coded caching

SUBPACKETIZATION PROBLEM

Subpacketization bottleneck of coded caching

Subpacketization problem

Theoretical gain $G = K\gamma$ needed astronomical file size $\gg$:

$$S = \binom{K}{K\gamma} \overset{\approx}{\longrightarrow} \binom{K}{K\gamma}\binom{K}{L} \leq \mathbf{S_{MAX}}$$

Subpacketization bottleneck of coded caching

SUBPACKETIZATION PROBLEM

Theoretical gain $G = K\gamma$ *needed astronomical file size* $\gg$*:*

$$S = \binom{K}{K\gamma} \overset{\approx}{\longrightarrow} \binom{K}{K\gamma}\binom{K}{L} \leq \mathbf{S_{MAX}}$$

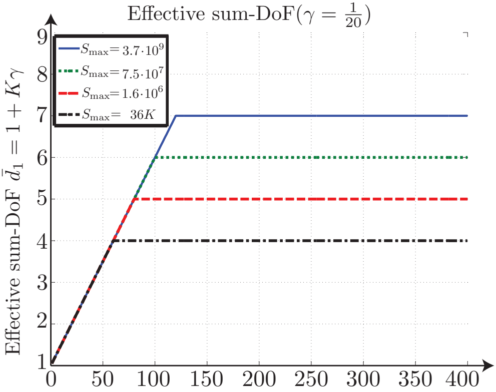
Can now encode over few users

$$\bar{K} = \arg\max_{K} \{S \leq S_{MAX}\}$$

Reduced EFFECTIVE CACHING GAIN

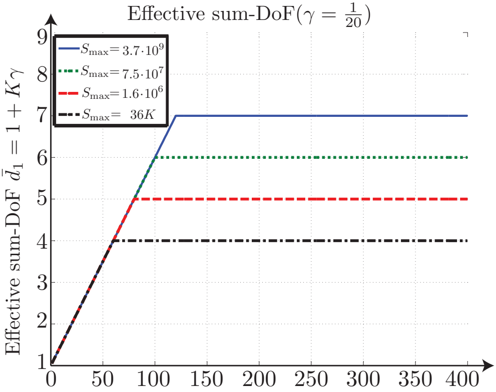
$$\bar{G} = \bar{K}\gamma \ll K\gamma$$

Diminished effective DoF from subpacketization



Effective DoF $\bar{d}_1 \triangleq 1 + \bar{K}\gamma$. MN algorithm. $L = 1$ tx-antenna. Cache-size $\gamma = 1/20$.

Diminished effective DoF from subpacketization



Effective DoF $\bar{d}_1 \triangleq 1 + \bar{K}\gamma$. MN algorithm. $L = 1$ tx-antenna. Cache-size $\gamma = 1/20$.

Unlike the theoretically unbounded caching gain $G = K\gamma$

THE EFFECTIVE ACTUAL GAIN REMAINED HARD-BOUNDED BY SMALL CONSTANTS

Subpacketization bottleneck - No ordinary elephant

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

- ***For any known coded caching algorithm***[2]

[2] Hypergraph #6 (Shangguan et al.). ($\bar{K} \approx 20000$ users). Theor. gain $G = 400$.

Subpacketization bottleneck - No ordinary elephant

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

- ***For any known coded caching algorithm***[2]
- ***In any known single-antenna or multi-antenna setting***

[2] Hypergraph #6 (Shangguan et al.). ($\bar{K} \approx 20000$ users). Theor. gain $G = 400$.

Subpacketization bottleneck - No ordinary elephant

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

- ***For any known coded caching algorithm***[2]
- ***In any known single-antenna or multi-antenna setting***

2

[2] Hypergraph #6 (Shangguan et al.). ($\bar{K} \approx 20000$ users). Theor. gain $G = 400$.

Main results

- We will show that having multiple antennas in fact dramatically reduces subpacketization
- By reducing the number of different caches necessary.

Main results

- We will show that having multiple antennas in fact dramatically reduces subpacketization
- By reducing the number of different caches necessary.

EXAMPLE: $K = 50$, $L = 5$ *and* $\gamma = \frac{M}{N} = \frac{3}{10}$

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

• *Previously needed subpacketization*

$$S_{MN} \approx 10^{12} \longrightarrow \mathbf{S_{MS} \approx 10^{17}}$$

Main results

- We will show that having multiple antennas in fact dramatically reduces subpacketization
- By reducing the number of different caches necessary.

EXAMPLE: $K = 50$, $L = 5$ *and* $\gamma = \frac{M}{N} = \frac{3}{10}$

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

- *Previously needed subpacketization*

$$S_{MN} \approx 10^{12} \longrightarrow \mathbf{S_{MS} \approx 10^{17}}$$

We will achieve desired sum-DoF with subpacketization

$$\mathbf{S_L = 120}.$$

Main results

Theorem

In the K-user cache-aided MISO BC with L tx-antennas, the sum-DoF

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

can be achieved with subpacketization

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

Main results

Corollary

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.*

Main results

Corollary

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.

Corollary

For $L = K\gamma$: can achieve DoF $L + K\gamma$ with subpacketization[a]

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

[a] Previously needed $\approx (\binom{K}{K\gamma})^2$.

Description of scheme

DESCRIPTION OF SCHEME
Via example

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 $L = 5$:

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

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 $L = 5$:
    $$\mathcal{G}_1 = \{1, 11, 21, 31, 41\}, \ldots, \mathcal{G}_{10} = \{10, 20, 30, 40, 50\}.$$

$\downarrow$ PROCEED AS IF $L = 1$, $K' = 10$ USERS $\downarrow$

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 $L = 5$:
    $$\mathcal{G}_1 = \{1, 11, 21, 31, 41\}, \ldots, \mathcal{G}_{10} = \{10, 20, 30, 40, 50\}.$$

$\downarrow$ PROCEED AS IF $L = 1$, $K' = 10$ USERS $\downarrow$

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

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

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

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

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

$$\vdots$$

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

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

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

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

$$\vdots$$

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

- 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

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

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

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

- Then simply transmit

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

  - $(\mathbf{H}^{\mathcal{G}_g})^{-1}$ 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 $\boldsymbol{w}_1^{(2,3,4)}$.

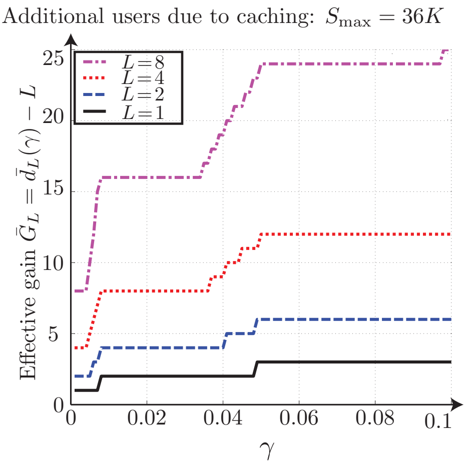
Scheme: $K = 50, L = 5, \gamma = 3/10$

- Then simply transmit

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

  - $(\mathbf{H}^{\mathcal{G}_g})^{-1}$ 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 $S_L = \binom{K'}{K'\gamma} = \binom{K/L}{K\gamma/L} = \binom{10}{3} = 120$.

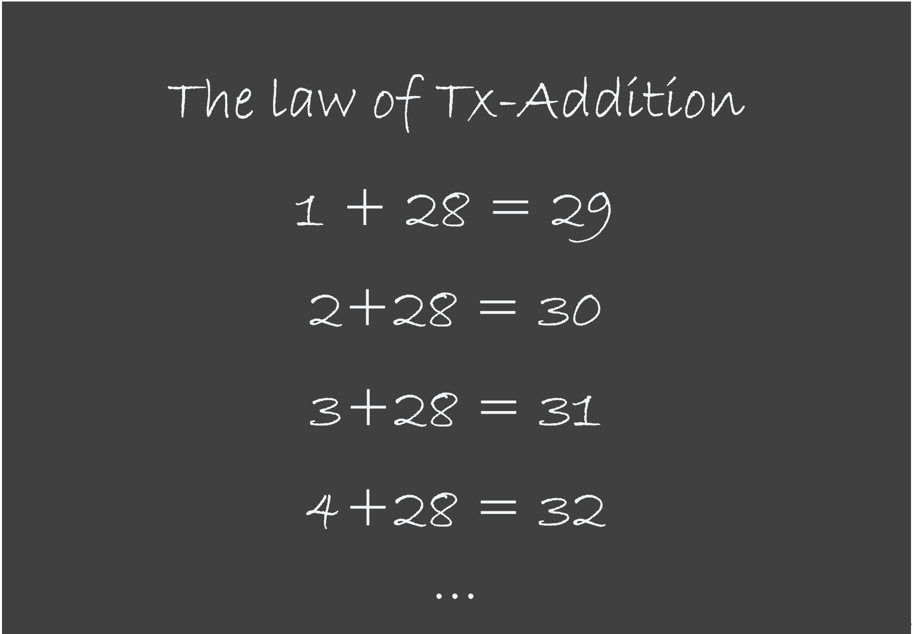
Multiplicative boost of caching gains



New effective caching gain $\bar{G}_L = d_L(\gamma) - L$ (maximized over all K) for different L.

SOME CONCLUSIONS

Combining gains – ∞ file size



Effective combining of gains – Constrained

The effective law of Tx-Addition

$$1 + 7 = 8$$

$$2+7 = 16$$

$$3+7 = 24$$

$$4+7 = 32$$

$$\ldots$$

Multiplicative DoF boost

Corollary

While in theory, adding antennas gives additive DoF increase

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

with finite file sizes, with adding antennas, the effective DoF:
- is either increased by a multiplicative factor of L

$$\bar{d}_L = L \cdot \bar{d}_1$$

- or it reaches the theoretical (unconstrained) DoF $d_L = L + K\gamma$.

Multiplicative boost of multi-transmitter systems

$$50 + 5 < 55$$

Improvement on MIMO system < 5 users

$$50+5 = 250$$

Lift entire MIMO system 5 times. 200 extra users

Bottleneck and scheme

- Scheme benefits from non-separability (joint caching and ‘PHY’)

Bottleneck and scheme

- Scheme benefits from non-separability (joint caching and ‘PHY’)
- Scheme is simple
  - ZF and Low-dimensional coded caching
  - Works for all values of $K, L, \gamma, K_T, \gamma_T$.
  - Multiple antennas is a standard ingredient in comm.

Bottleneck and scheme

- Scheme benefits from non-separability (joint caching and ‘PHY’)
- Scheme is simple
  - ZF and Low-dimensional coded caching
  - Works for all values of $K, L, \gamma, K_T, \gamma_T$.
  - Multiple antennas is a standard ingredient in comm.

- Subpacketization complexity is the **clear, major, brutal,... barbaric** bottleneck of coded caching

Bottleneck and scheme

- Scheme benefits from non-separability (joint caching and ‘PHY’)
- Scheme is simple
  - ZF and Low-dimensional coded caching
  - Works for all values of $K, L, \gamma, K_T, \gamma_T$.
  - Multiple antennas is a standard ingredient in comm.

- Subpacketization complexity is the **clear, major, brutal,... barbaric** bottleneck of coded caching
- Advocates for clever placement of transmitters
  - No longer just about multiplexing gains

Bottleneck and scheme

- Scheme benefits from non-separability (joint caching and ‘PHY’)
- Scheme is simple
  - ZF and Low-dimensional coded caching
  - Works for all values of $K, L, \gamma, K_T, \gamma_T$.
  - Multiple antennas is a standard ingredient in comm.

- Subpacketization complexity is the **clear, major, brutal,... barbaric** bottleneck of coded caching
- Advocates for clever placement of transmitters
  - No longer just about multiplexing gains
- Accentuates need/utility for reduced-subpacketization coded caching algorithms (designed for $L = 1$).
  - improvement $\longrightarrow L\times$ improvement

THANK YOU