Coded Caching Schemes with Reduced Subpacketization from Linear Block Codes

Li Tang and Aditya Ramamoorthy

December 4, 2018

Department of Electrical and Computer Engineering, Iowa State University (ISU)

Content Caching Networks

Content Caching Network

Akamai

edgecast

ChinaCache

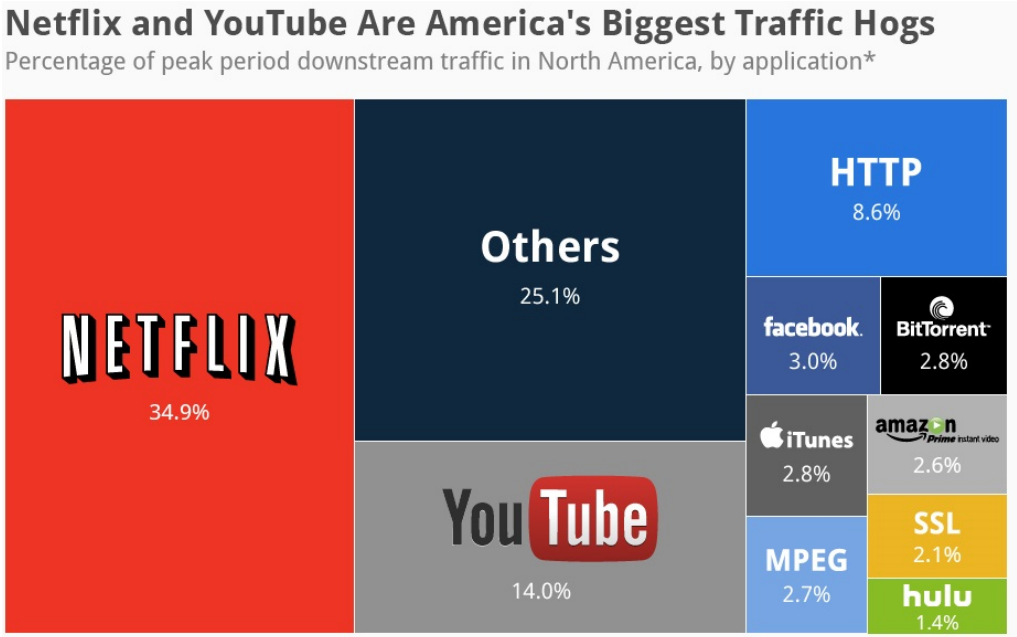
Level(3) COMMUNICATIONS

amazon web services™

fastly

Incapsula

Internet Video Traffic (source: Statista)



- Caching is the technique of choice for reducing video latency and improving end user experience.

One line summary of the talk

Linear block codes (under mild conditions) $\Longrightarrow$ **Caching schemes**

Content Delivery with Caching

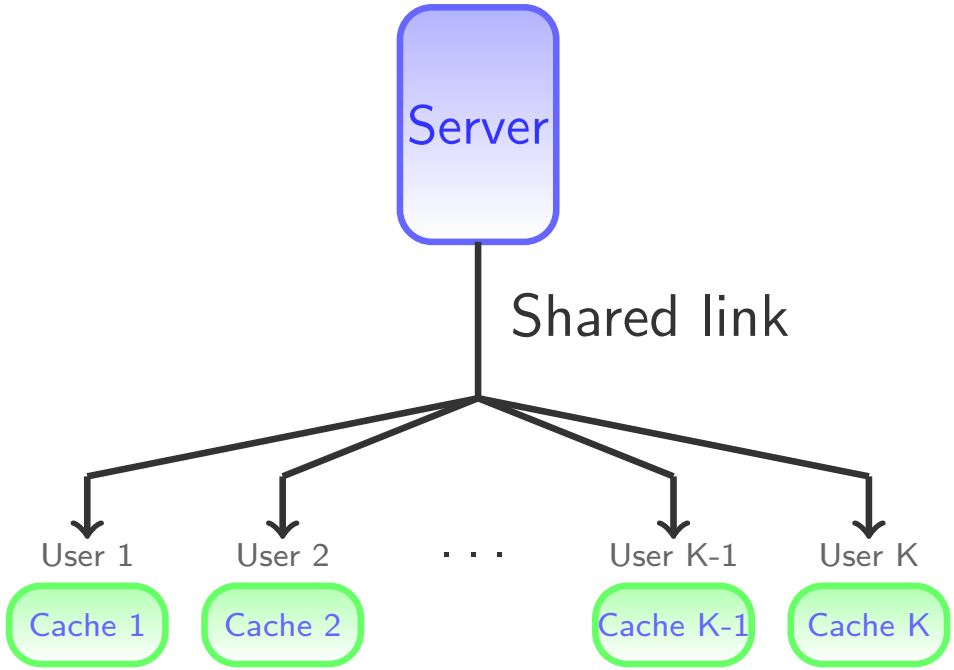
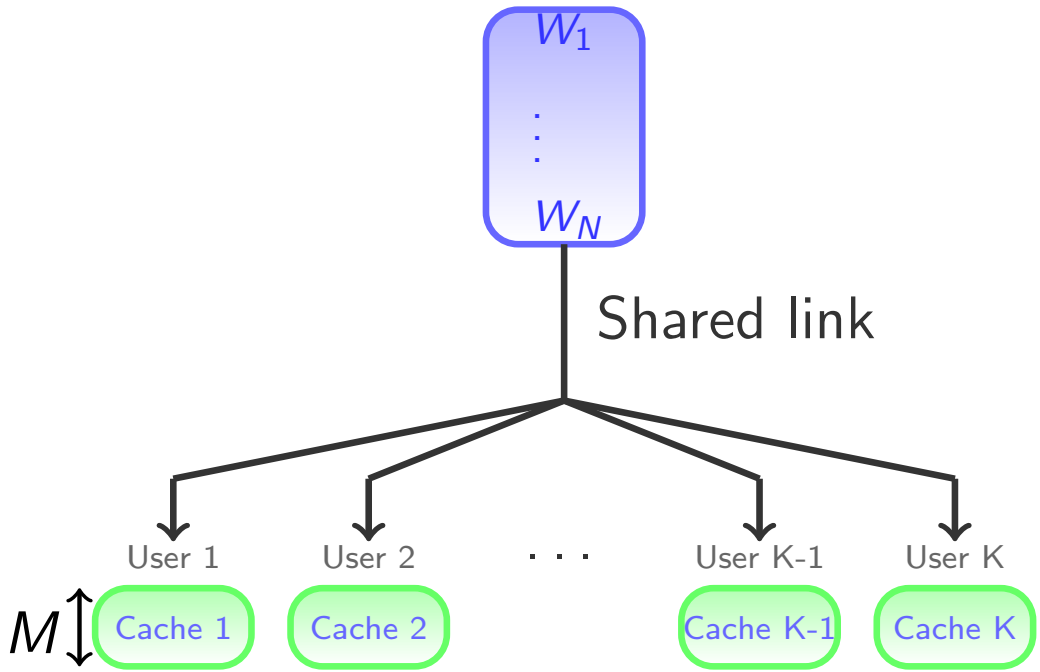


Figure: Content Delivery Network

- Conventional approach: clients cache portions of popular content.
- Coding in the cache and coded transmission from the server are typically not considered.

Coded Caching Formulation *[Maddah-Ali & Niesen '13]*

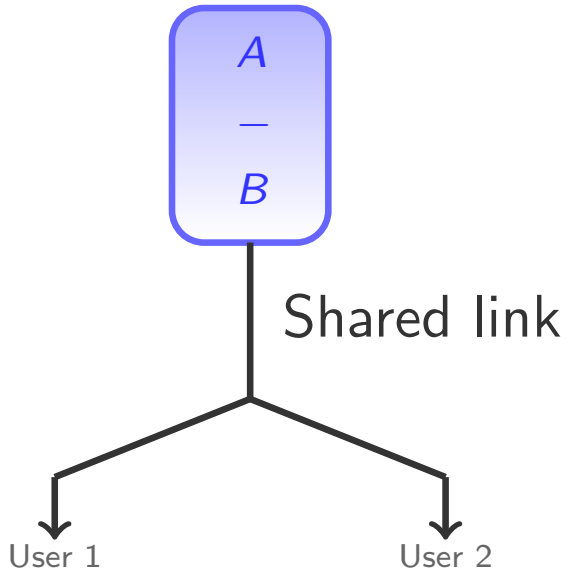
- Server contains a library of N files each of size F subfiles (a subfile is a unit of information, e.g., 1 Mb).
- K users each with a cache of size MF subfiles.
- The i-th user requests file W_{d_i} where $d_i \in \{1, \ldots, N\}$.



Caching Example with $N = K = 2$

Without caching

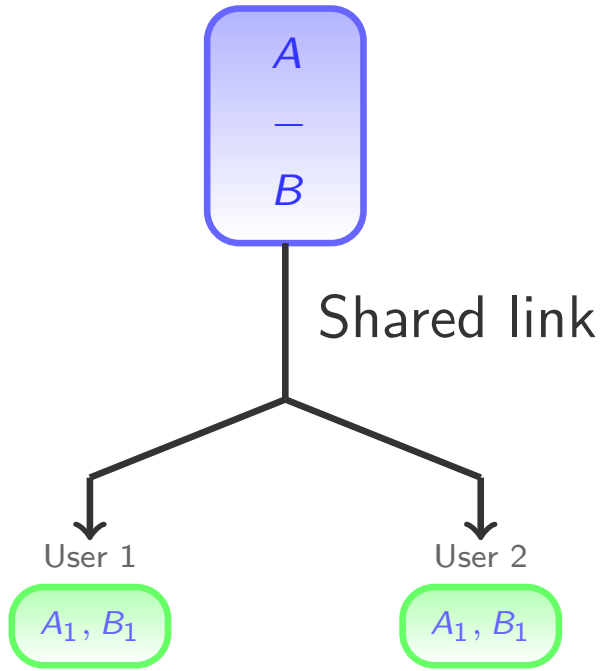
- User 1 requests A and User 2 requests B,
- Server has to send both A and B (two files),



Caching Example with $N = K = 2$

Conventional caching

- Split each file into two equal parts $A = \{A_1, A_2\}, B = \{B_1, B_2\}$.
- Off-peak hours: Users save $M/N = 1/2$ fraction of each file.
- Peak hours: Suppose users 1 and 2 request A and B.
- Server sends file parts A_2 and B_2; equivalent to one file.

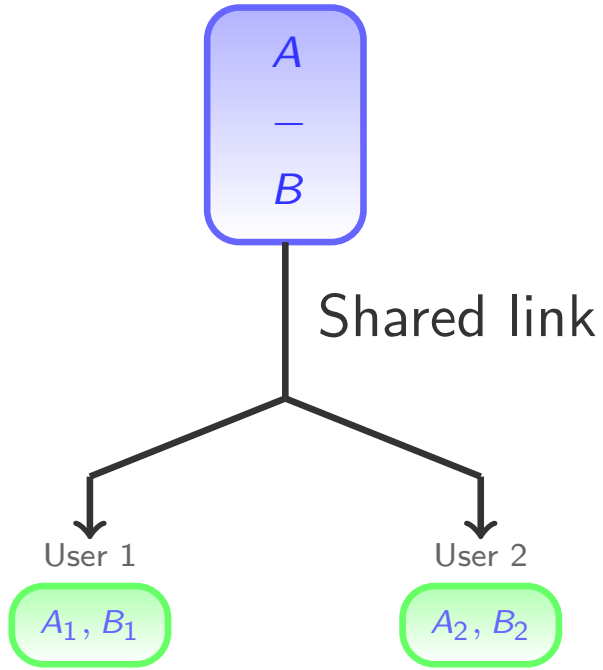


Caching Example with $N = K = 2$

Coded caching

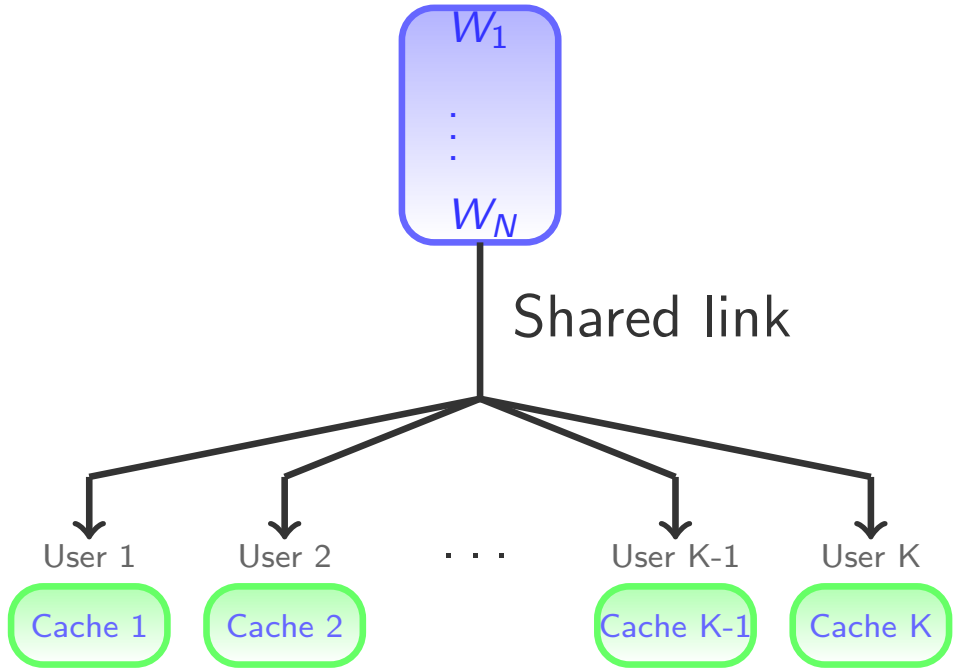
- Split each file into two equal parts $A = \{A_1, A_2\}, B = \{B_1, B_2\}$.
- Off-peak hours: Users cache carefully chosen subsets of the file
  - User 1 caches $Z_1 = \{A_1, B_1\}$.
  - User 2 caches $Z_2 = \{A_2, B_2\}$.
- Peak hours

User 1 Req.	User 2 Req.	Shared link
A	B	$A_2 + B_1$
B	A	$B_2 + A_1$
A	A	$A_2 + A_1$
B	B	$B_2 + B_1$



Over all demand patterns, transmitted rate is at most half a file!!

1. *Placement phase*: Caches are populated. Does NOT depend on actual user requests (off-peak hours).
2. *Delivery phase*: Server transmits a signal of rate RF subfiles over the shared link so that each user's request is satisfied (peak hours).



Potential gains can be huge!

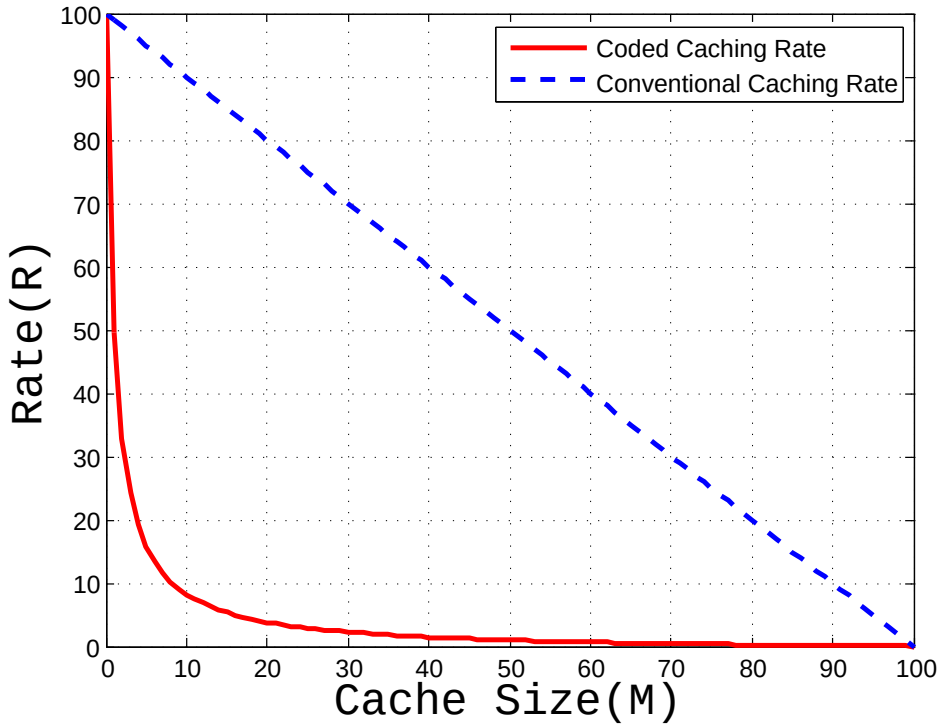
- Number of files N, number of users K, normalized cache per user M. When $N \geq K$

$$R^{\text{uncoded}} = K\left(1 - \frac{M}{N}\right)$$

vs.

$$R^{\text{coded}} = K\left(1 - \frac{M}{N}\right)\frac{1}{1 + \frac{KM}{N}}$$

System with $N = 100, K = 100$



As with most things ...

Source: www.appitierre.com

Practical Issues: Subpacketization Level

- Huge gains of coded caching require each file to be divided into a large number of parts (exponential in problem parameters). Subpacketization level F_s

$$F_s = \binom{K}{KM/N} \approx 2^{KH_2(M/N)}$$

- Example: System with $K = 50$ users, with $M/N = 0.4$, original scheme needs $F_s \approx 10^{14}$.
  - At the bare minimum each file should be of size 100 terabits.
  - In practice, much higher since the basic unit of storage on a computer is a sector.

Related Work

- Problem was first noticed and studied in the context of decentralized coded caching in the work of [Shanmugam et al.].
- For the centralized case, a placement delivery array (PDA) based scheme was proposed in [Yan et al.].
  - Their result can be obtained as a specific case of our result.
- Prior work by us [Tang-Ramamoorthy] also matches the parameters in [Yan et al.].

Related Work

- A hypergraph perspective introduced by [Shangguan et al.]
  - Demonstrates that (asymptotically) constant rate with subpacketization F_s linear in K is impossible.
  - Shows the existence of schemes where subpacketization scales as $\exp(c\sqrt{K})$.
- Work by [Shanmugam et al.] demonstrates connections with induced matchings in Ruzsa-Szemeredi graphs.
  - Shows the existence of schemes where subpacketization is linear but rate scales as K^{δ}, where $0 < \delta < 1$.
- Both these results require rather large values of K and are not applicable in a real-life scenarios.

Designs

Definition

A design is a pair $(X, \mathcal{A})$ such that

1. X is a set of elements called points, and
2. $\mathcal{A}$ is a collection (i.e., multiset) of nonempty subsets of X called blocks, where each block contains the same number of points.

$$X = \{1, 2, 3, 4\} \text{ (Point set)}$$

$$\mathcal{A} = \{\{1, 2\}, \{3, 4\}, \{1, 3\}, \{2, 4\}, \{1, 4\}, \{2, 3\}\} \text{ (Block set)}$$

Resolvable design

Definition

A parallel class $\mathcal{P}$ in a design $(X, \mathcal{A})$ is a subset of disjoint blocks from $\mathcal{A}$ whose union is X. A partition of $\mathcal{A}$ into several parallel classes is called a resolution, and $(X, \mathcal{A})$ is said to be a resolvable design if $\mathcal{A}$ has at least one resolution.

$$X = \{1, 2, 3, 4\} \text{ (Point set)}$$

$$\mathcal{A} = \{\{1, 2\}, \{3, 4\}, \{1, 3\}, \{2, 4\}, \{1, 4\}, \{2, 3\}\} \text{ (Block set)}$$

Each parallel class is a partition of X.

P1	P2	P3
{1, 2}	{1, 3}	{1, 4}
{3, 4}	{2, 4}	{2, 3}

Linear Codes and Resolvable Designs

Let $q = 2$, $k = 3$. Consider a $(3, 2)$ Single Parity Check code over $\mathbb{Z}_2$,

$$\mathbf{G}_{SPC} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \text{ (Generator Matrix)}$$

Columns of matrix $\mathbf{T}$ list all $2^2 = 4$ codewords.

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

obtained by computing $\mathbf{c} = \mathbf{u} \cdot \mathbf{G}_{SPC}$ for all possible message vectors $\mathbf{u}$.

Codes and Resolvable Designs

- Points correspond to columns $\{1, \ldots, 4\}$.

$$\mathbf{T} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

- Blocks are constructed by identifying column index of points in a row.

$$\begin{aligned} [0\ 0\ 1\ 1] &\Rightarrow \mathcal{P}_1 : \{1,2\},\{3,4\} \\ [0\ 1\ 0\ 1] &\Rightarrow \mathcal{P}_2 : \{1,3\},\{2,4\} \\ [0\ 1\ 1\ 0] &\Rightarrow \mathcal{P}_3 : \{1,4\},\{2,3\} \end{aligned}$$

Usage in a coded caching scenario

- Users correspond to blocks (total of six). System with $K = 6$ users and $N \geq K$.

$$\{1,2\} \Rightarrow U_{\{1,2\}} \quad \{3,4\} \Rightarrow U_{\{3,4\}}$$
$$\{1,3\} \Rightarrow U_{\{1,3\}} \quad \{2,4\} \Rightarrow U_{\{2,4\}}$$
$$\{1,4\} \Rightarrow U_{\{1,4\}} \quad \{2,3\} \Rightarrow U_{\{2,3\}}$$

- Subfiles correspond to points (total of four). Split each file into 4 subfiles $W_n = \{W_{n,1}, \ldots, W_{n,4}\}$.
- For original scheme, user corresponds to points (K points), subfile corresponds to $\binom{K}{\frac{KM}{N}}$ blocks (all subsets of size $\frac{KM}{N}$ of $\{1, \cdots, K\}$)

Usage in a coded caching scenario

- Placement Phase: User $U_{\{i,j\}}$ caches for each file W_n in the library,

$$Z_{i,j} = \{W_{n,i}, W_{n,j}\}$$

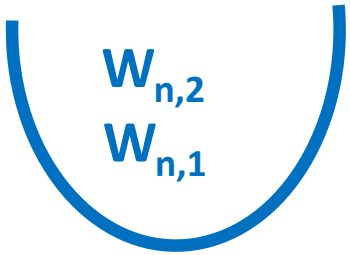
  - For example, user $U_{\{1,2\}}$ caches $\{W_{n,1}, W_{n,2}\}_{n=1}^{N}$.
  - Each user thus caches exactly half of each file. Therefore,

$$\frac{M}{N} = \frac{1}{2}.$$

Usage in a coded caching scenario

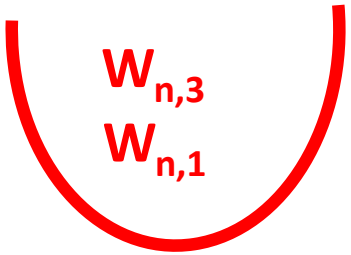
- Delivery Phase: Suppose that each user requests a distinct file (worst case). Thus user $U_{\{i,j\}}$ requests $W_{d_{ij}}$.
- How do we leverage coding in this case?
- Transmit equations that are simultaneously useful to multiple users!!

Example of delivery phase scheme



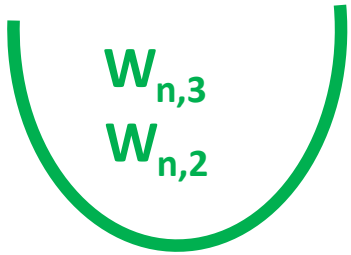
User U_{12}

Wants
$W_{d12,3}$
$W_{d12,4}$



User U_{13}

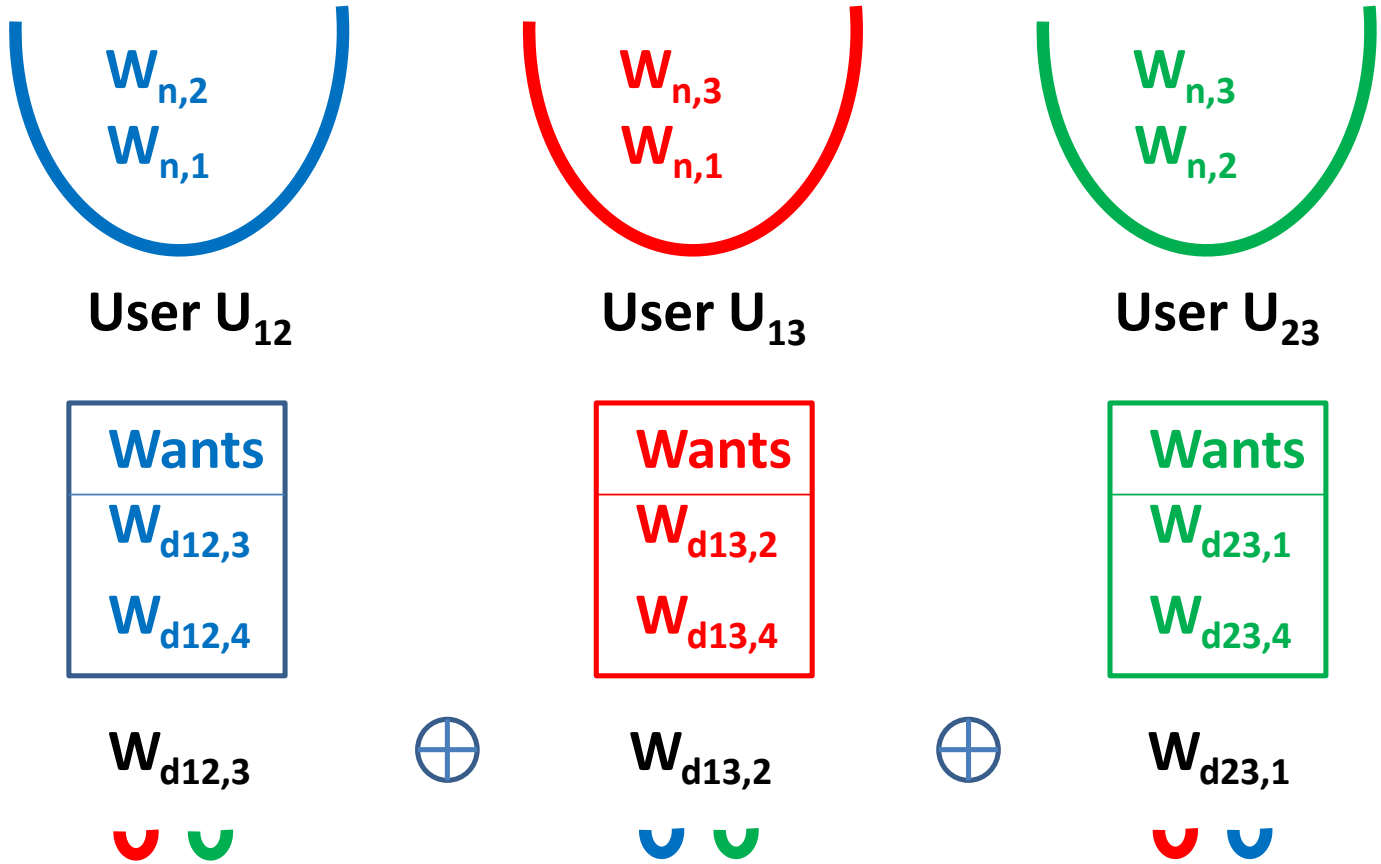
Wants
$W_{d13,2}$
$W_{d13,4}$



User U_{23}

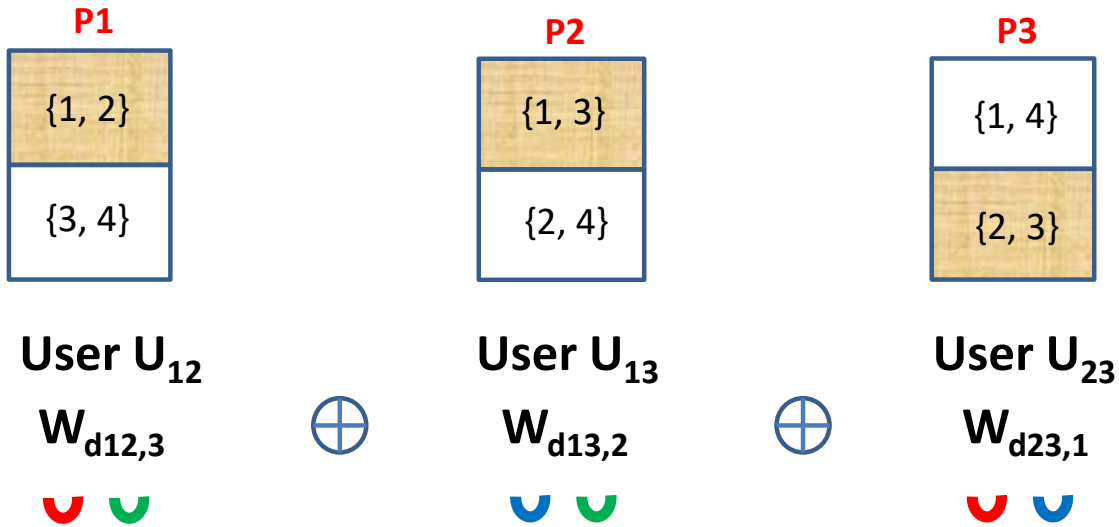
Wants
$W_{d23,1}$
$W_{d23,4}$

Example of delivery phase scheme



Example of delivery phase scheme

Any two blocks from distinct parallel classes have one intersection.



- Any two of blocks share one point that does not contains in the third block. ($\{1,2\},\{1,3\}$ contain point 1 which is not in $\{2,3\}$).
  - The missing subfile of one user in the equation has been cached in another two users.
- Four sets of blocks. Four equations.

Rate Analysis

- Four such equations can be found. Each equation corresponds to $\frac{1}{4}$-th of a file. Total rate is therefore 1 (normalized by file size).
- System has $K = 6$ users, $N \geq K$, $M/N = 1/2$ and a subpacketization level of $F_s = 4$.
- Original scheme would have subpacketization level of $\binom{6}{3} = 20$ and a rate of 0.75.

Rate Analysis

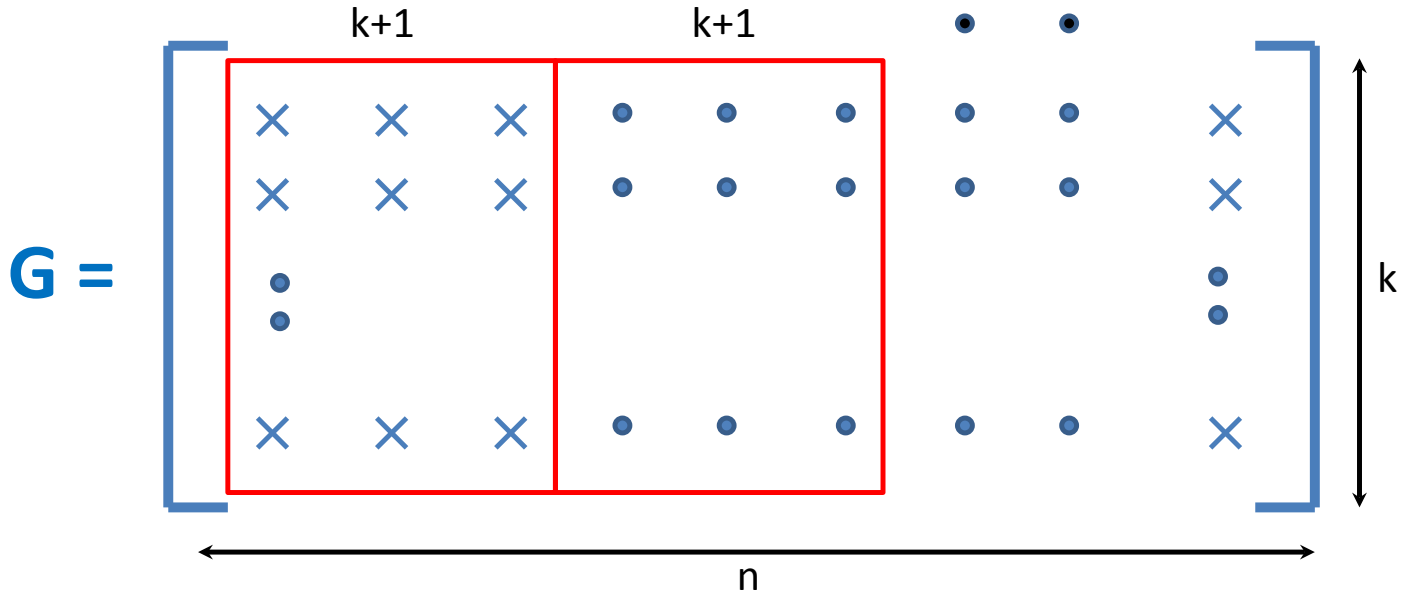
- Four such equations can be found. Each equation corresponds to $\frac{1}{4}$-th of a file. Total rate is therefore 1 (normalized by file size).
- System has $K = 6$ users, $N \geq K$, $M/N = 1/2$ and a subpacketization level of $F_s = 4$.
- Original scheme would have subpacketization level of $\binom{6}{3} = 20$ and a rate of 0.75.

Generalization of approach to a larger class of linear block codes exists....

$(k, k+1)$- Consecutive Column Property (CCP)

- Generator matrix **G** of size $k \times n$ over $GF(q)$. Let z be the *least* integer such that $k+1 \mid nz$
- Recovery Set: Consider the columns of **G** specified by $\mathcal{S}_a$

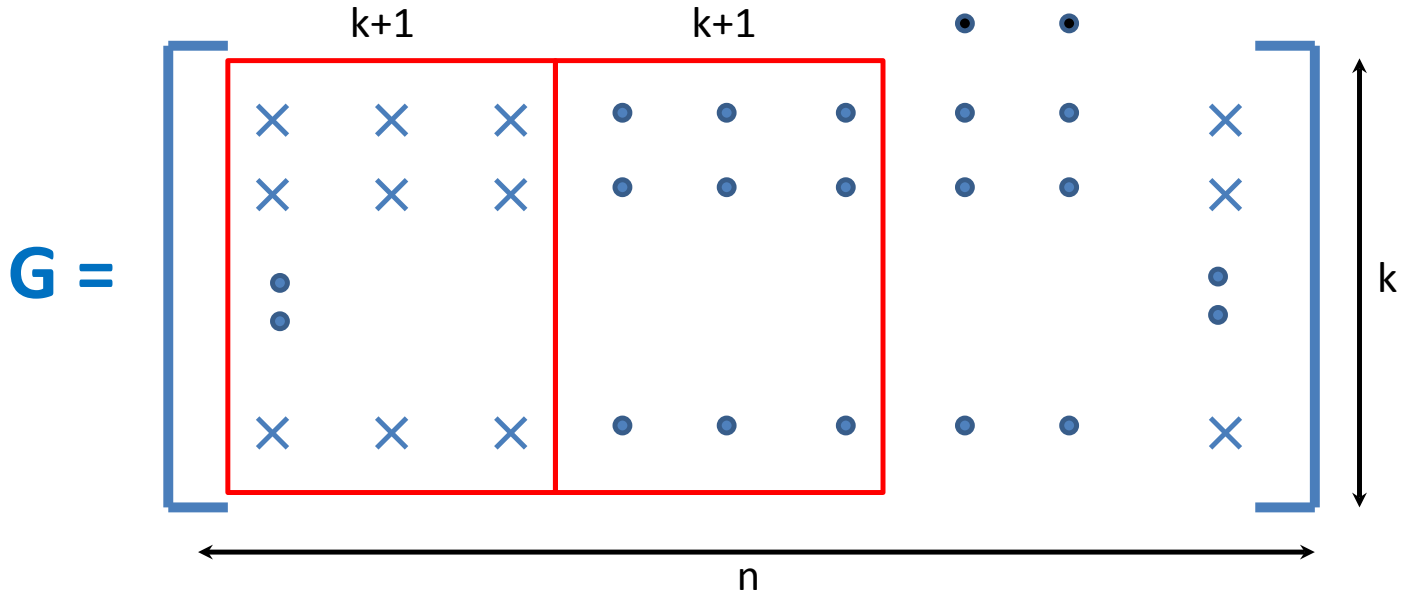
$$\mathcal{S}_a = \{a(k+1), \cdots, a(k+1)+k\} \mod n.$$



$(k, k+1)$- Consecutive Column Property (CCP)

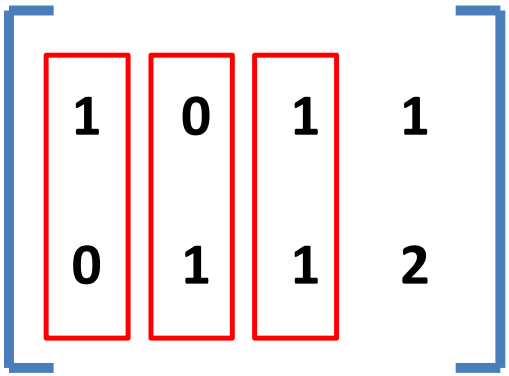
Definition

Consider the submatrices of $\mathbf{G}$ specified by $\mathbf{G}_{\mathcal{S}_a}$ for $0 \leq a \leq \frac{zn}{k+1} - 1$. We say that $\mathbf{G}$ satisfies the $(k, k+1)$-consecutive column property if all $k \times k$ submatrices of each $\mathbf{G}_{\mathcal{S}_a}$ are full rank.



CCP Example - $(4, 2)$ code over $GF(3)$

$$\mathcal{S}_0 = \{0, 1, 2\}$$

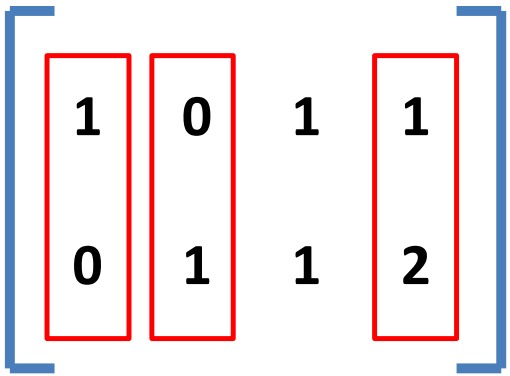


$$\mathbf{G}_{\mathcal{S}_0} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

All 2×2 submatrices are full-rank.

CCP Example - $(4, 2)$ code over $GF(3)$

$$\mathcal{S}_1 = \{3, 1, 0\}$$

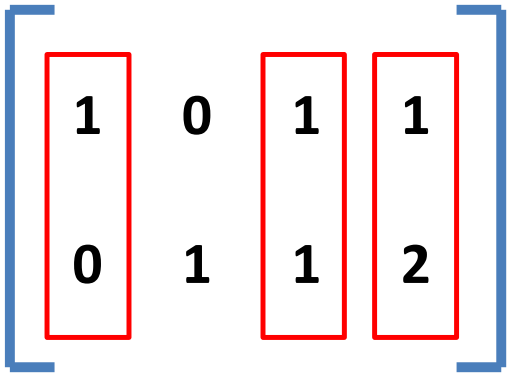


$$\mathbf{G}_{\mathcal{S}_1} = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

All 2×2 submatrices are full-rank.

CCP Example - $(4, 2)$ code over $GF(3)$

$$\mathcal{S}_2 = \{2, 3, 0\}$$

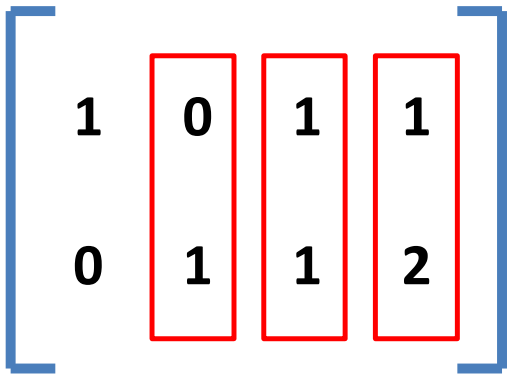


$$\mathbf{G}_{\mathcal{S}_2} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 0 \end{bmatrix}$$

All 2×2 submatrices are full-rank.

CCP Example - $(4, 2)$ code over $GF(3)$

$$\mathcal{S}_3 = \{1, 2, 3\}$$



$$\mathbf{G}_{\mathcal{S}_3} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

All 2×2 submatrices are full-rank.

Usage in coded caching scenario

$$\mathbf{T} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \\ 0 & 2 & 1 & 1 & 0 & 2 & 2 & 1 & 0 \end{bmatrix}.$$

- Point set

$$X = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}.$$

- Block set (Parallel class)

$$P_0 : \{\{0, 1, 2\}, \{3, 4, 5\}, \{6, 7, 8\}\}$$
$$P_1 : \{\{0, 3, 6\}, \{1, 4, 7\}, \{2, 5, 8\}\}$$
$$P_2 : \{\{0, 5, 7\}, \{1, 3, 8\}, \{2, 4, 6\}\}$$
$$P_3 : \{\{0, 4, 8\}, \{2, 3, 7\}, \{1, 5, 6\}\}$$

Usage in coded caching scenario

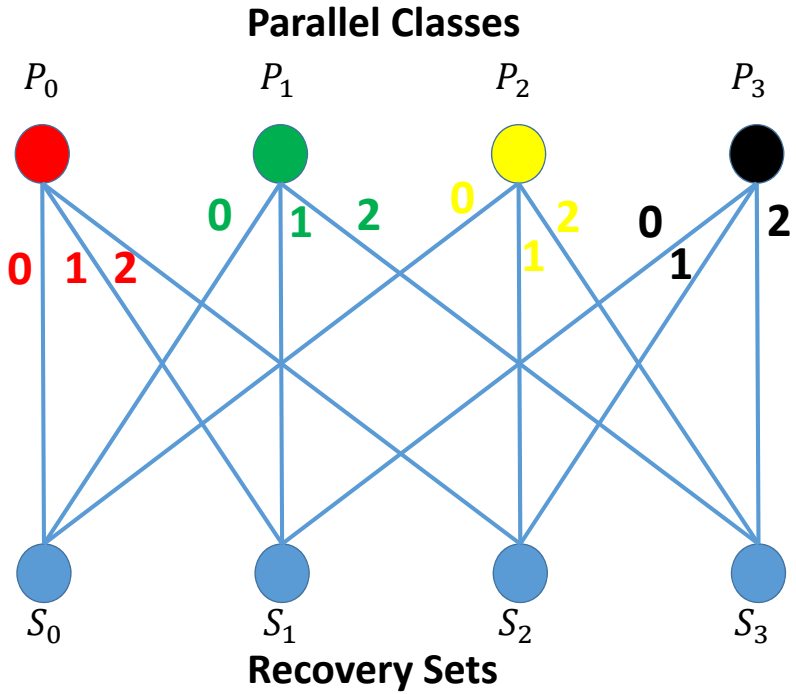
$$X = \{0,1,2,3,4,5,6,7,8\}$$
$$P_0 : \{\{0,1,2\},\{3,4,5\},\{6,7,8\}\}$$
$$P_1 : \{\{0,3,6\},\{1,4,7\},\{2,5,8\}\}$$
$$P_2 : \{\{0,5,7\},\{1,3,8\},\{2,4,6\}\}$$
$$P_3 : \{\{0,4,8\},\{2,3,7\},\{1,5,6\}\}$$

- Users correspond to blocks.
- Subfiles correspond to points and another index $\in \{0,\ldots,z-1\}$.

Example

User U_{012} caches $W^s_{n,0}, W^s_{n,1}, W^s_{n,2}$ for $s = 0,1,2$.

Delivery Phase Algorithm



- Each parallel class $\mathcal{P}_i$ recovers missing subfiles with a specific superscript for each user.
  - Similar to single-parity-check (SPC) code.
- Recovery set bipartite graph formalizes the idea.
- Recovery set $\mathcal{S}_0$: Constructive procedure that picks a user from each parallel class in $\mathcal{P}_0, \mathcal{P}_1$ and $\mathcal{P}_2$.

Delivery Phase Algorithm

- With some work, we can show that the number of user is $K = nq$, cache size is $\frac{M}{N} = \frac{1}{q}$, the rate of our scheme is

$$R^* = \frac{(q-1)n}{k+1}.$$

  Then coded gain is

$$g^* = k + 1.$$

- The subpacketization level is

$$F_s^* = q^k z$$

- Important fact: Both coded gain g^* and subpacketization level F_s^* increase with larger k!!

Delivery Phase Algorithm

- Scheme for $\frac{M}{N} = \frac{1}{q}$ can be converted into a scheme for $\frac{M}{N} = 1 - \frac{k+1}{nq}$ with

$$R^* = \frac{k+1}{(q-1)n}$$

$$F_s^* = (q-1)q^k \frac{zn}{k+1}$$

Some Linear Codes satisfying the CCP

- MDS code over $GF(q)$ of dimension $k \times n$
- Satisfies a stronger condition. *Any $k \times k$ submatrix is full-rank.*
  - Drawback: MDS codes typically need $q + 1 \geq n$.
  - Thus, with increasing n, the cached fraction $M/N = 1/q$ keeps decreasing.

Some Linear Codes satisfying the CCP

- A large class of cyclic codes!!

$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & \cdot & \cdot & \cdot & g_{n-k} & 0 & \cdot & \cdot & 0 \\ 0 & g_0 & g_1 & \cdot & \cdot & \cdot & g_{n-k} & 0 & \cdot & 0 \\ \vdots & & & & & & & & & \vdots \\ 0 & 0 & \cdot & 0 & g_0 & g_1 & \cdot & \cdot & \cdot & g_{n-k} \end{bmatrix}$$

- Any $k \times n$ generator matrix of a cyclic code over $GF(q)$ (if it exists) is such that
  - Any k consecutive columns are linearly independent.
  - We need slightly more... All $k \times k$ submatrices of $k+1$ consecutive columns are full-rank.
  - Equivalent to conditions on the generator polynomial of the cyclic code. Several examples of such codes can be found.

Some Linear Codes satisfying the CCP

- Constructions leveraging properties of smaller base matrices
  Eg. $\mathbf{A}$ satisfies the CCP, $\mathbf{G} = \mathbf{A} \otimes \mathbf{I}_{a \times a}$ satisfies the CCP.
- Some linear codes over $\mathbb{Z}$ mod q satisfy the CCP (q is not necessarily prime or prime power)
  - E.g. $(k+1, k)$ single parity check (SPC) code over $\mathbb{Z}$ mod q.
  - The conditions under which a resolvable design can be obtained by forming the matrix $\mathbf{T}$ are a little more involved.
- Some linear codes satisfy (k, α) CCP, $\alpha \leq k$.
  - α columns in each recovery set of size α have full rank.
  - It yields a lower coded gain $g = \alpha \leq k+1$ but larger range of parameters.
  - E.g. All cyclic codes satisfy (k, k)-CCP.

Example: $K = nq = 60$ users, $M/N = 1/q = 1/5$

k	F_s	g	R	Notes
-	1.4×10^{12}	13	3.69	Maddah-Ali & Niesen
11	4.9×10^{7}	12	4	
10	-	-	-	$(k, k+1)$-CCP code not exist
9	6.8×10^{6}	10	4.8	
8	1.2×10^{6}	9	5.3	
7	1.5×10^{5}	8	6	
6	10^{5}	7	6.85	
5	3125	6	8	
4	3125	5	9.6	
3	125	4	12	
2	25	3	16	
1	5	2	24	

Table: List of k values and corresponding F_s, g and R

Parameters: $K = nq$ users, $M/N = 1/q$ cache fraction, $F_s = q^k z$ subpacketization level

- Rate comparison (loss is tunable with choice of (n, k) code)

$$\frac{R^*}{R^{MN}} = \frac{n+1}{k+1}$$

- Subpacketization level (exponential improvement)

$$\frac{F_s^*}{F_s^{MN}} \approx q^{k-n} z \left(\frac{q-1}{q} \right)^{n(q-1)}.$$

Parameters: $K = nq$ users, $M/N = 1/q$ cache fraction, $F_s = q^k z$ subpacketization level

- Comparisons when rates of both schemes are almost the same is somewhat harder ...
- Use memory-sharing in Maddah-Ali & Niesen.

Parameters: $K = nq$ users, $M/N = 1/q$ cache fraction, $F_s = q^k z$ subpacketization level

- $(9, 5)$ linear block code over $GF(2)$ that satisfies CCP.

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

  Parameters: $K = 9 \times 2 = 18$, $\frac{M}{N} = \frac{1}{2}$, $R^* = \frac{3}{2}$ and $F_s^* = 64$.

- Maddah-Ali & Niesen approach would need memory-sharing between $\frac{M_1}{N_1} = \frac{2}{9}$ and $\frac{M_2}{N_2} = \frac{7}{9}$.
  $R^{MN} \approx 3/2$. However, the subpacketization level is
  $F_s^{MN} = \binom{K}{KM_1/N_1} + \binom{K}{KM_2/N_2} = 6120$. (6120 *vs* 64)

Summary

- Subpacketization is an important issue in coded caching.
- Our work establishes a simple and useful connection between classes of linear codes and placement schemes in coded caching.
- The schemes are easy to construct and analyze.
- Allow us to obtain schemes with exponentially smaller subpacketization levels compared to the original scheme.

Tang & R., "*Coded Caching Schemes with Reduced Subpacketization from Linear Block Codes*",
IEEE Trans. on Information Theory, May 2018.

Several Open Questions

- The subpacketization level of our construction still scales exponentially with the problem parameters.
  - Albeit with a much smaller exponent.
  - Provides constructions for tens of users with manageable subpacketization.

Several Open Questions

- The subpacketization level of our construction still scales exponentially with the problem parameters.
  - Albeit with a much smaller exponent.
  - Provides constructions for tens of users with manageable subpacketization.
  - Open issue: Constructions for general values of K that scale polynomially in problem parameters.

Several Open Questions

- The subpacketization level of our construction still scales exponentially with the problem parameters.
  - Albeit with a much smaller exponent.
  - Provides constructions for tens of users with manageable subpacketization.
  - Open issue: Constructions for general values of K that scale polynomially in problem parameters.
- Constructions that are flexible with respect to the cache fractions that they support.