Info-Theo 2 Fall 2024

CODED CACHING

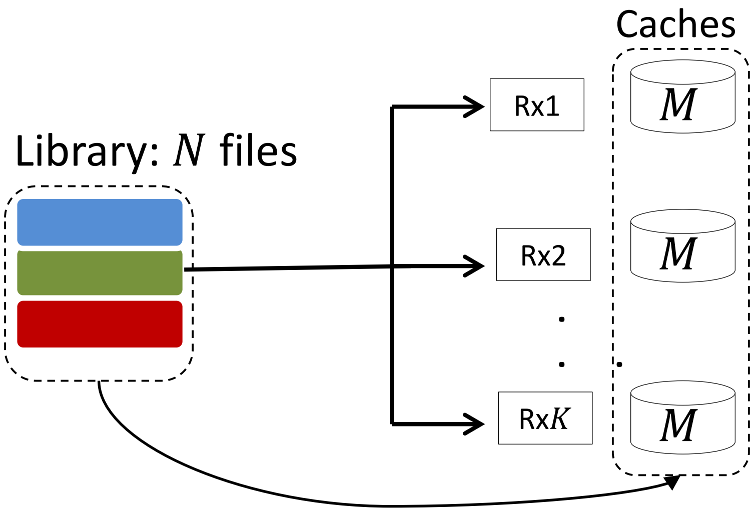
A powerful Information Theoretic Approach

PETROS ELIA

Two Problems Sharing Some Fundamentals

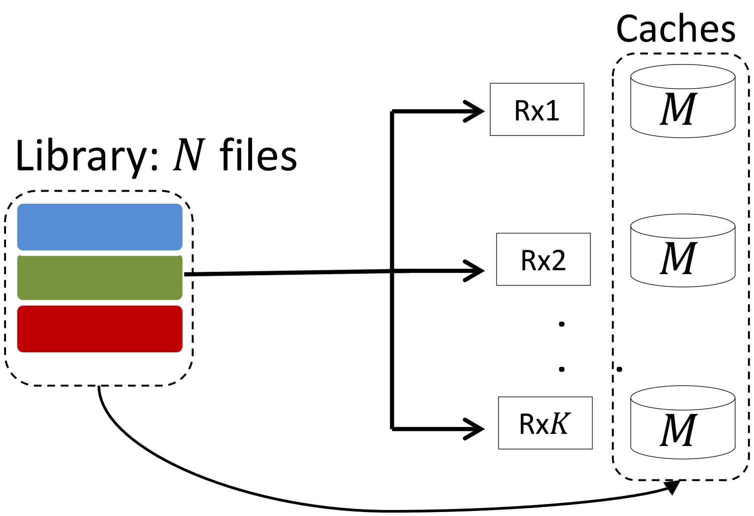
Two Problems Sharing Some Fundamentals

Cache-aided Communications

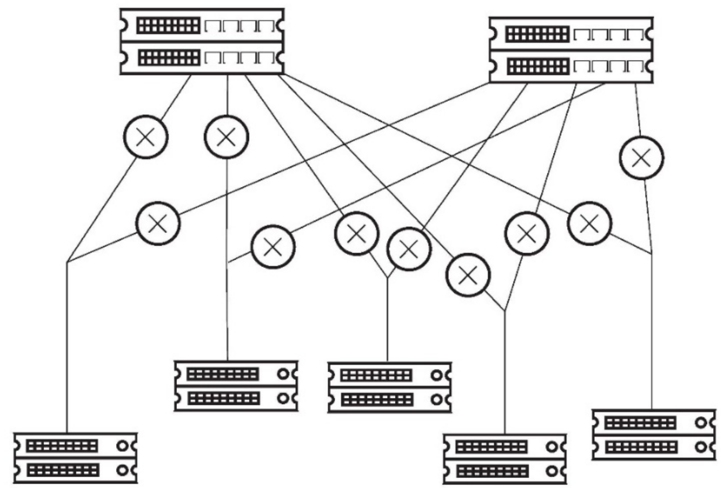


Two Problems Sharing Some Fundamentals

Cache-aided Communications

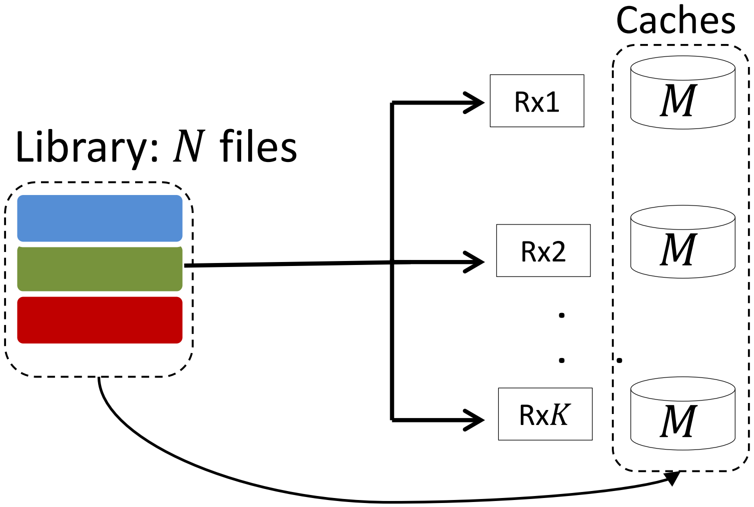


Communications in Distributed Computing

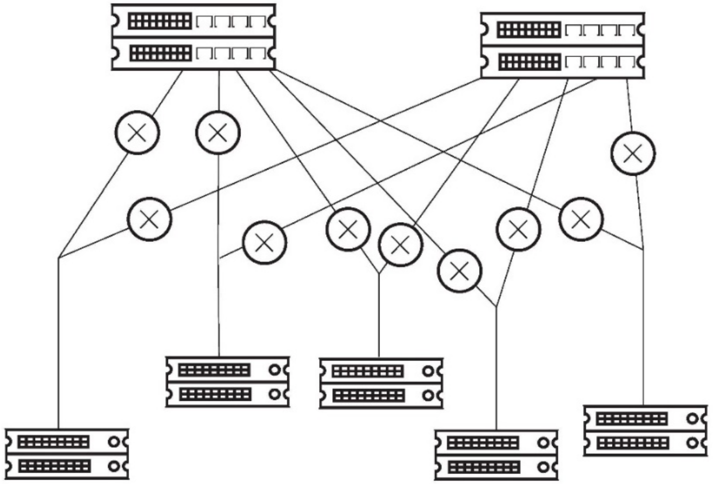


Two Problems Sharing Some Fundamentals

Cache-aided Communications



Communications in Distributed Computing

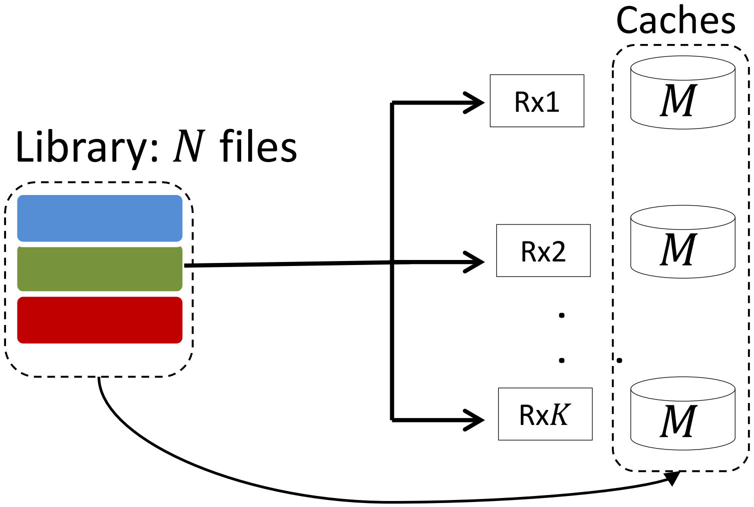


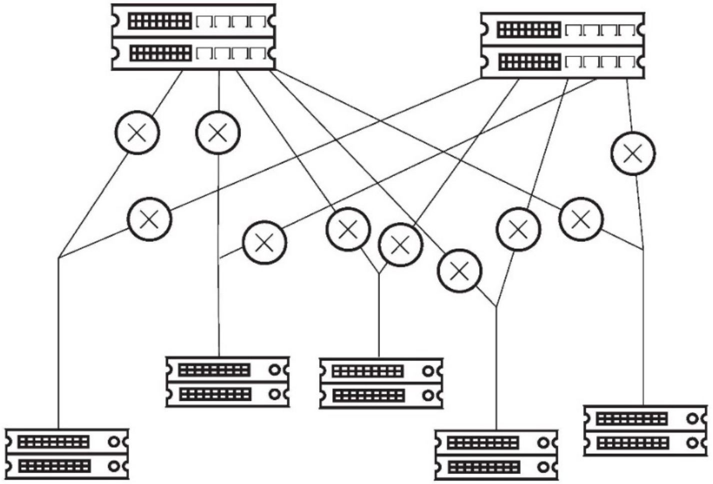
Common ingredients:
Communication cost and redundancy of stored/computed data

Two Problems Sharing Some Fundamentals

Cache-aided Communications

Communications in Distributed Computing





Common ingredients:
Communication cost and redundancy of stored/computed data

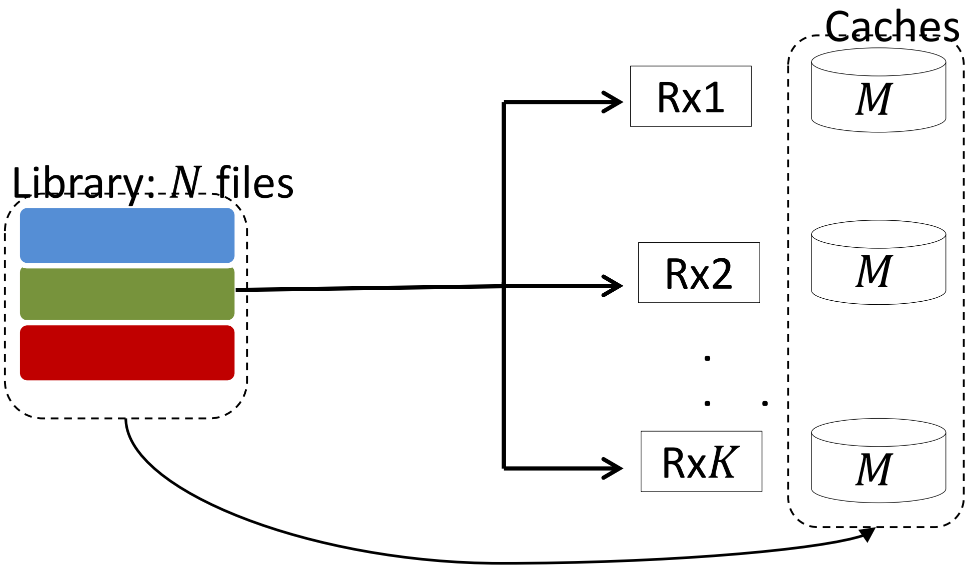
Common tool: Clique-based coding
Common Bottlenecks/Challenges

Outline

- Basic elements of coded caching (first communications)
  - Basic properties
  - Main gains & Important variants
- Ramifications of coded caching in distributed computing
  - Coded map reduce
  - Tradeoff between computing and communicating
  - Main bottlenecks
- Main challenges
  - Subpacketization: The bottleneck and new algorithmic solutions
  - Non uniformity: The bottleneck and new solutions
- Exploiting multiple dimensions to alleviate bottlenecks
  - Multiplicative gains by reducing subpacketization
  - Reducing effect of non uniformity
  - New coding structures
- Open problems and closing remarks

Simple Caching

First: Cache-Aided Communications

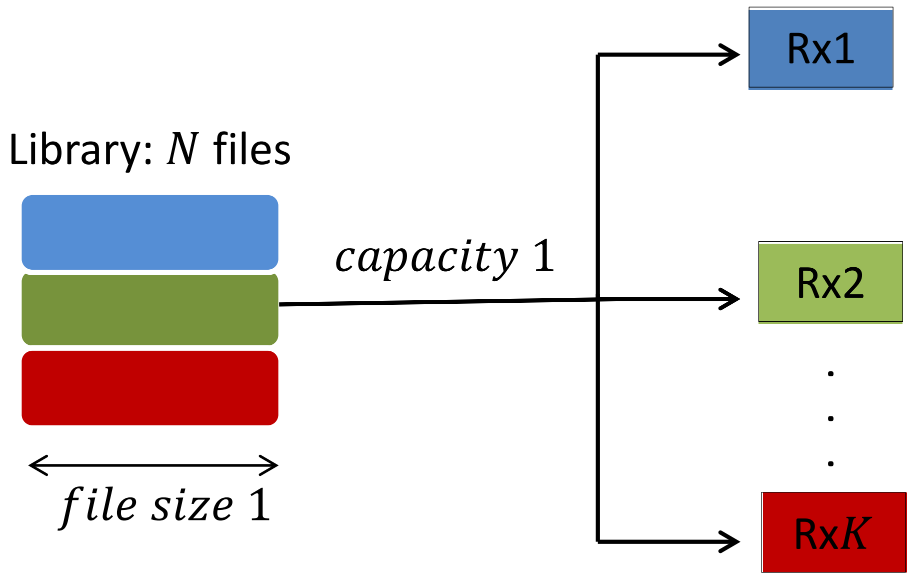


$$\gamma \overset{\text{def}}{=} \frac{M}{N} \overset{\text{def}}{=} \frac{\textit{indivial cache size}}{\textit{library size}}$$

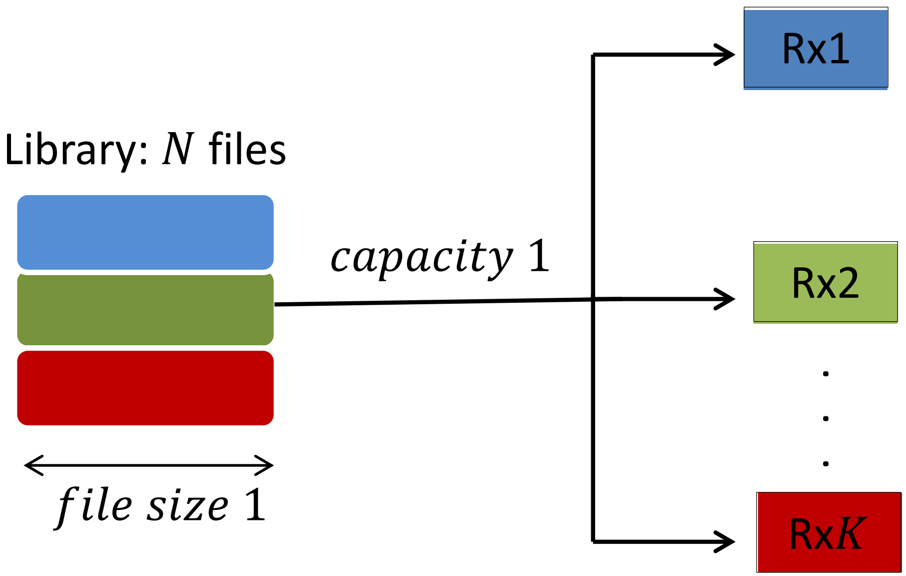
$T(\gamma)$: *duration of delivery phase*

OBJECTIVE: reduce $T(\gamma)$

Single-stream Channel: No Caching ($M = 0$)

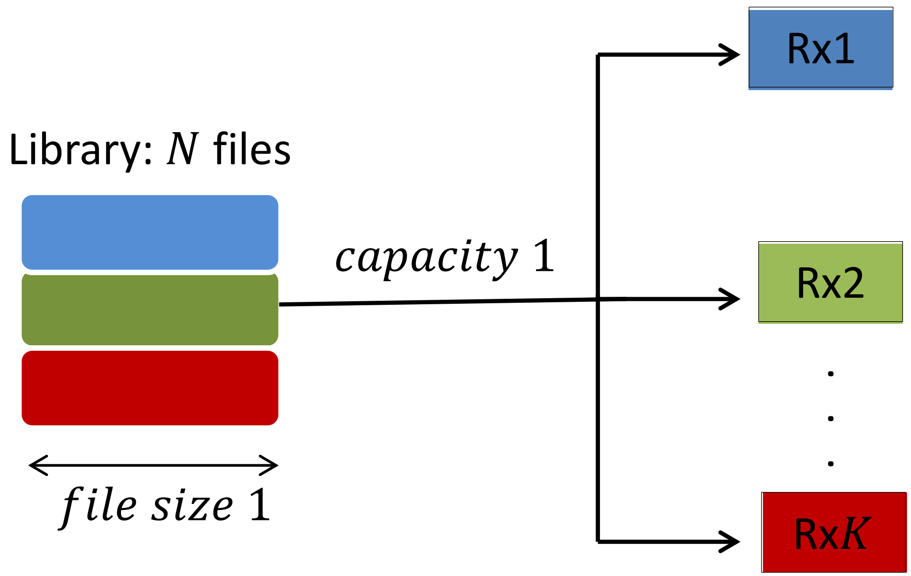


Single-stream Channel: No Caching ($M = 0$)



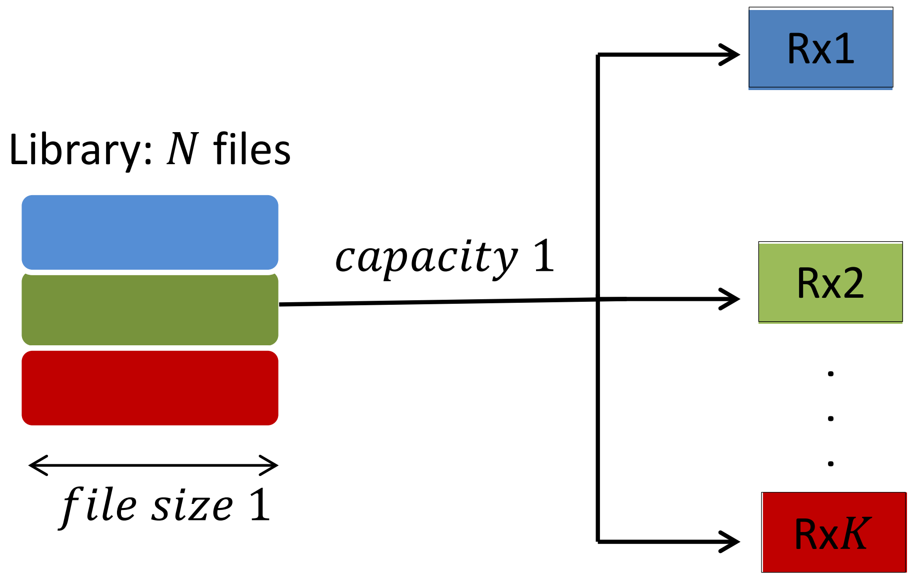
- Transmission sequence:

Single-stream Channel: No Caching ($M = 0$)



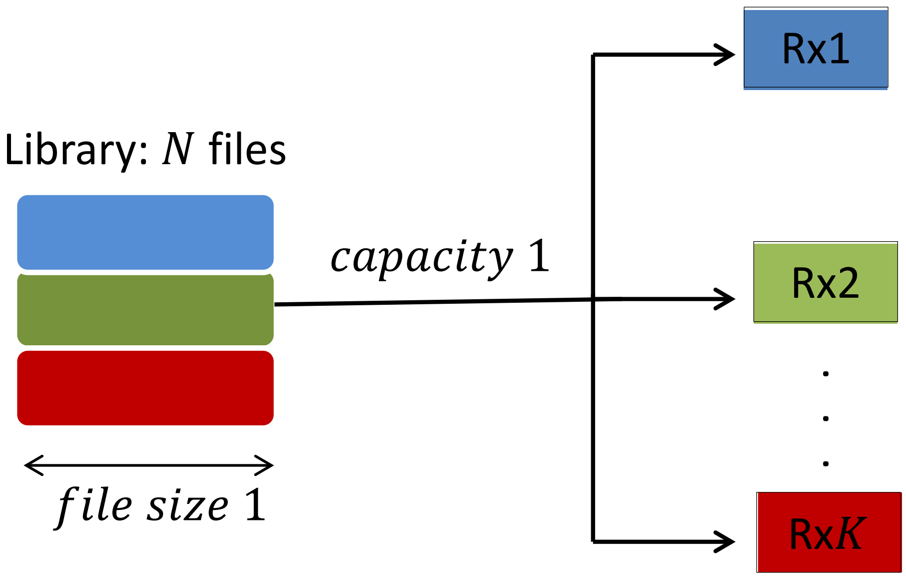
- Transmission sequence: 

Single-stream Channel: No Caching ($M = 0$)



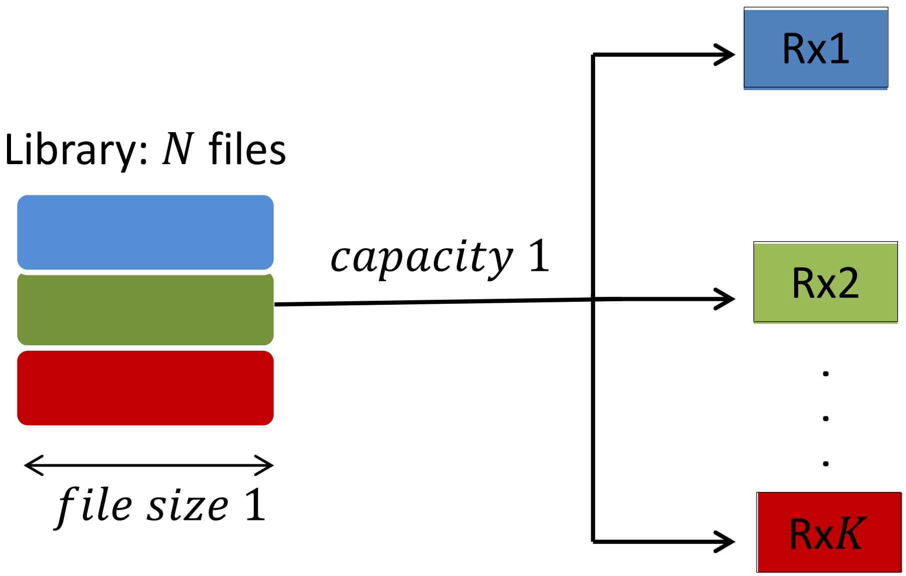
- Transmission sequence:

Single-stream Channel: No Caching ($M = 0$)



- Transmission sequence:

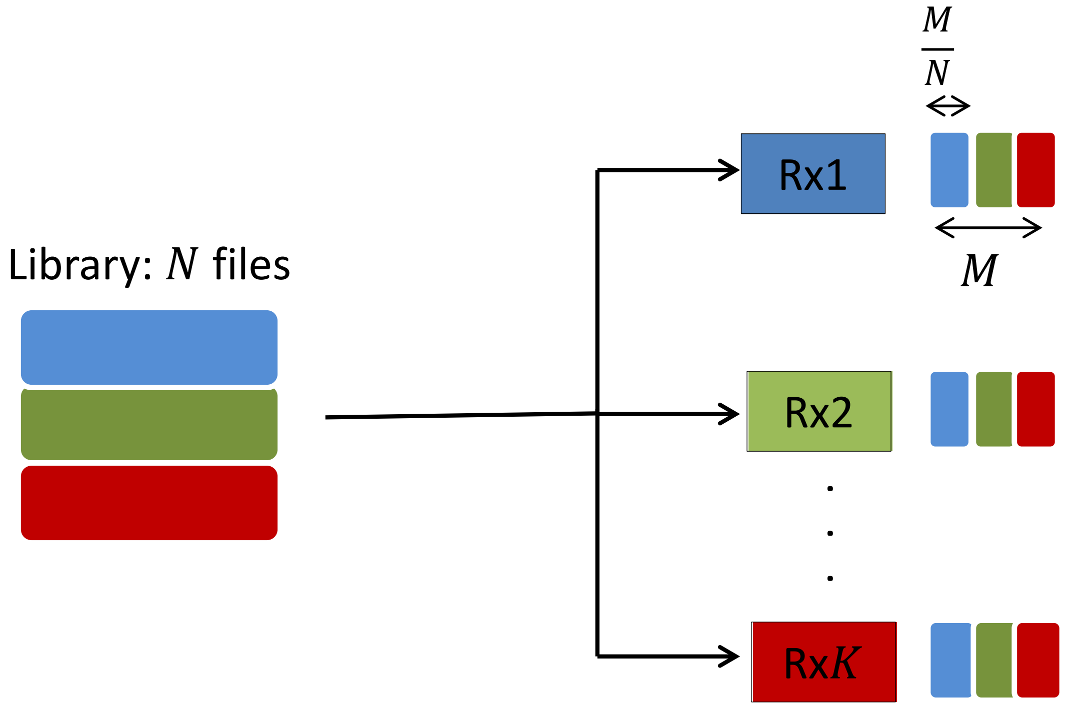
Single-stream Channel: No Caching ($M = 0$)



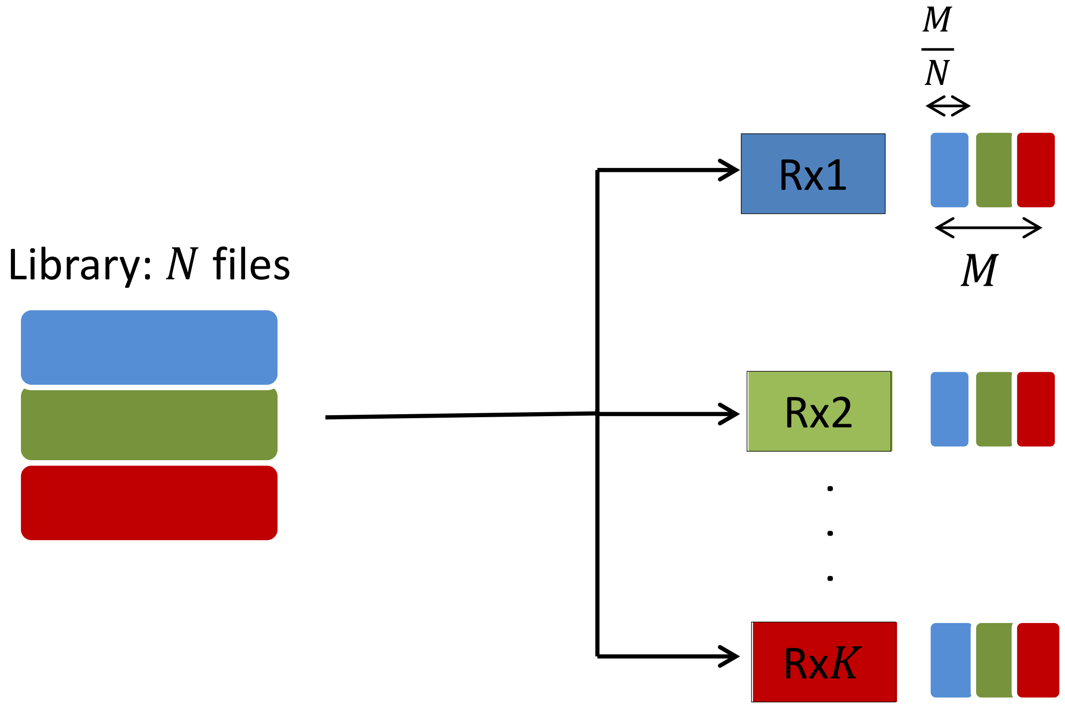
- Transmission sequence:

$$T = K$$

Traditional Caching (Worst-Case Consideration)

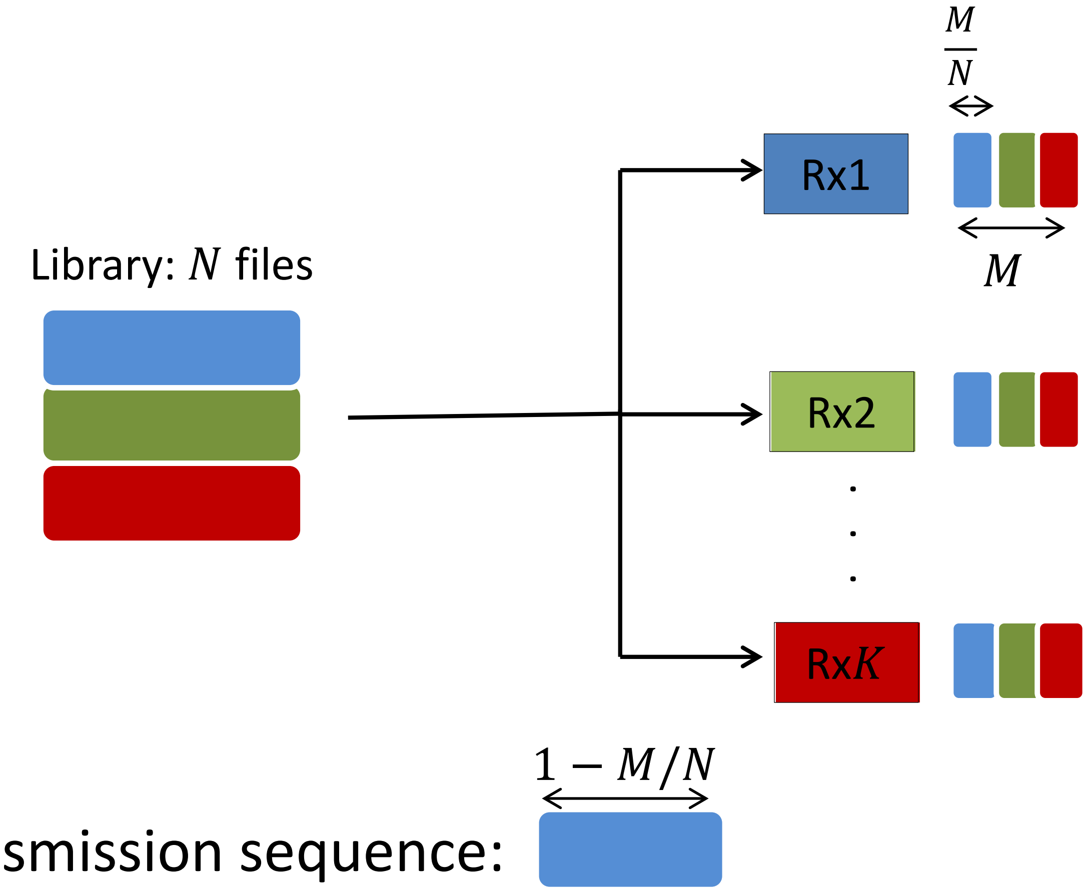


Traditional Caching (Worst-Case Consideration)



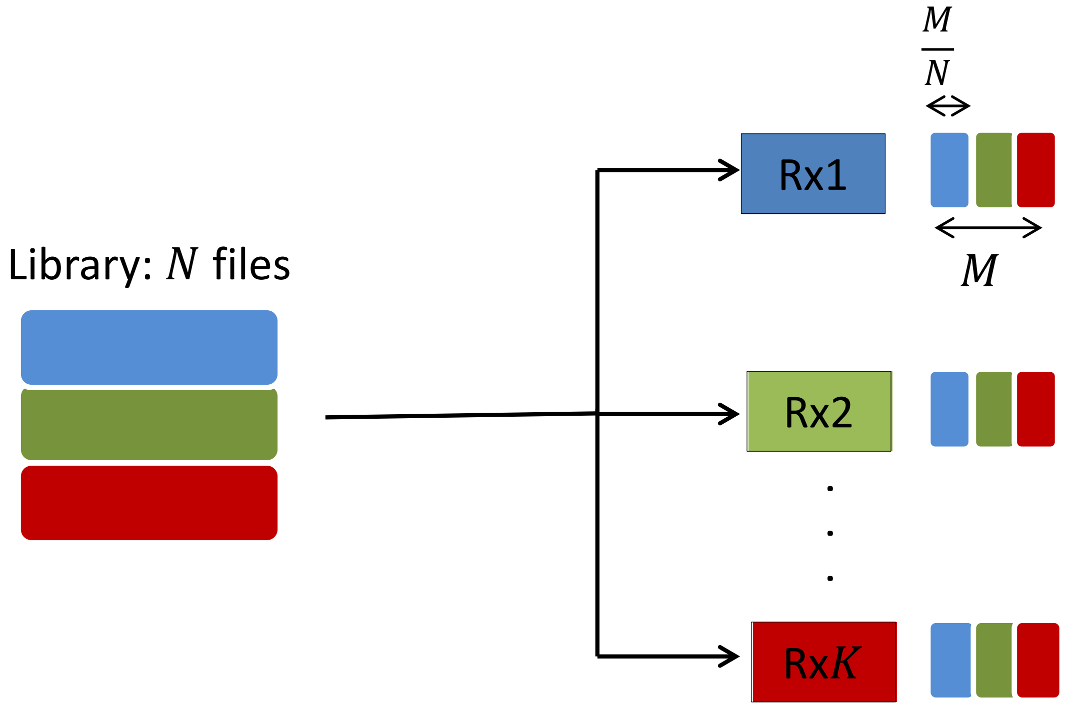
- Transmission sequence:

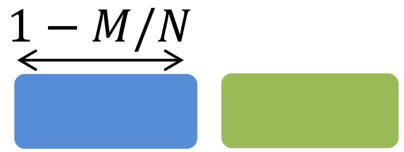
Traditional Caching (Worst-Case Consideration)



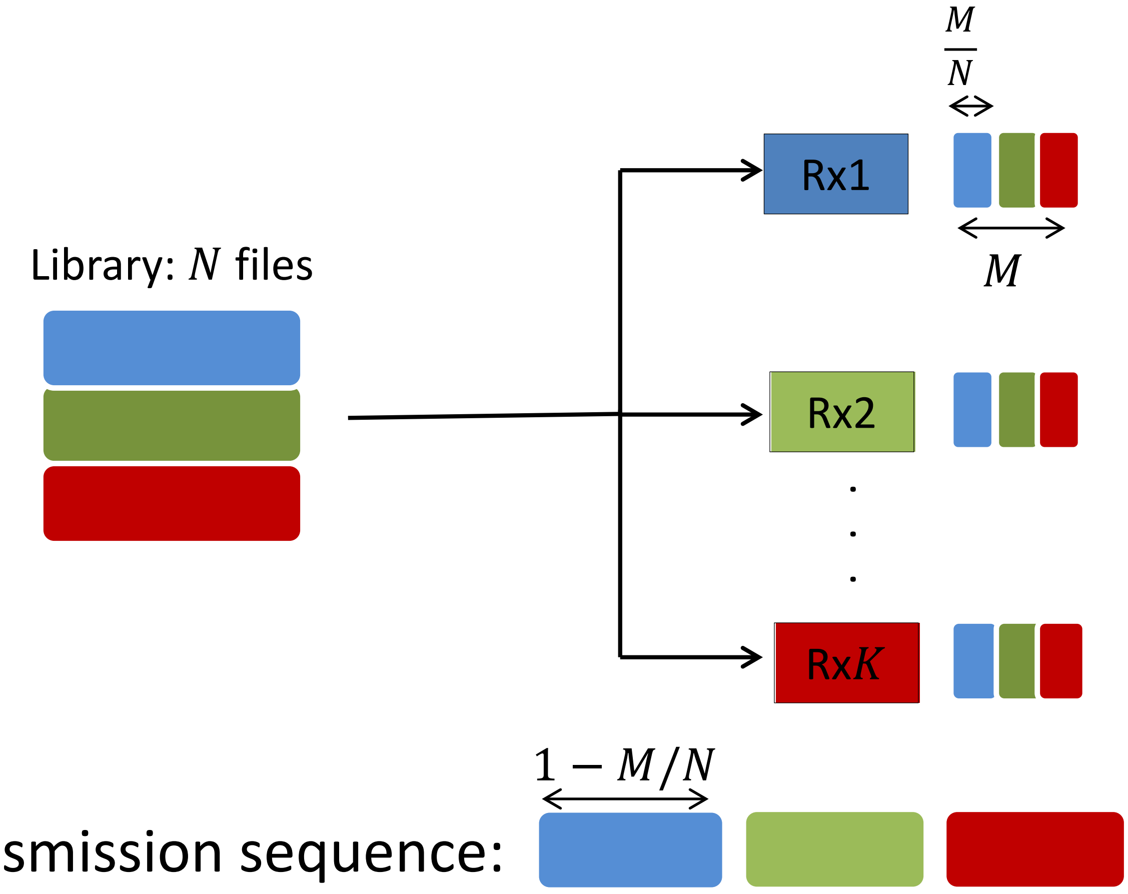
- Transmission sequence:

Traditional Caching (Worst-Case Consideration)



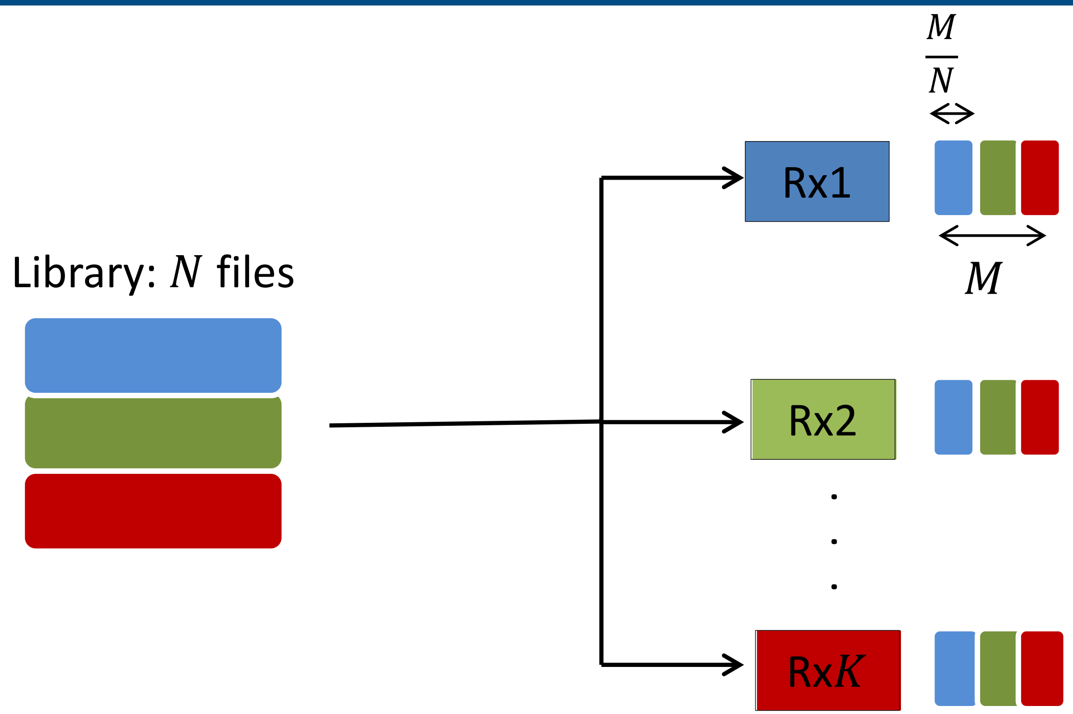
- Transmission sequence: 

Traditional Caching (Worst-Case Consideration)



- Transmission sequence:

Traditional Caching (Worst-Case Consideration)



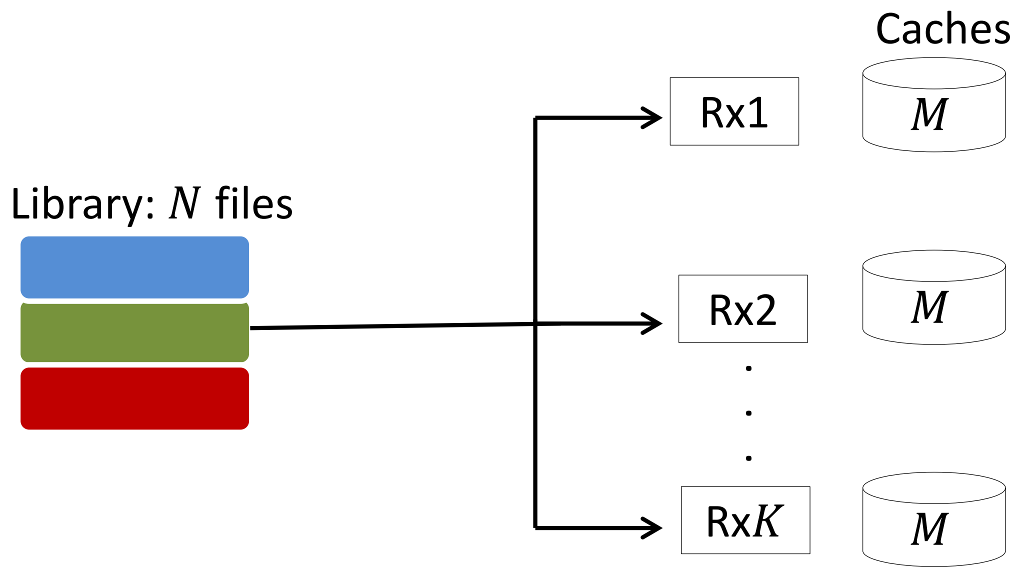
- Transmission sequence:

- Local cache gain: $(1 - M/N)$ for each user
- The rate:

$$T = K(1 - M/N) = K(1 - \gamma), \quad \gamma \overset{\text{def}}{=} \frac{M}{N}$$

The power of advanced caching

Coded Caching

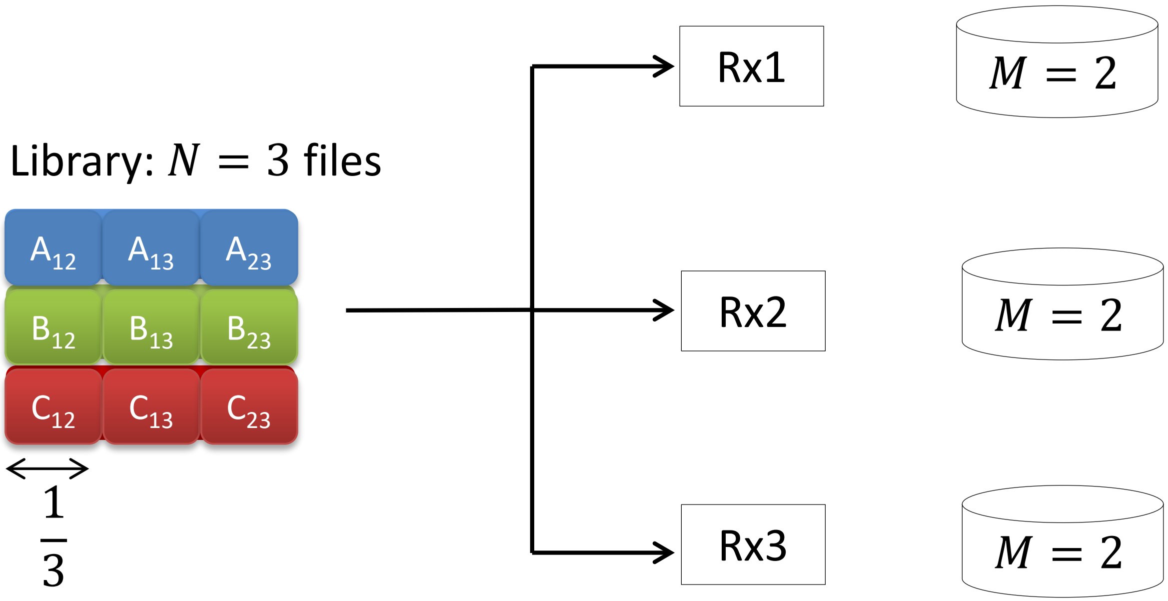


Key breakthrough: USE CACHES TO CANCEL INTERFERENCE

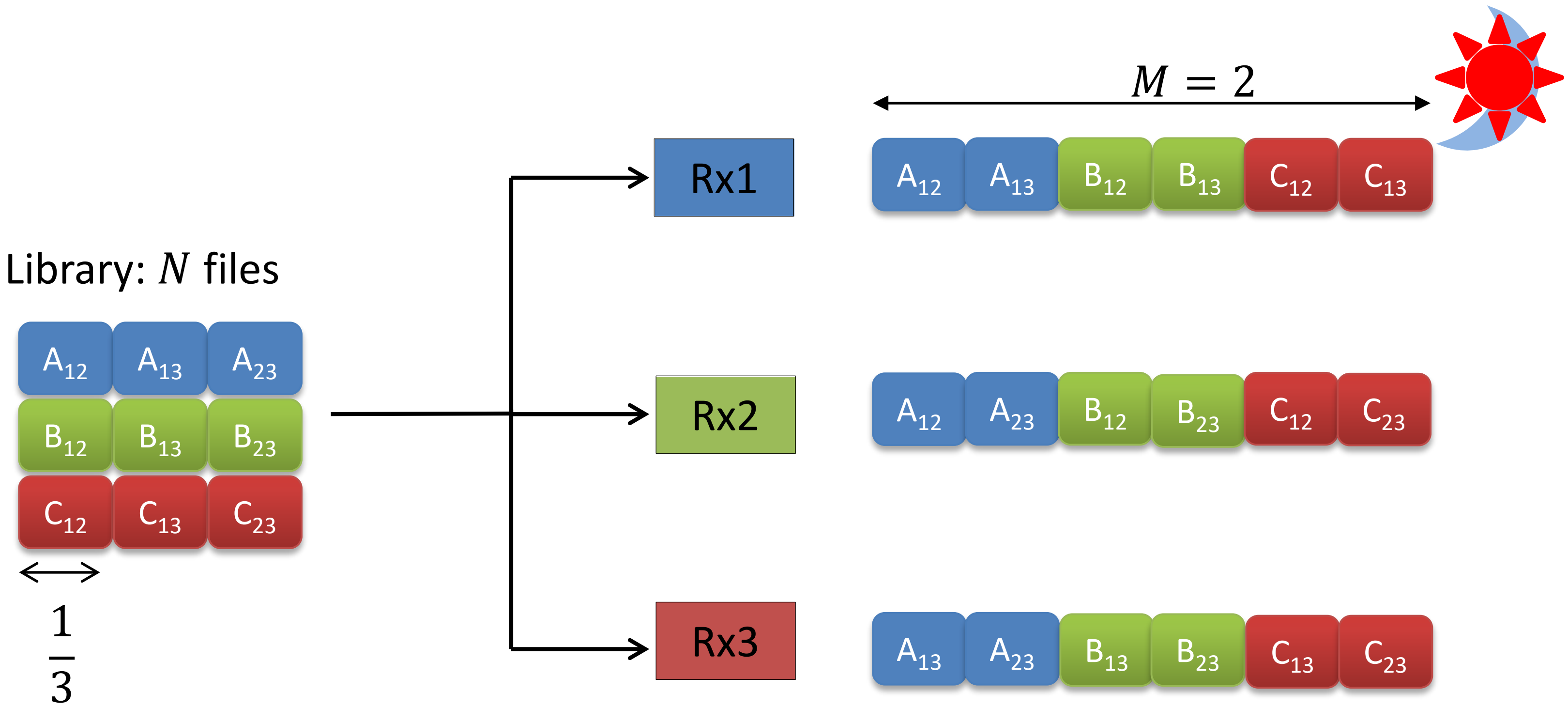
- Cache so that one transmission is useful to many
  - Even if requested files are different
- Large decreases in delay

Result: Maddah-Ali, Niesen (2013)

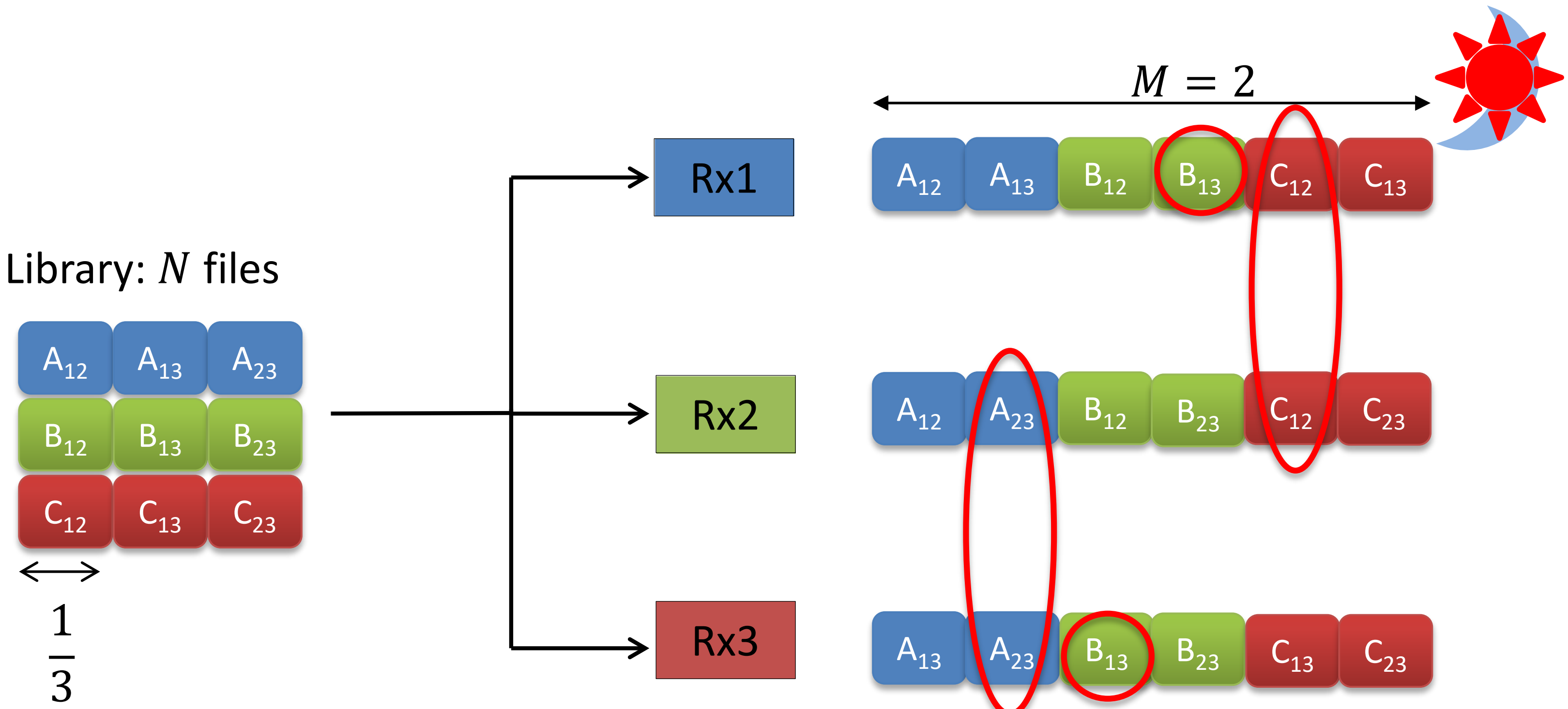
Example: $N = K = 3, M = 2 \quad (\gamma = \frac{2}{3})$



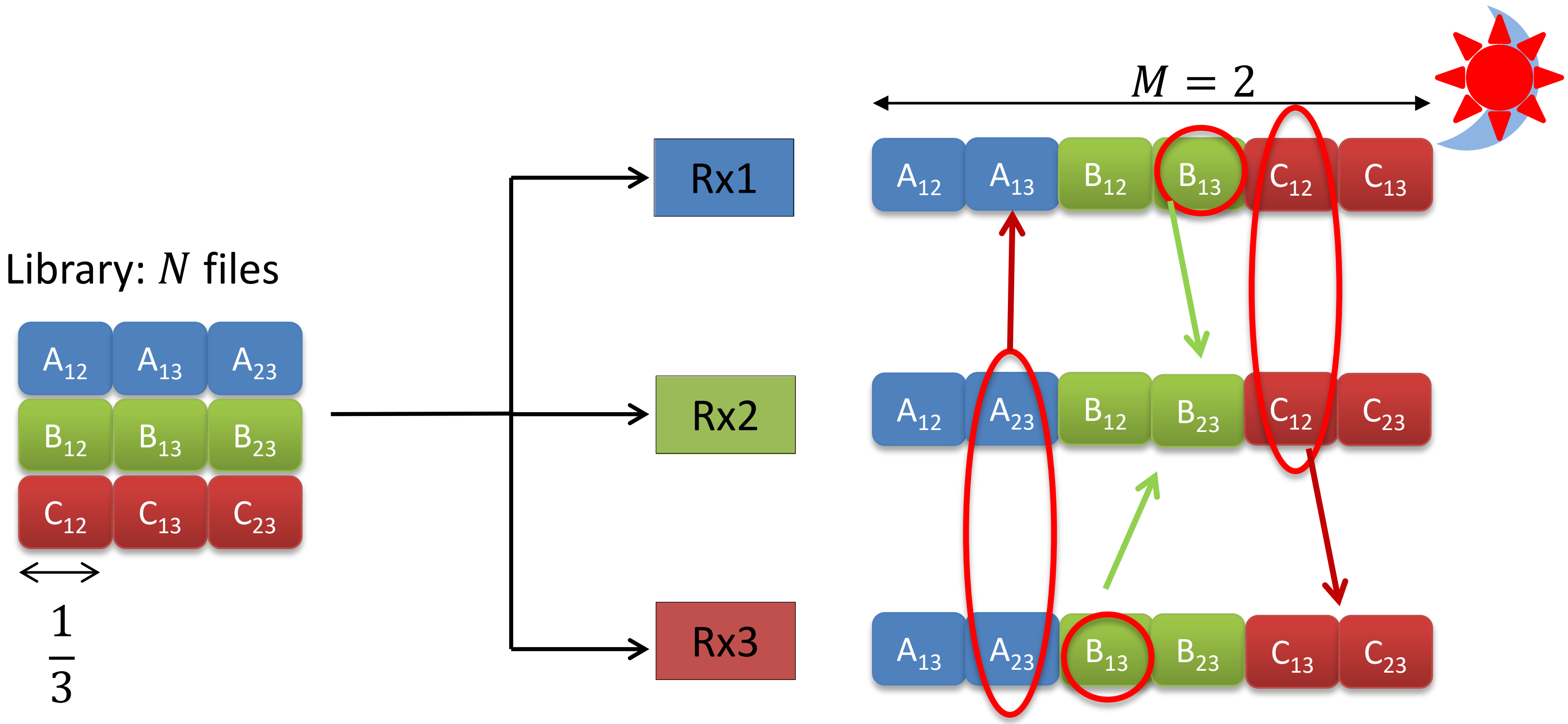
Example: $N = K = 3, M = 2 \quad (\gamma = \frac{M}{N} = \frac{2}{3})$



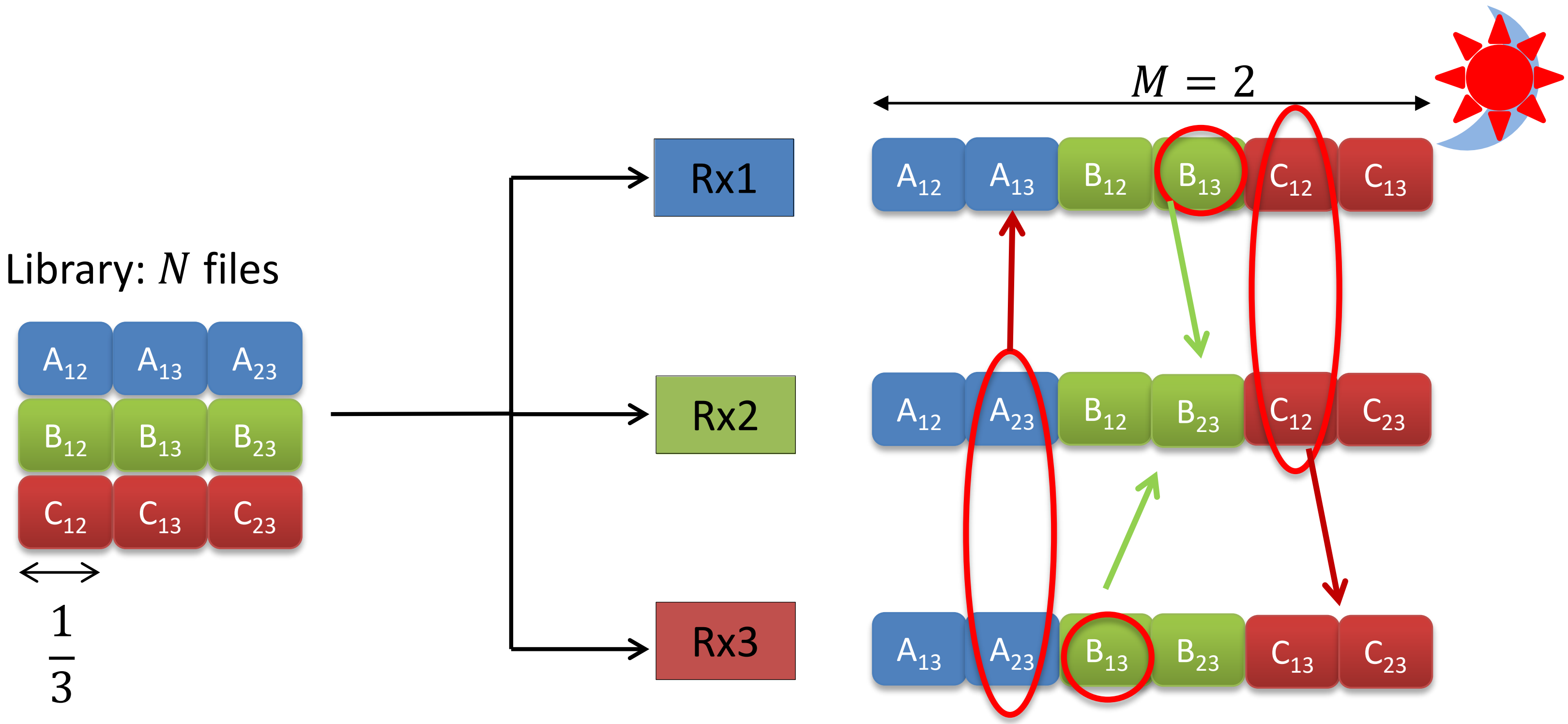
Example: $N = K = 3, M = 2 \quad (\gamma = \frac{M}{N} = \frac{2}{3})$



Example: $N = K = 3, M = 2 \quad (\gamma = \frac{M}{N} = \frac{2}{3})$

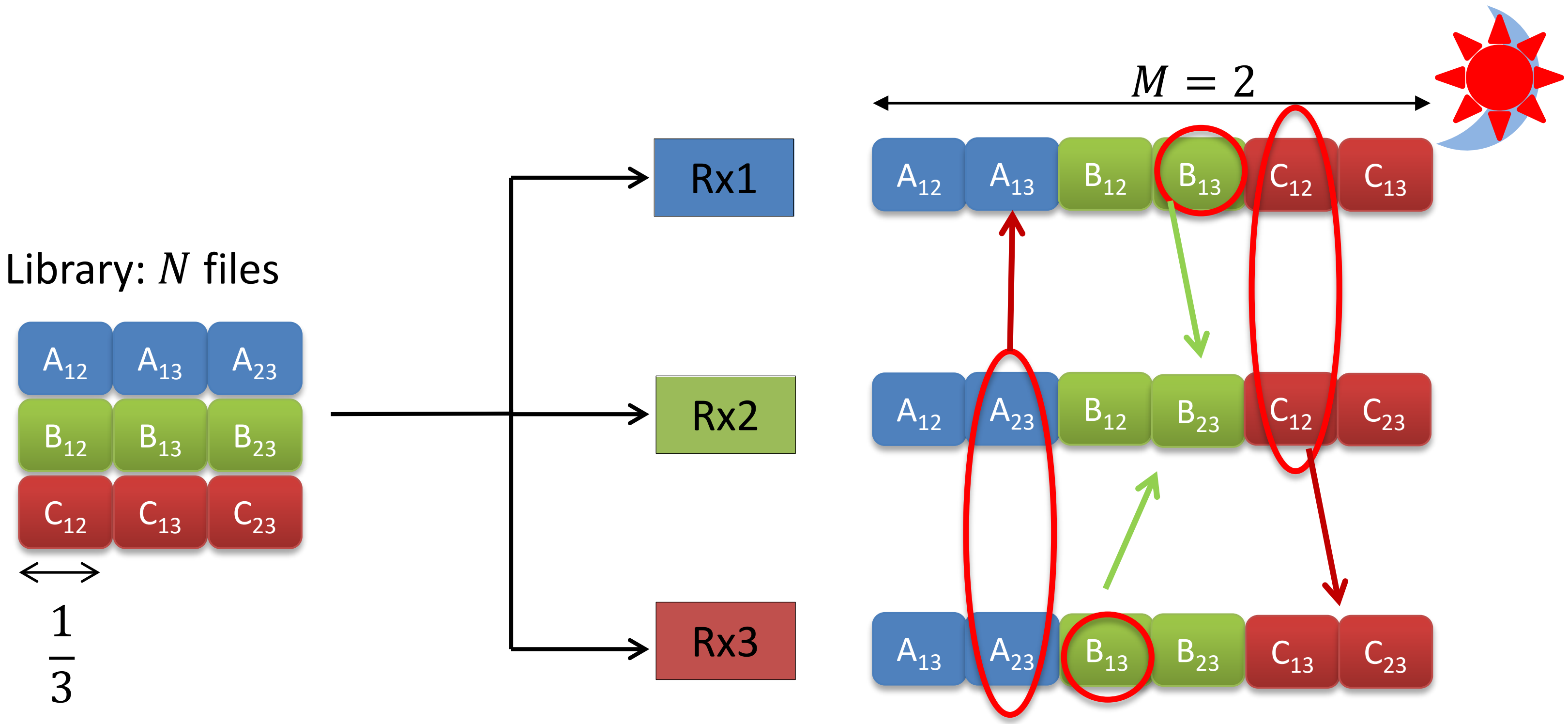


Example: $N = K = 3, M = 2 \quad (\gamma = \frac{M}{N} = \frac{2}{3})$



- Transmit : $A_{23} \oplus B_{13} \oplus C_{12}$ (a common message for all)

Example: $N = K = 3, M = 2 \quad (\gamma = \frac{M}{N} = \frac{2}{3})$



- Transmit : $A_{23} \oplus B_{13} \oplus C_{12}$ (a common message for all)

$$Gain = 3 = \frac{KM}{N} \ users \ at \ a \ time$$

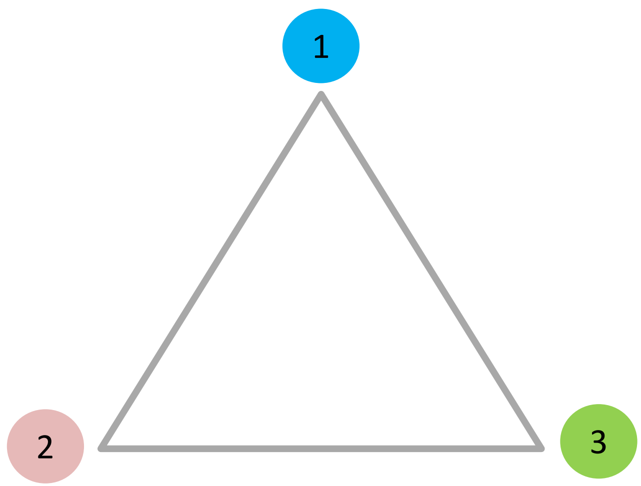
Coded Caching: Intuition - Clique

- *Deliver to $K\gamma + 1$ users at a time*
- *Via XORs with $K\gamma + 1$ subfiles.*
    - ***Each user (out of the $K\gamma + 1$ now served) knows all summands except its own***

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

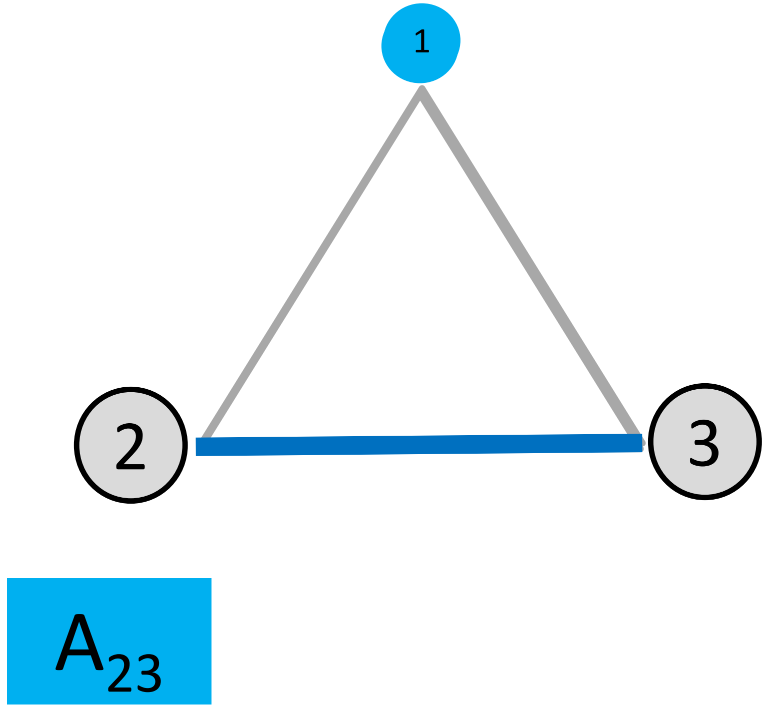
Coded Caching: Intuition - Clique

$$K = 3, K\gamma = 2$$



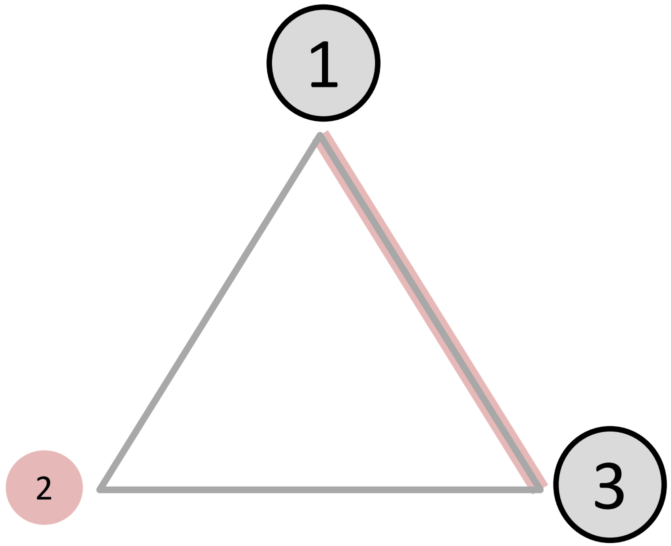
Coded Caching: Intuition - Clique

$$K = 3, K\gamma = 2$$



Coded Caching: Intuition - Clique

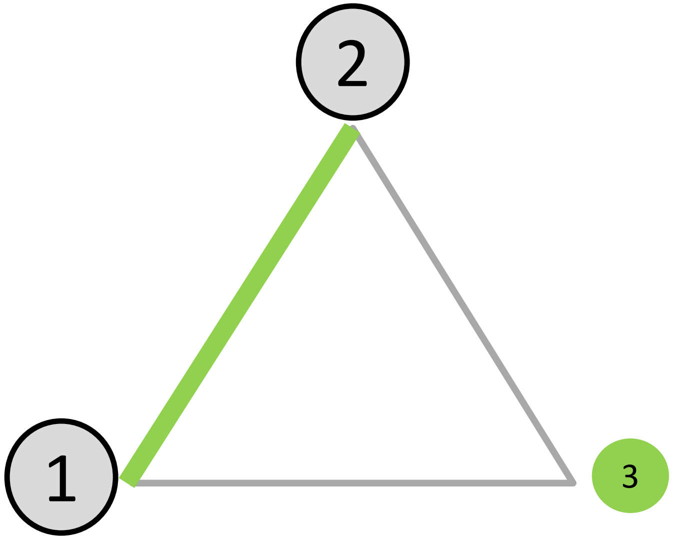
$$K = 3, K\gamma = 2$$



$A_{23} \oplus B_{13}$

Coded Caching: Intuition - Clique

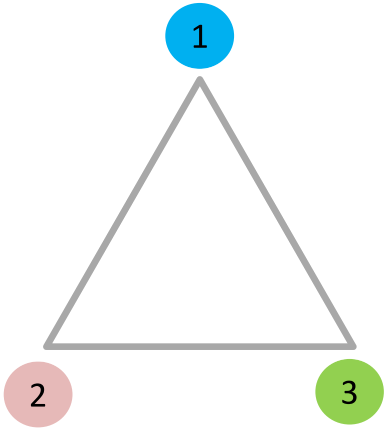
$$K = 3, K\gamma = 2$$



$A_{23} \oplus B_{13} \oplus C_{12}$

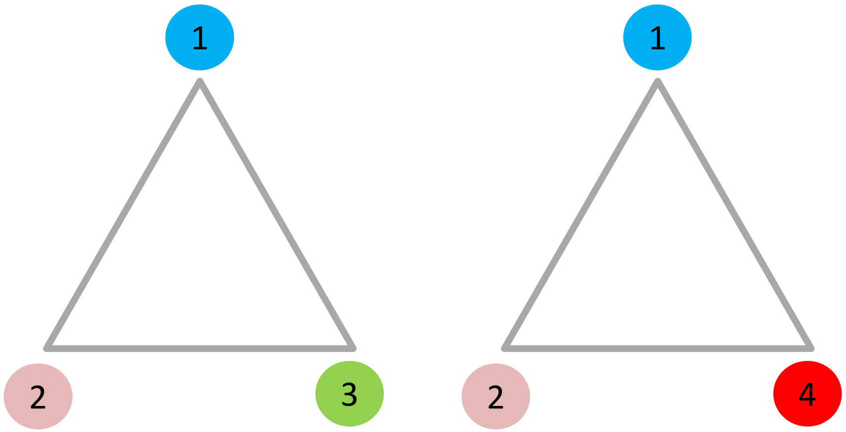
Coded Caching: Intuition - Cliques

$$K = 4, K\gamma = 2$$



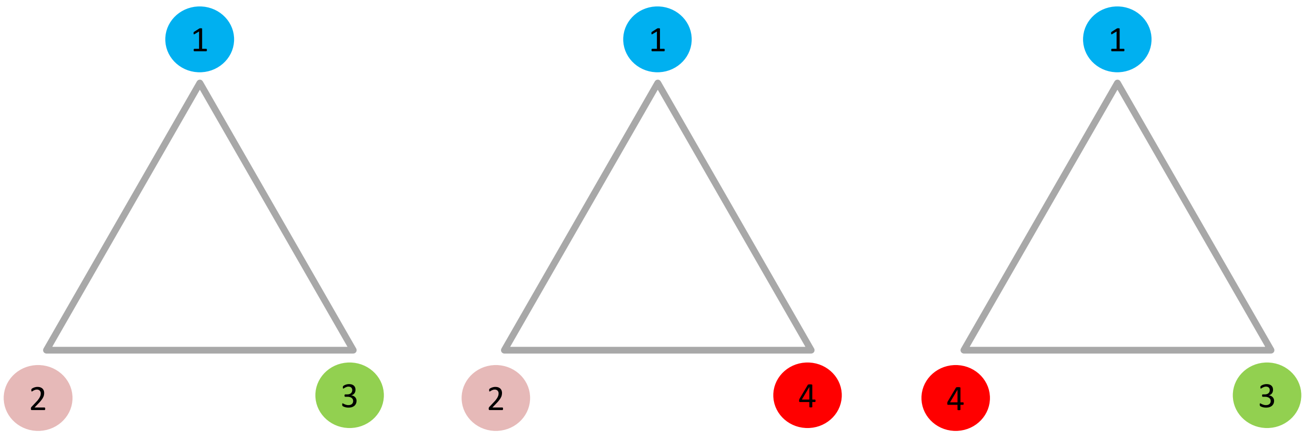
Coded Caching: Intuition - Cliques

$$K = 4, K\gamma = 2$$



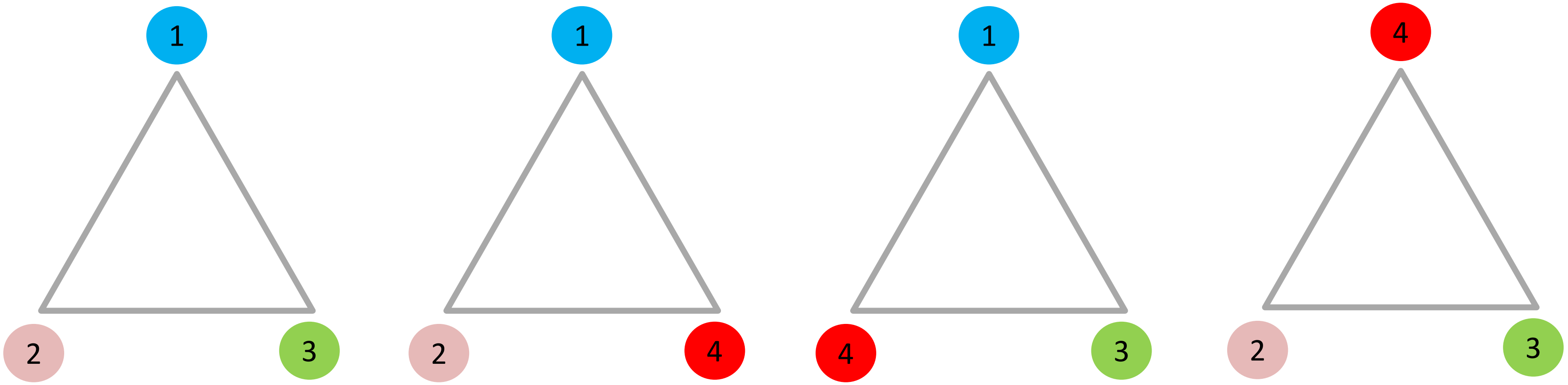
Coded Caching: Intuition - Cliques

$$K = 4, K\gamma = 2$$



Coded Caching: Intuition - Cliques

$$K = 4, K\gamma = 2$$

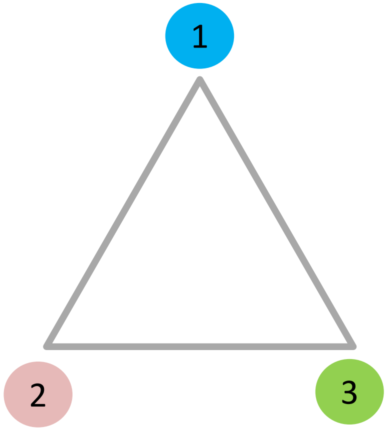


Coded Caching: Intuition - Cliques

$$K = 4, K\gamma = 2$$

Coded Caching: Intuition - Cliques

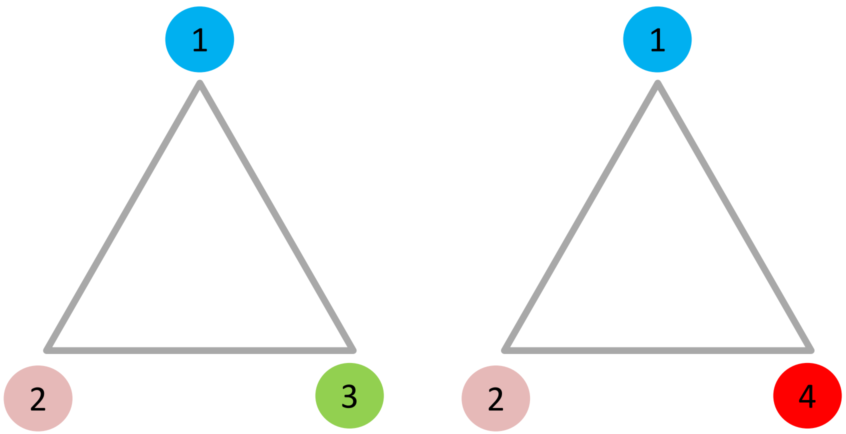
$$K = 4, K\gamma = 2$$



$$A_{23} \oplus B_{13} \oplus C_{12}$$

Coded Caching: Intuition - Cliques

$$K = 4, K\gamma = 2$$

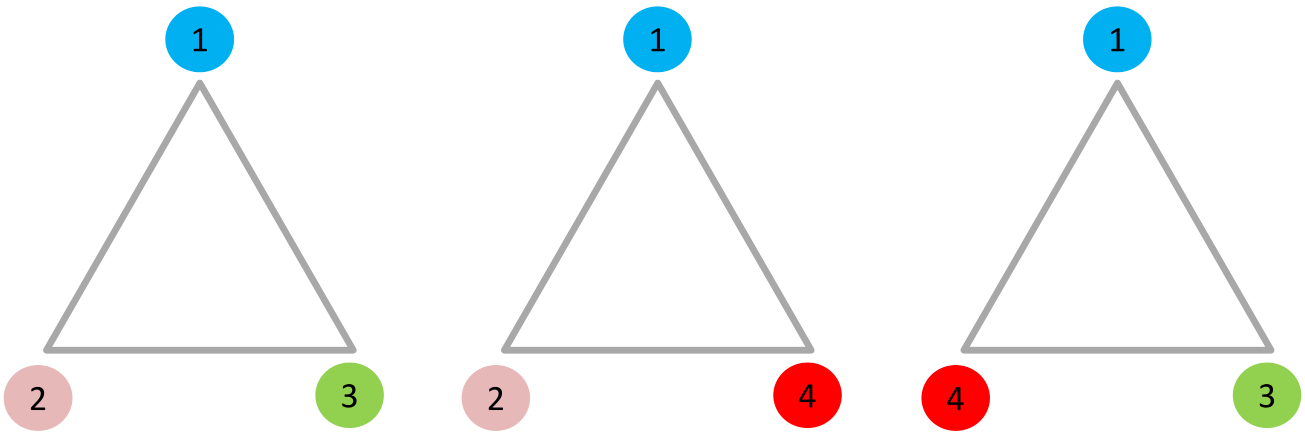


$$A_{23} \oplus B_{13} \oplus C_{12}$$

$$A_{24} \oplus B_{14} \oplus D_{12}$$

Coded Caching: Intuition - Cliques

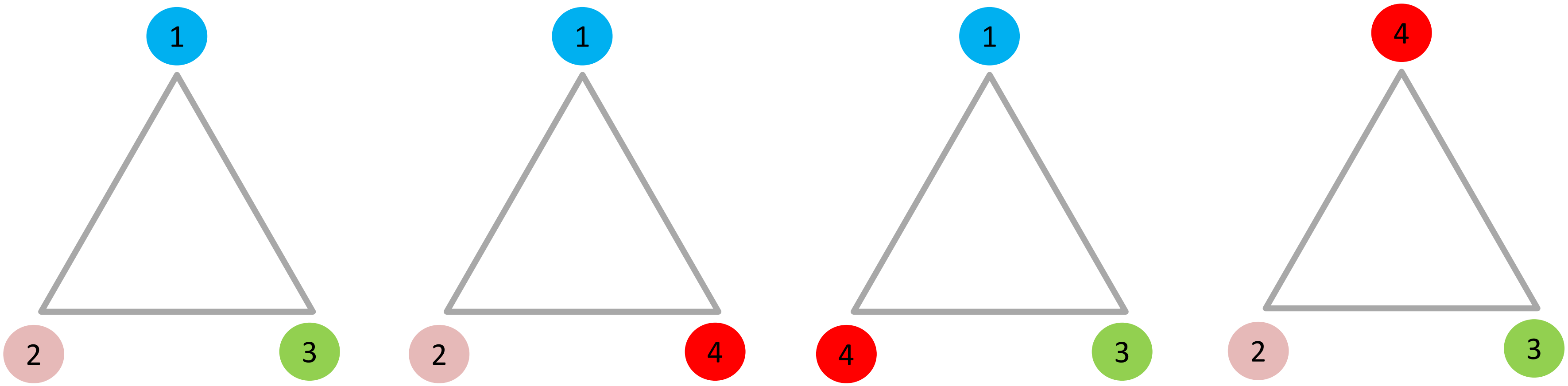
$$K = 4, K\gamma = 2$$



$$A_{23} \oplus B_{13} \oplus C_{12}$$
$$A_{24} \oplus B_{14} \oplus D_{12}$$
$$A_{34} \oplus D_{13} \oplus C_{14}$$

Coded Caching: Intuition - Cliques

$$K = 4, K\gamma = 2$$



$$A_{23} \oplus B_{13} \oplus C_{12}$$

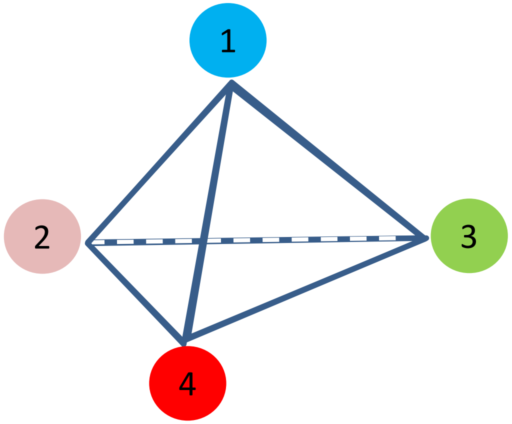
$$A_{24} \oplus B_{14} \oplus D_{12}$$

$$A_{34} \oplus D_{13} \oplus C_{14}$$

$$D_{23} \oplus B_{34} \oplus C_{24}$$

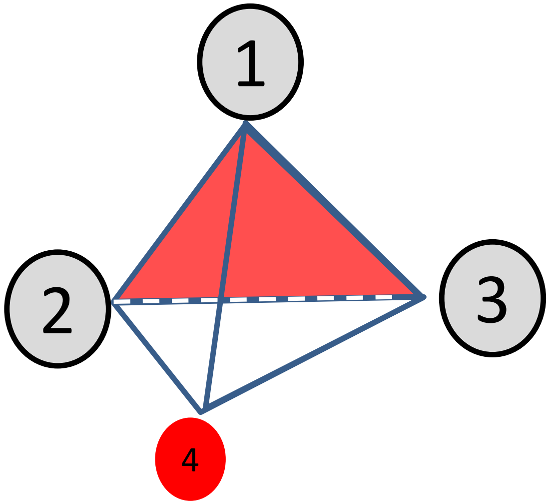
Coded Caching: Intuition - Clique

$$K = 4, K\gamma = 3$$



Coded Caching: Intuition - Clique

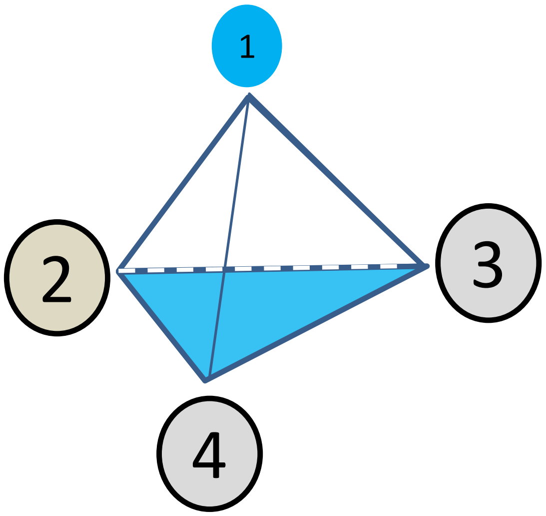
$$K = 4, K\gamma = 3$$



D_{123}

Coded Caching: Intuition - Clique

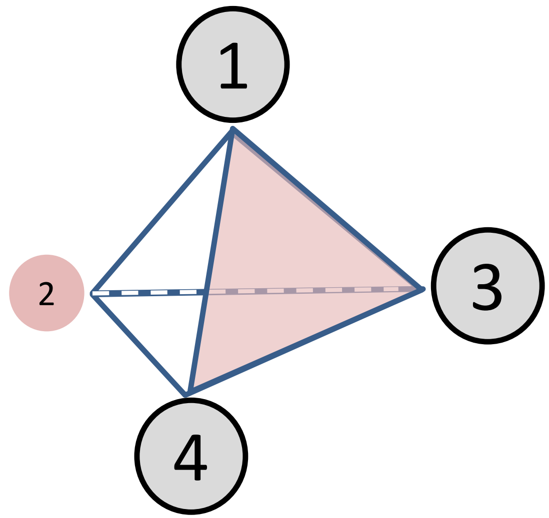
$$K = 4, K\gamma = 3$$



$D_{123} \oplus A_{234}$

Coded Caching: Intuition - Clique

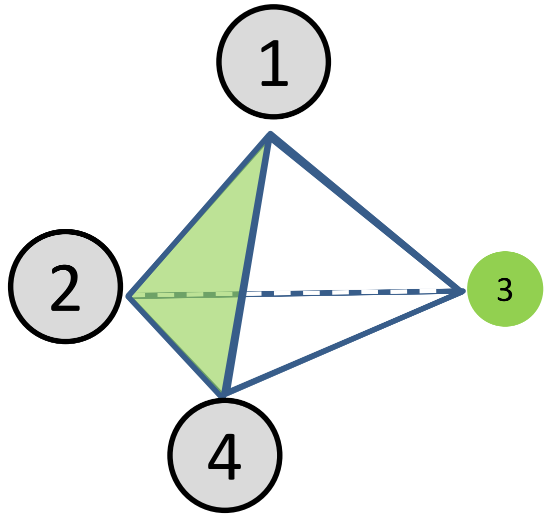
$$K = 4, K\gamma = 3$$



$D_{123} \oplus A_{234} \oplus B_{134}$

Coded Caching: Intuition - Clique

$$K = 4, K\gamma = 3$$



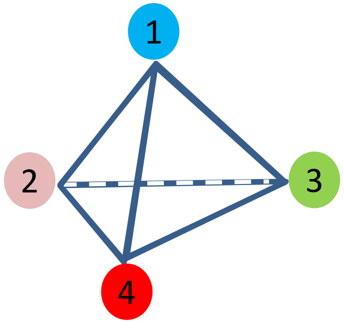
$$D_{123} \oplus A_{234} \oplus B_{134} \oplus C_{124}$$

Coded Caching: Intuition - Cliques

$$K = 5, K\gamma = 3$$

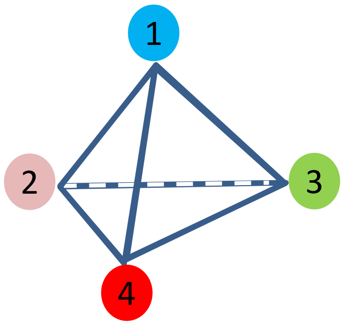
Coded Caching: Intuition - Cliques

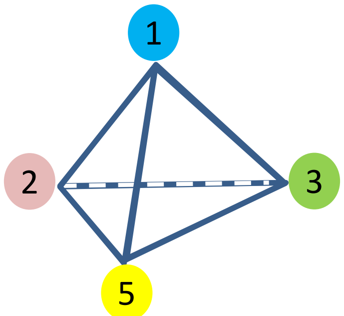
$$K = 5, K\gamma = 3$$



Coded Caching: Intuition - Cliques

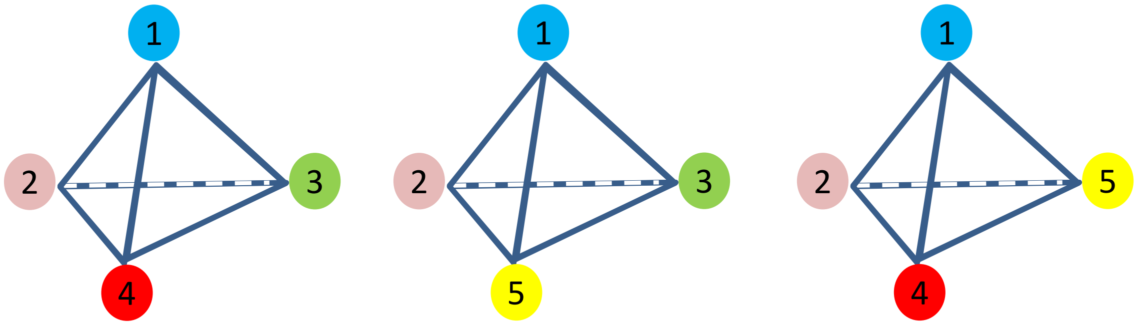
$$K = 5, K\gamma = 3$$





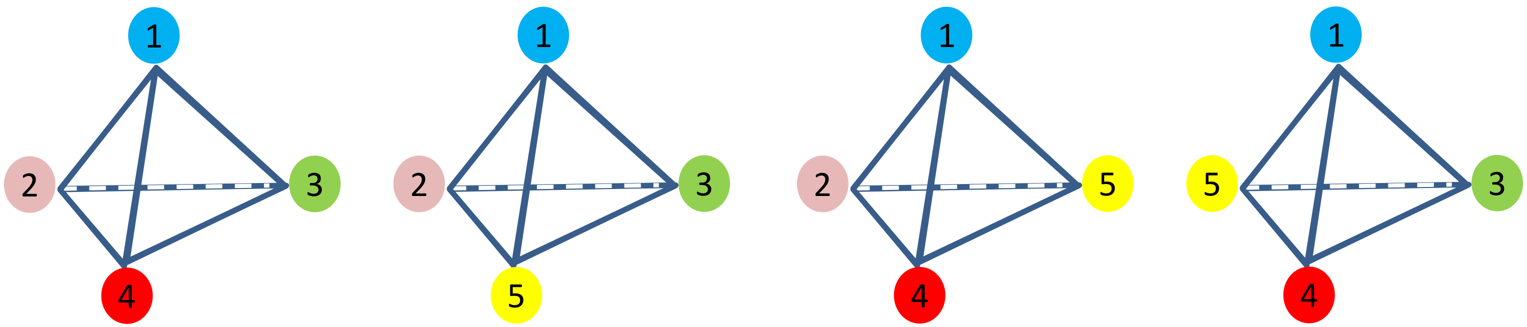
Coded Caching: Intuition - Cliques

$$K = 5, K\gamma = 3$$



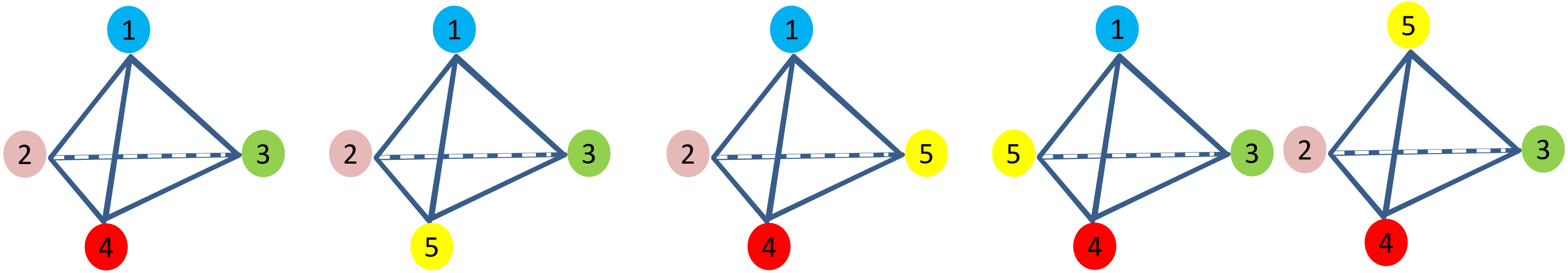
Coded Caching: Intuition - Cliques

$$K = 5, K\gamma = 3$$



Coded Caching: Intuition - Cliques

$$K = 5, K\gamma = 3$$

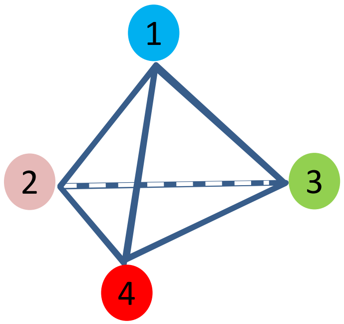


Coded Caching: Intuition - Cliques

$$K = 5, K\gamma = 3$$

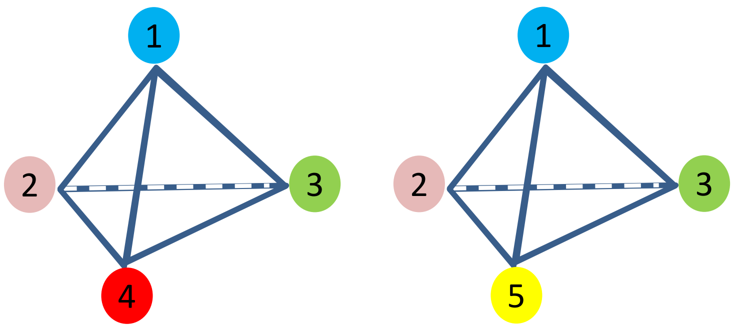
Coded Caching: Intuition - Cliques

$$K = 5, K\gamma = 3$$



$D_{123} \oplus A_{234} \oplus B_{134} \oplus C_{124}$

Coded Caching: Intuition - Cliques

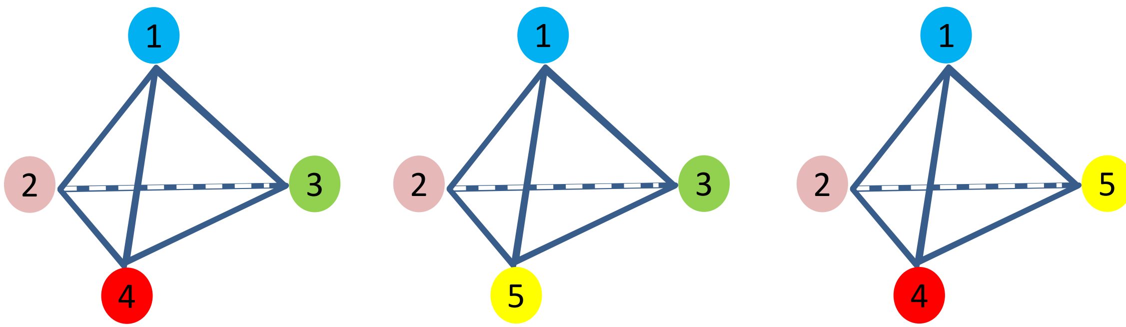


$$D_{123} \oplus A_{234} \oplus B_{134} \oplus C_{124}$$

$$E_{123} \oplus A_{235} \oplus B_{135} \oplus C_{125}$$

Coded Caching: Intuition - Cliques

$$K = 5, K\gamma = 3$$



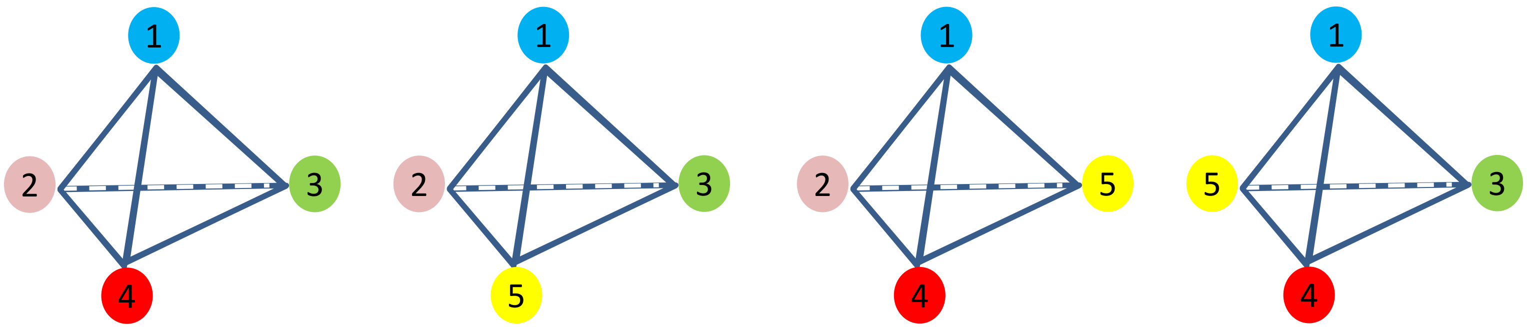
$$D_{123} \oplus A_{234} \oplus B_{134} \oplus C_{124}$$

$$E_{123} \oplus A_{235} \oplus B_{135} \oplus C_{125}$$

$$E_{124} \oplus A_{245} \oplus D_{125} \oplus B_{145}$$

Coded Caching: Intuition - Cliques

$$K = 5, K\gamma = 3$$



$$D_{123} \oplus A_{234} \oplus B_{134} \oplus C_{124}$$

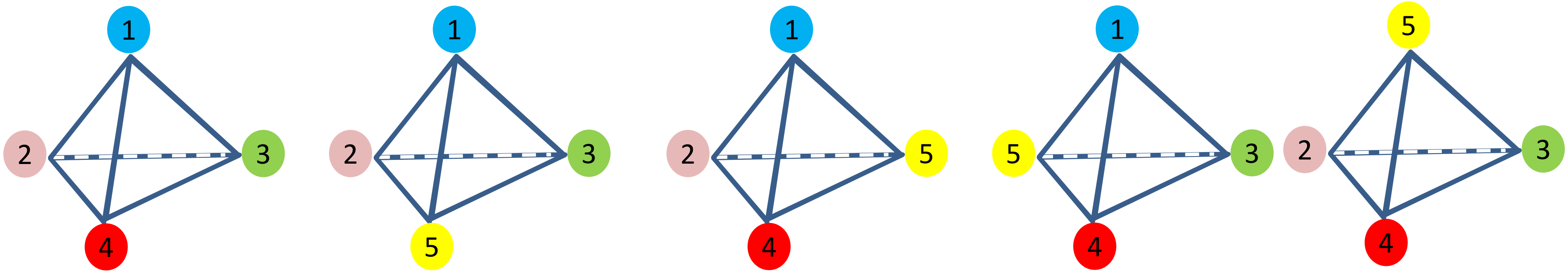
$$E_{123} \oplus A_{235} \oplus B_{135} \oplus C_{125}$$

$$E_{124} \oplus A_{245} \oplus D_{125} \oplus B_{145}$$

$$E_{134} \oplus A_{345} \oplus D_{135} \oplus C_{145}$$

Coded Caching: Intuition - Cliques

$$K = 5, K\gamma = 3$$



$$D_{123} \oplus A_{234} \oplus B_{134} \oplus C_{124}$$

$$E_{123} \oplus A_{235} \oplus B_{135} \oplus C_{125}$$

$$E_{124} \oplus A_{245} \oplus D_{125} \oplus B_{145}$$

$$E_{134} \oplus A_{345} \oplus D_{135} \oplus C_{145}$$

$$E_{234} \oplus B_{345} \oplus D_{235} \oplus C_{245}$$

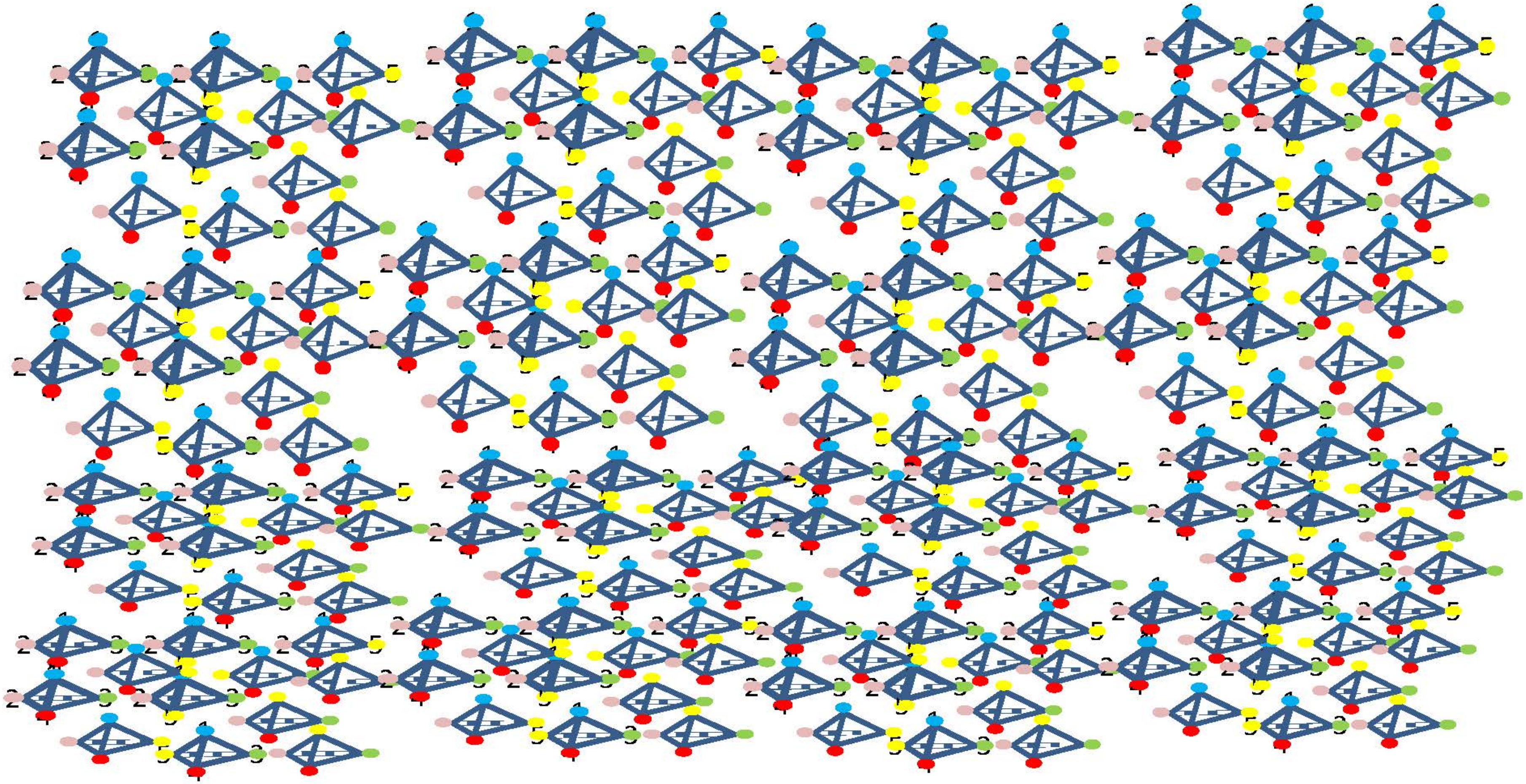
Coded Caching: Intuition – Problematically Many Cliques

$$K = 100, \qquad K\gamma = 9$$

Coded Caching: Intuition – Problematically Many Cliques

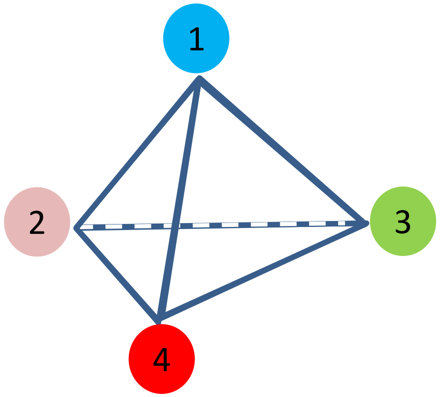
$K = 100, \qquad K\gamma = 9$

$9 \; users \; at \; a \; time \; , \qquad 2 \cdot 10^{13} \;\; cliques$



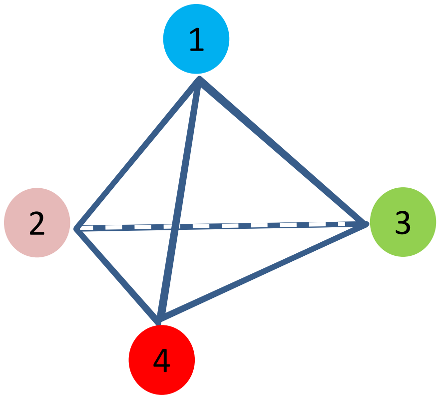
Extension of Clique-based Idea

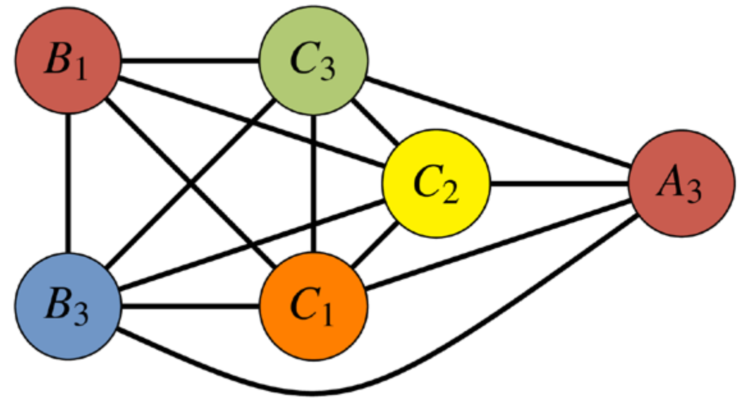
Extension of Clique-based Idea



$D_{123} \oplus A_{234} \oplus B_{134} \oplus C_{124}$

Extension of Clique-based Idea

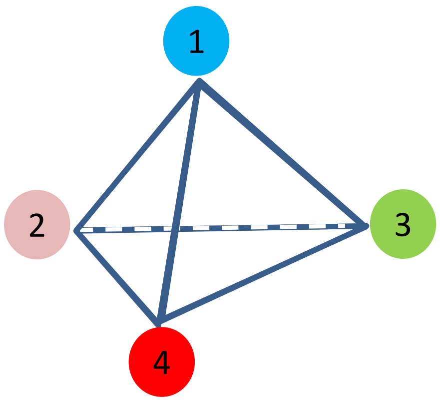


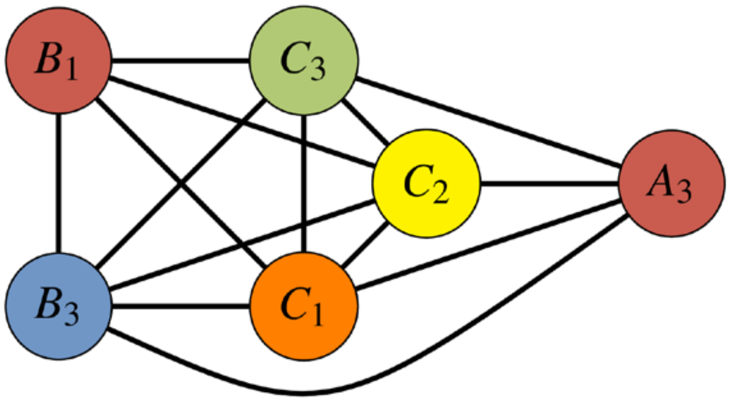


$D_{123} \oplus A_{234} \oplus B_{134} \oplus C_{124}$

$A_3 \oplus B_1$ B_3 C_1 C_2 C_3

Extension of Clique-based Idea





$D_{123} \oplus A_{234} \oplus B_{134} \oplus C_{124}$

$A_3 \oplus B_1$ B_3 C_1 C_2 C_3

Variety of settings:

- Non-uniform demands
- Decentralized placement

Ji, Tulino, Llorca, Caire (Trans IT 2017)

Example: Non-uniform File Popularity (random placement)

Example: Non-uniform File Popularity (random placement)

- 3 files $\{A, B, C\}$. A very popular, File B popular, File C not popular

Example: Non-uniform File Popularity (random placement)

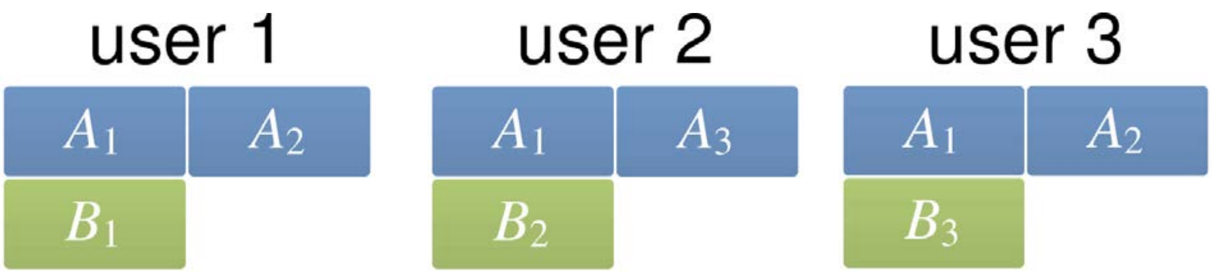
- 3 files $\{A, B, C\}$. A very popular, File B popular, File C not popular
- Split each file into 3 parts each. E.g. $A = \{A_1, A_2, A_3\}$

Example: Non-uniform File Popularity (random placement)

- 3 files $\{A, B, C\}$. A very popular, File B popular, File C not popular
- Split each file into 3 parts each. E.g. $A = \{A_1, A_2, A_3\}$
- Cache distribution $\boldsymbol{p} = \{A = \frac{2}{3}, B = \frac{1}{3}, C = 0\}$

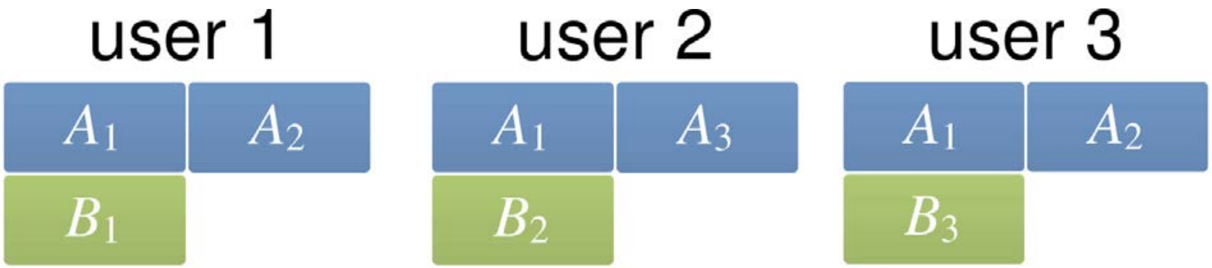
Example: Non-uniform File Popularity (random placement)

Cache realization $\mathcal{C}$



Example: Non-uniform File Popularity (random placement)

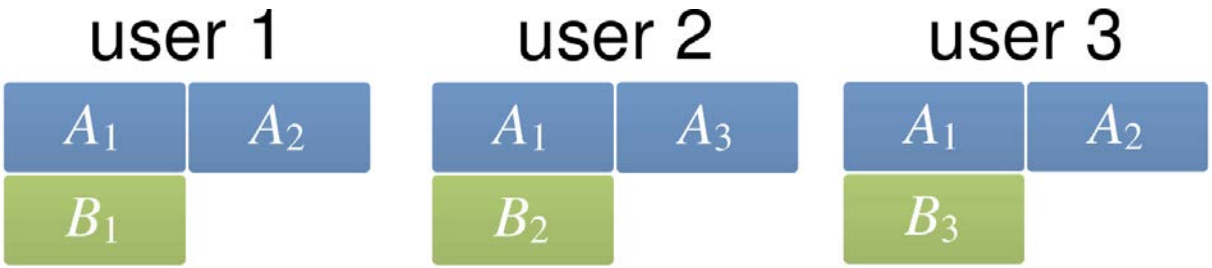
Cache realization $\mathcal{C}$



Request: user1→ A, user2→ B, user3→ C

Example: Non-uniform File Popularity (random placement)

Cache realization $\mathcal{C}$



Request: user1→ A, user2→ B, user3→ C

Needed subfiles : $\mathcal{Q} = \{A_3, B_1, B_3, C_1, C_2, C_3\}$

Non-uniform Example: Conflict Graph

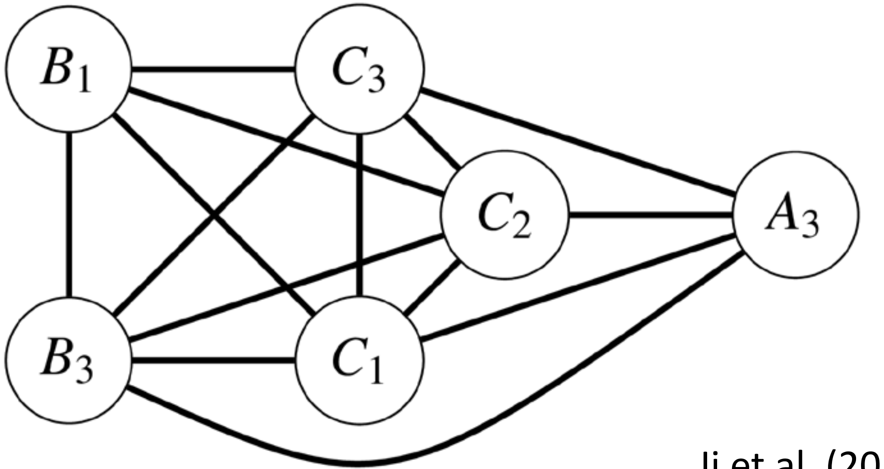
Vertex for each requested subpart:

- Replicate if multiple requests of a subfile

Edge if

- Not same identity (cannot connect subfile to itself)
- Request(er) not among users caching the other vertex
  - see (A_3, B_1)

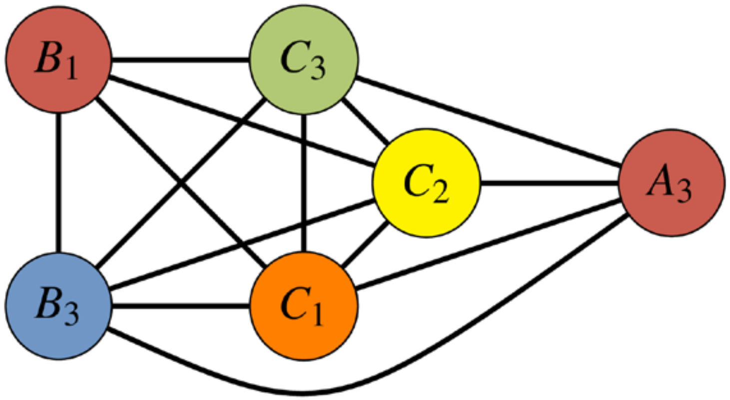
Requests: user1→ A, user2→ B, user3→ C
Queried parts: $Q = \{A_3, B_1, B_3, C_1, C_2, C_3\}$



Ji et al. (2017)

Non-uniform Example: Graph Coloring for XORs

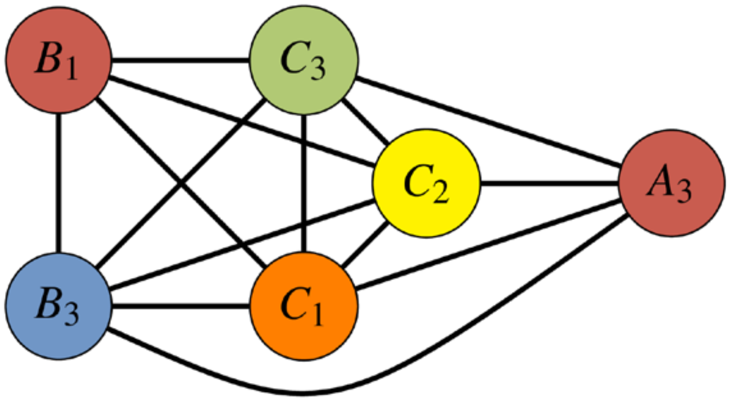
Connected vertices must have different colors



Ji et al. (2017)

Non-uniform Example: Graph Coloring for XORs

Connected vertices must have different colors



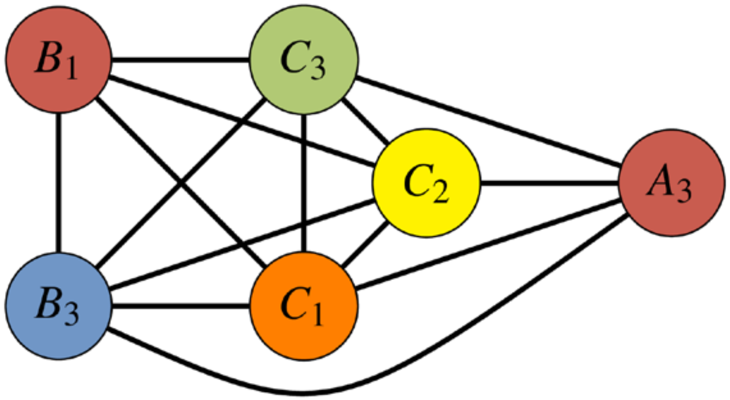
Transmission

$A_3 \oplus B_1$ B_3 C_1 C_2 C_3

Non-uniform Example: Graph Coloring for XORs

Connected vertices must have different colors

Transmission



$$\boxed{T(\gamma) = 5/3}$$

Gain

$$\frac{|Q|}{\chi(H_{C,Q})} \quad (\chi \text{ is chromatic number})$$

Calculation:

$$\frac{|Q|}{\chi(H_{C,Q})} = \frac{K(1-\gamma)}{T}$$

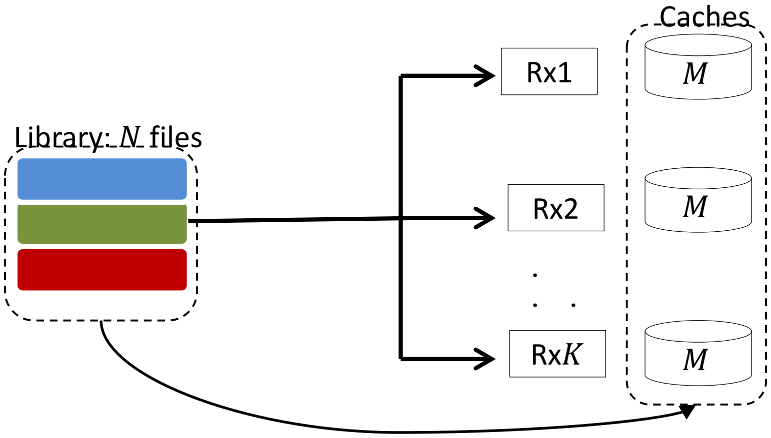
$$= \frac{3\left(1 - \frac{1}{3}\right)}{5/3} = \frac{6}{5}$$

The Optimization Problem:

Optimize how you cache and send each bit

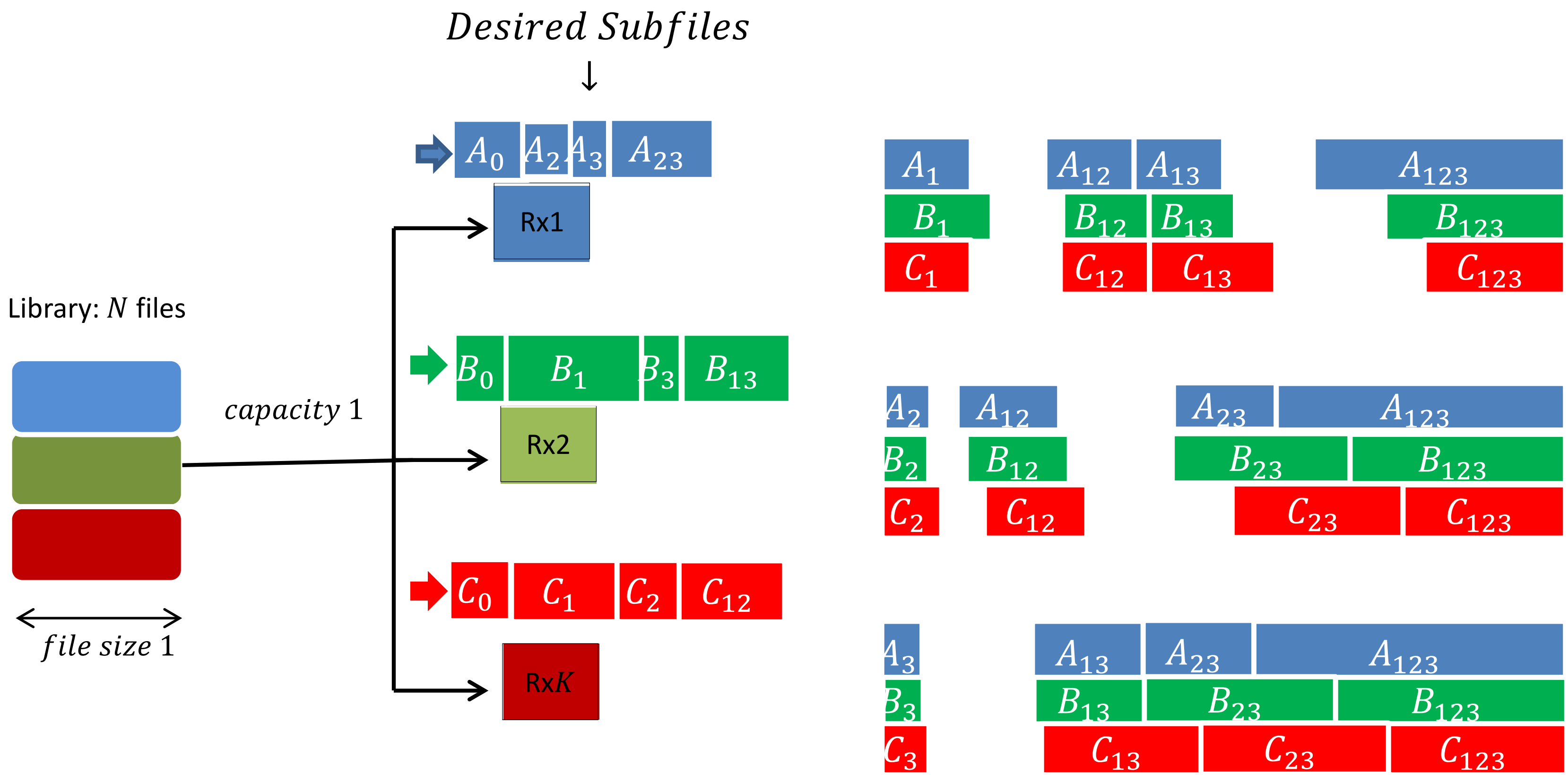
Another (ongoing) Success of Information Theory

The Optimization Problem (Caching Placement Part)



$$T^*(M) = \min_{schemes\ \chi} \max_{demands\ \mathbf{d}} T(\mathbf{d}, \chi, M)$$

The Optimization Problem (Delivery Part)



Transmission Scheme F $\rightarrow$ $F_1(A_{P_1}, B_{Q_1}, C_{R_1})$ $F_2(A_{P_2}, B_{Q_2}, C_{R_2})$ ….. $F_{T_F}(A_{P_T}, B_{Q_T}, C_{R_T})$

Scheme Φ $\rightarrow$ $\Phi_1(A_{P_1}, B_{Q_1}, C_{R_1})$ $\Phi_2(A_{P_2}, B_{Q_2}, C_{R_2})$ ….. $\Phi_{T_\Phi}(A_{P_T}, B_{Q_T}, C_{R_T})$

Scheme Ж $\rightarrow$ $Ж_1(A_{P_1}, B_{Q_1}, C_{R_1})$ $Ж_2(A_{P_2}, B_{Q_2}, C_{R_2})$ ….. $Ж_{T_Ж}(A_{P_T}, B_{Q_T}, C_{R_T})$

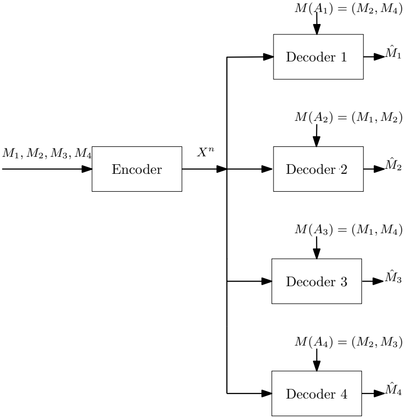
⋮

Tool: Coded Caching $\rightarrow$ Index Coding $\rightarrow$ Graphs

- This is an index coding problem

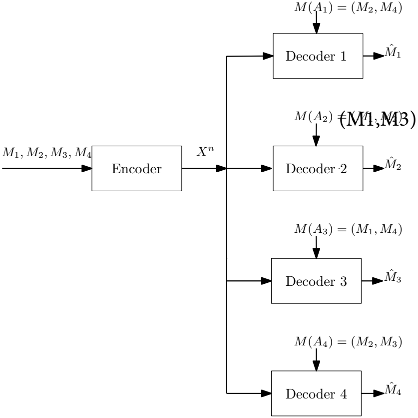
Tool: Coded Caching → Index Coding → Graphs

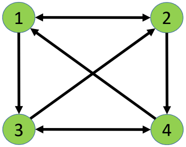
- This is an index coding problem
- Example: 4-message index coding problem



Tool: Coded Caching → Index Coding → Graphs

- This is an index coding problem
- Example: 4-message index coding problem





Tool: Maximum Acyclic Subgraph Bound

Theorem

If G is acyclic, then optimal delay is

$$T^*(G) = |G|.$$

[1] Arbabjolfaei et al., 2013.

Tool: Maximum Acyclic Subgraph Bound

Theorem

If G is acyclic, then optimal delay is

$$T^*(G) = |G|.$$

Figure: As if no side information (i.e., TDMA is best you can do)

[1] Arbabjolfaei et al., 2013.

Tool: Maximum Acyclic Subgraph Bound

Theorem

If G is acyclic, then optimal delay is

$$T^*(G) = |G|.$$

Figure: As if no side information (i.e., TDMA is best you can do)

Theorem

For any problem corresponding to an arbitrary G, then

$$T \geq \sum_{j \in J} |M_j|$$

for all <u>*acyclic*</u> *subgraphs $J \subset G$.*

1

[1] Arbabjolfaei et al., 2013.

Step: Create Graph for One Caching Problem

- Get $\mathbf{d}$ (e.g. $\mathbf{d} = (1, 2, 3)$)

Step: Create Graph for One Caching Problem

- Get $\mathbf{d}$ (e.g. $\mathbf{d} = (1, 2, 3)$)
- General split F_i to $F_{i,\mathcal{W}}$
  - $\mathcal{W} \in 2^{[3]} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$

Step: Create Graph for One Caching Problem

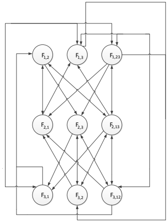
- Get $\mathbf{d}$ (e.g. $\mathbf{d} = (1, 2, 3)$)
- General split F_i to $F_{i,\mathcal{W}}$
  - $\mathcal{W} \in 2^{[3]} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$
- Each node $=$ desired sub-file

Step: Create Graph for One Caching Problem

- Get $\mathbf{d}$ (e.g. $\mathbf{d} = (1, 2, 3)$)
- General split F_i to $F_{i,\mathcal{W}}$
  - $\mathcal{W} \in 2^{[3]} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$
- Each node = desired sub-file
- Row j for user j

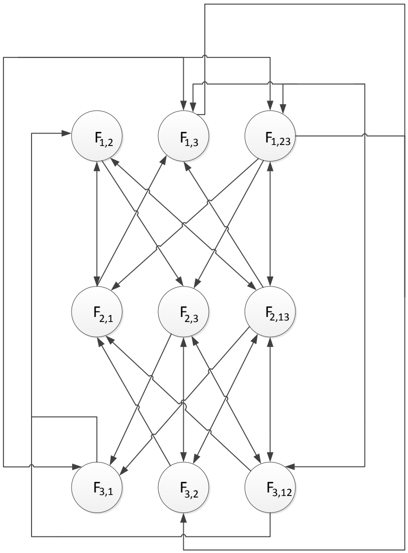
Step: Create Graph for One Caching Problem

- Get $\mathbf{d}$ (e.g. $\mathbf{d} = (1, 2, 3)$)
- General split F_i to $F_{i,\mathcal{W}}$
  - $\mathcal{W} \in 2^{[3]} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$
- Each node = desired sub-file
- Row j for user j
- Edge from i to j if user j knows sub-file i



- K. Wan, D. Tuninetti and P. Piantanida, "On the Optimality of Uncoded Cache Placement", 2016

Step: Create Graph for One Caching Problem



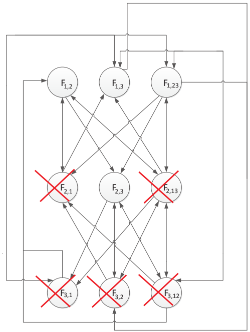
[1] Image source: Wan et al. 2017

Step: Create Maximal Acyclic Subgraph - Arrows Down

$$F_{d_1,\mathcal{W}_1}\text{for all } \mathcal{W}_1 \subseteq [1:3] \setminus \{1\}$$
$$F_{d_2,\mathcal{W}_2}\text{for all } \mathcal{W}_2 \subseteq [1:3] \setminus \{1,2\}$$
$$F_{d_3,\mathcal{W}_3}\text{for all } \mathcal{W}_3 \subseteq [1:3] \setminus \{1,2,3\} = \emptyset$$

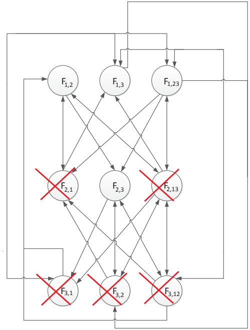
Step: Create Maximal Acyclic Subgraph - Arrows Down

$$F_{d_1,\mathcal{W}_1}\text{for all } \mathcal{W}_1 \subseteq [1:3] \setminus \{1\}$$
$$F_{d_2,\mathcal{W}_2}\text{for all } \mathcal{W}_2 \subseteq [1:3] \setminus \{1,2\}$$
$$F_{d_3,\mathcal{W}_3}\text{for all } \mathcal{W}_3 \subseteq [1:3] \setminus \{1,2,3\} = \emptyset$$



Step: Create Maximal Acyclic Subgraph - Arrows Down

$$F_{d_1,\mathcal{W}_1}\text{for all } \mathcal{W}_1 \subseteq [1:3] \setminus \{1\}$$
$$F_{d_2,\mathcal{W}_2}\text{for all } \mathcal{W}_2 \subseteq [1:3] \setminus \{1,2\}$$
$$F_{d_3,\mathcal{W}_3}\text{for all } \mathcal{W}_3 \subseteq [1:3] \setminus \{1,2,3\} = \emptyset$$



$$T_u(M) \geq |F_{1,\emptyset}| + |F_{1,2}| + |F_{1,3}| + |F_{1,23}| + |F_{2,\emptyset}| + |F_{2,3}| + |F_{3,\emptyset}|$$

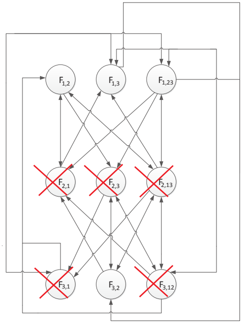
Step: More Acyclic Subgraphs - Permute Users ($\pi = (1, 3, 2)$)

$$d_{\pi_1} = d_1 = 1; \mathcal{W}_1 \subseteq [1:3] \setminus \{\pi_1\} = \{2, 3\}$$
$$d_{\pi_2} = d_3 = 3; \mathcal{W}_1 \subseteq [1:3] \setminus \{\pi_1, \pi_2\} = \{2\}$$
$$d_{\pi_3} = d_2 = 2; \mathcal{W}_3 \subseteq [1:3] \setminus \{\pi_1, \pi_2, \pi_3\} = \emptyset$$

Step: More Acyclic Subgraphs - Permute Users ($\pi = (1, 3, 2)$)

$$d_{\pi_1} = d_1 = 1; \mathcal{W}_1 \subseteq [1:3] \setminus \{\pi_1\} = \{2, 3\}$$
$$d_{\pi_2} = d_3 = 3; \mathcal{W}_1 \subseteq [1:3] \setminus \{\pi_1, \pi_2\} = \{2\}$$
$$d_{\pi_3} = d_2 = 2; \mathcal{W}_3 \subseteq [1:3] \setminus \{\pi_1, \pi_2, \pi_3\} = \emptyset$$

$$T_u(M) \geq |F_{1,\emptyset}| + |F_{1,2}| + |F_{1,3}| + |F_{1,23}| + |F_{3,2}| + |F_{3,\emptyset}| + |F_{2,\emptyset}|$$



Step: More Graphs for Symmetry - Add Acyclic Subgraphs

- $[3!]$ possible demand vectors
- $[3!]$ possible permutations
- sum $[3!]^2$ possible bounds

[1] Wan-Tuninetti-Piantanida 16

Step: More Graphs for Symmetry - Add Acyclic Subgraphs

- $[3!]$ possible demand vectors
- $[3!]$ possible permutations
- sum $[3!]^2$ possible bounds

$$T_u(M) \geq |F_{1,\emptyset}| + |F_{1,2}| + |F_{1,3}| + |F_{1,23}| + |F_{2,\emptyset}| + |F_{2,3}| + |F_{3,\emptyset}|$$
$$T_u(M) \geq |F_{1,\emptyset}| + |F_{1,2}| + |F_{1,3}| + |F_{1,23}| + |F_{3,2}| + |F_{3,\emptyset}| + |F_{2,\emptyset}|$$
$$\vdots$$

[1] Wan-Tuninetti-Piantanida 16

Step: More Graphs for Symmetry - Add Acyclic Subgraphs

- $[3!]$ possible demand vectors
- $[3!]$ possible permutations
- sum $[3!]^2$ possible bounds

$$T_u(M) \geq |F_{1,\emptyset}| + |F_{1,2}| + |F_{1,3}| + |F_{1,23}| + |F_{2,\emptyset}| + |F_{2,3}| + |F_{3,\emptyset}|$$
$$T_u(M) \geq |F_{1,\emptyset}| + |F_{1,2}| + |F_{1,3}| + |F_{1,23}| + |F_{3,2}| + |F_{3,\emptyset}| + |F_{2,\emptyset}|$$
$$\vdots$$

$$T_u(M)(3!)^2 \geq \sum_{\mathbf{d}} \sum_{\pi} \sum_{j \in [3]} \sum_{\mathcal{W}_j \in 2^{[3]} : \mathcal{W}_j \setminus \{\pi_1, \cdots, \pi_j\}} |F_{d_{\pi_j}, \mathcal{W}_j}|$$

[1] Wan-Tuninetti-Piantanida 16

Step: Exploit Symmetry - Counting Arguments

- Each sub-file that is cached in $|\mathcal{W}| = t$ caches, appears an equal number of times

[1] Wan-Tuninetti-Piantanida 16

Step: Exploit Symmetry - Counting Arguments

- Each sub-file that is cached in $|\mathcal{W}| = t$ caches, appears an equal number of times
- Allows for nice transition, from

$$T_u(M) \geq \sum_{j \in [3]} \sum_{\mathcal{W}_j \in 2^{[3]} : \mathcal{W}_j \setminus \{\pi_1, \cdots, \pi_j\}} |F_{d_{\pi_j}, \mathcal{W}_j}|$$

[1] Wan-Tuninetti-Piantanida 16

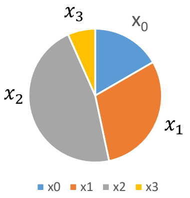
Step: Exploit Symmetry - Counting Arguments

- Each sub-file that is cached in $|\mathcal{W}| = t$ caches, appears an equal number of times
- Allows for nice transition, from

$$T_u(M) \geq \sum_{j \in [3]} \sum_{\mathcal{W}_j \in 2^{[3]} : \mathcal{W}_j \setminus \{\pi_1, \cdots, \pi_j\}} |F_{d_{\pi_j}, \mathcal{W}_j}|$$

- to

$$T_u(M) \geq \sum_{i \in [0:3]} x_i \frac{1 - i/3}{1 + i} = x_0 + \frac{1}{3} x_1 + \frac{1}{9} x_2 + 0 x_3 \qquad (1)$$



[1] Wan-Tuninetti-Piantanida 16

Step: Final optimization

- Optimize

$$x_0 + \frac{1}{3} \cdot x_1 + \frac{1}{9} \cdot x_2 + 0 \cdot x_3$$

[1] Wan-Tuninetti-Piantanida 16

Step: Final optimization

- Optimize

$$x_0 + \frac{1}{3} \cdot x_1 + \frac{1}{9} \cdot x_2 + 0 \cdot x_3$$

- Under total file size constraint:

$$x_0 + x_1 + x_2 + x_3 = 3$$

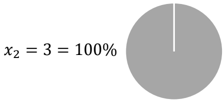
- total cache size constraint:

$$0 \cdot x_0 + 1 \cdot x_1 + 2x_2 + 3 \cdot x_3 \leq 3M$$

[1] Wan-Tuninetti-Piantanida 16

Step: Final optimization

- Optimize
$$x_0 + \frac{1}{3} \cdot x_1 + \frac{1}{9} \cdot x_2 + 0 \cdot x_3$$
- Under total file size constraint:
$$x_0 + x_1 + x_2 + x_3 = 3$$
- total cache size constraint:
$$0 \cdot x_0 + 1 \cdot x_1 + 2x_2 + 3 \cdot x_3 \leq 3M$$
- Specific solution (*See also Yu, Maddah-Ali, Avestimehr*):



[1]Wan-Tuninetti-Piantanida 16

Step: Final optimization

- Optimize

$$x_0 + \frac{1}{3} \cdot x_1 + \frac{1}{9} \cdot x_2 + 0 \cdot x_3$$

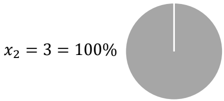
- Under total file size constraint:

$$x_0 + x_1 + x_2 + x_3 = 3$$

- total cache size constraint:

$$0 \cdot x_0 + 1 \cdot x_1 + 2x_2 + 3 \cdot x_3 \leq 3M$$

- Specific solution (*See also Yu, Maddah-Ali, Avestimehr*):



- Final answer:

$$T^* \geq \frac{K - t}{1 + t} = \frac{K(1 - \gamma)}{1 + K\gamma}$$

[1] Wan-Tuninetti-Piantanida 16

Toward average delay: N<K

- Optimality result extends to case of $N < K$

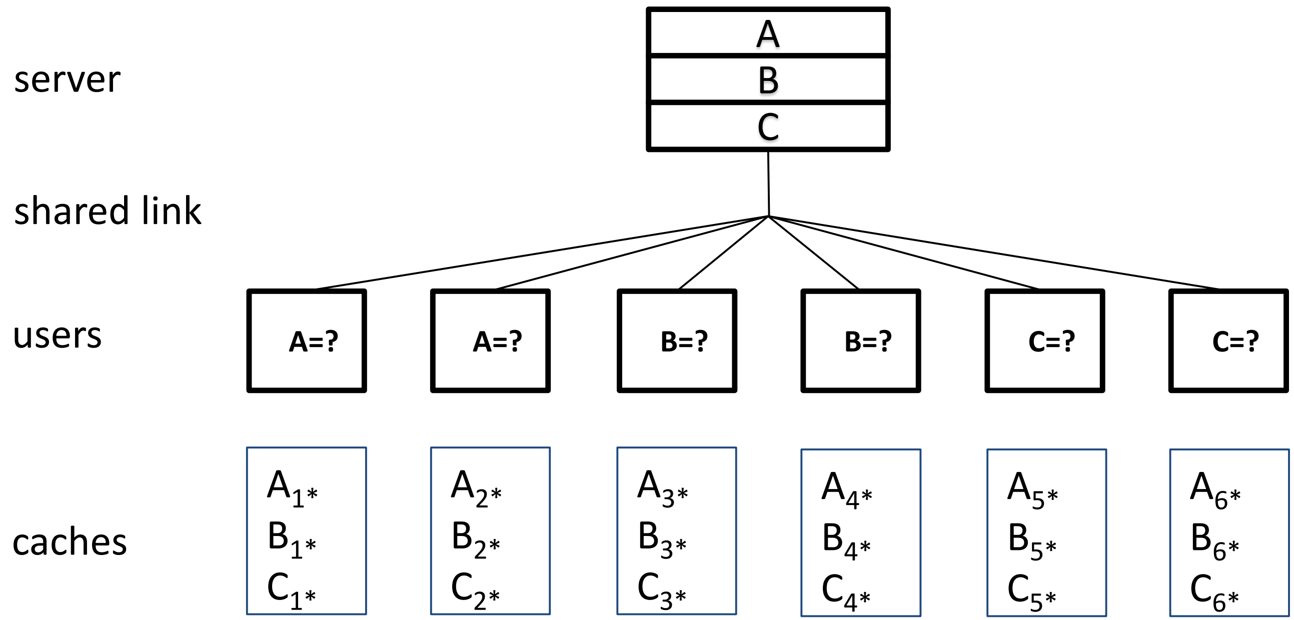
$$T^* = \frac{\binom{K}{K\gamma+1} - \binom{K-\min(K,N)}{K\gamma+1}}{\binom{K}{K\gamma}}$$

- As well as to average-delay case for uniform distribution
  - Single stream scenario

- Proof based on new scheme for $N < K$

Yu, Maddah-Ali, Avestimehr TiT 2018

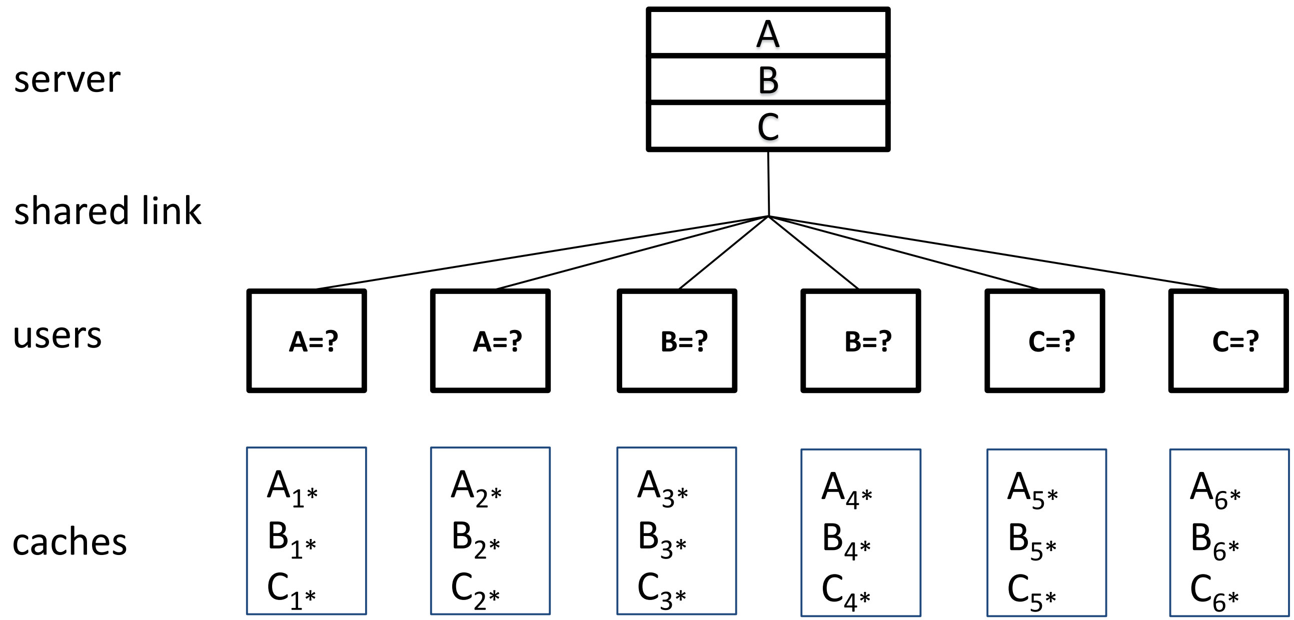
Toward average delay: N<K



- $K = 6, N = 3, M = 1$

Image-source and Example: Yu et al.

Toward average delay: N<K



- $K = 6, N = 3, M = 1$
- Subpacketization $\binom{6}{2}$ = 15

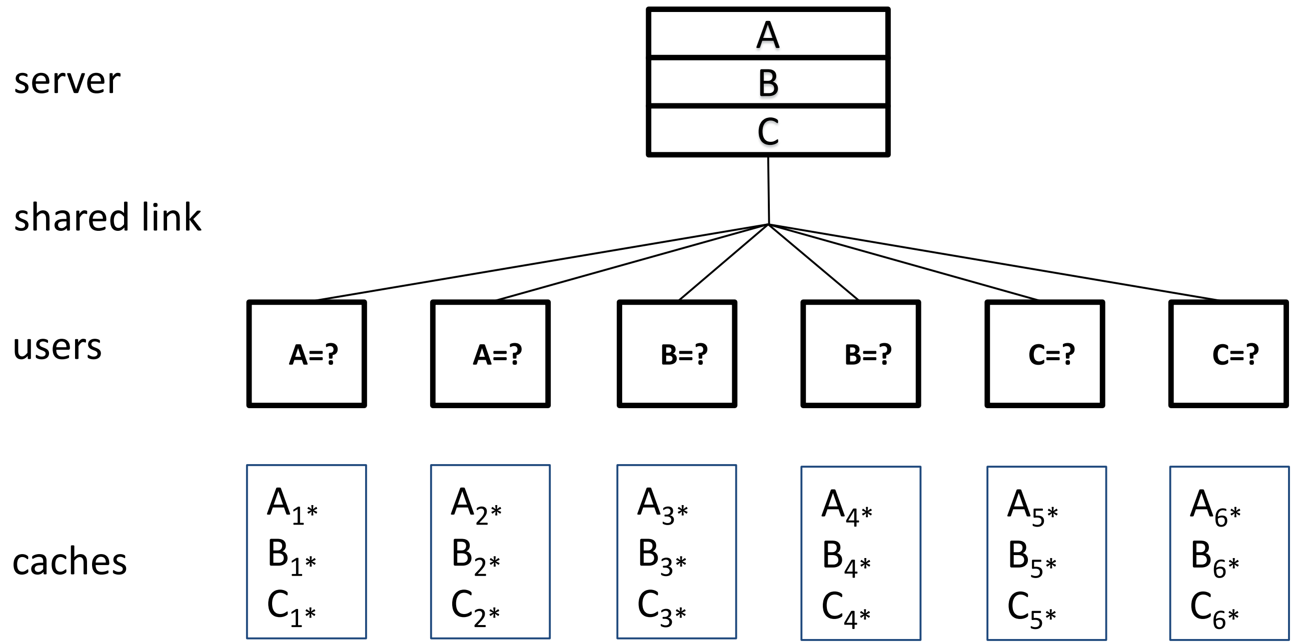
  A_{12}, A_{13}, A_{14}, …, A_{56},
  B_{12}, …, B_{56}, C_{12}, …, C_{56}.

- MN Placement
- MN Delivery:

Broadcast $\binom{6}{3}$ = 20 XORs $\Rightarrow T = \frac{20}{15}$.

Image-source and Example: Yu et al.

Toward average delay: N<K



- $K = 6, N = 3, M = 1$
- Subpacketization $\binom{6}{2} = 15$

  $A_{12}, A_{13}, A_{14}, \ldots, A_{56},$
  $B_{12}, \ldots, B_{56}, C_{12}, \ldots, C_{56}.$

- MN Placement

- MN Delivery:

Broadcast $\binom{6}{3} = 20$ XORs $\Rightarrow T = \frac{20}{15}$.

- Modified scheme can skip 1 XOR

$$\Rightarrow T = \frac{19}{15} \quad \text{Optimal}$$

Image-source and Example: Yu et al.

Toward average delay: N<K

- Ordinarily would send 20 XORs

- $X_{123} = A_{23} \oplus A_{13} \oplus \mathbf{B_{12}}$
- $X_{124} = A_{24} \oplus A_{14} \oplus \mathbf{B_{12}}$
- $X_{125} = A_{25} \oplus A_{15} \oplus \mathbf{C_{12}}$
- $X_{126} = A_{46} \oplus A_{26} \oplus \mathbf{C_{12}}$
- …..
- $X_{146} = \mathbf{A_{46}} \oplus B_{16} \oplus C_{14}$
- …..
- $X_{246} = \mathbf{A_{46}} \oplus B_{26} \oplus C_{24}$
- …..
- $X_{456} = B_{56} \oplus C_{46} \oplus C_{45}$

Example: Yu et al.

Toward average delay: N<K

- Ordinarily would send 20 XORs

- $X_{123} = A_{23} \oplus A_{13} \oplus \mathbf{B_{12}}$
- $X_{124} = A_{24} \oplus A_{14} \oplus \mathbf{B_{12}}$
- $X_{125} = A_{25} \oplus A_{15} \oplus \mathbf{C_{12}}$
- $X_{126} = A_{46} \oplus A_{26} \oplus \mathbf{C_{12}}$
- .....
- $X_{146} = \mathbf{A_{46}} \oplus B_{16} \oplus C_{14}$
- .....
- $X_{246} = \mathbf{A_{46}} \oplus B_{26} \oplus C_{24}$
- .....
- $X_{456} = B_{56} \oplus C_{46} \oplus C_{45}$

Example: Yu et al.

Toward average delay: N<K

- Ordinarily would send 20 XORs

- $X_{123} = A_{23} \oplus A_{13} \oplus \mathbf{B_{12}}$
- $X_{124} = A_{24} \oplus A_{14} \oplus \mathbf{B_{12}}$
- $X_{125} = A_{25} \oplus A_{15} \oplus \mathbf{C_{12}}$
- $X_{126} = A_{46} \oplus A_{26} \oplus \mathbf{C_{12}}$
- .....
- $X_{146} = \mathbf{A_{46}} \oplus B_{16} \oplus C_{14}$
- .....
- $\cancel{X_{246} = \mathbf{A_{46}} \oplus B_{26} \oplus C_{24}}$
- .....
- $X_{456} = B_{56} \oplus C_{46} \oplus C_{45}$

Example: Yu et al.

Toward average delay: N<K

- Skip $\mathbf{X_{246}} = A_{46} \oplus B_{26} \oplus C_{24}$
- Focus on user 2 (needs $\mathbf{A_{46}}$)

Example: Yu et al.

Toward average delay: N<K

- Skip $\mathbf{X_{246}} = A_{46} \oplus B_{26} \oplus C_{24}$
- Focus on user 2 (needs $\mathbf{A_{46}}$)

- Have transmitted:
  - $X_{146} = A_{46} \oplus B_{16} \oplus C_{14}$
  - $X_{145} = A_{45} \oplus B_{15} \oplus C_{14}$
  - $X_{136} = A_{36} \oplus B_{16} \oplus C_{13}$
  - $X_{135} = A_{35} \oplus B_{15} \oplus C_{13}$

Example: Yu et al.

Toward average delay: N<K

- Skip $\mathbf{X_{246}} = A_{46} \oplus B_{26} \oplus C_{24}$
- Focus on user 2 (needs $\mathbf{A_{46}}$)

- Have transmitted:
  - $X_{146} = A_{46} \oplus B_{16} \oplus C_{14}$
  - $X_{145} = A_{45} \oplus B_{15} \oplus C_{14}$
  - $X_{136} = A_{36} \oplus B_{16} \oplus C_{13}$
  - $X_{135} = A_{35} \oplus B_{15} \oplus C_{13}$

- Adding up:
  - $X_{146} \oplus X_{145} \oplus X_{136} \oplus X_{135} = A_{46} \oplus \mathbf{A_{45}} \oplus \mathbf{A_{36}} \oplus \mathbf{A_{35}}$

Example: Yu et al.

Toward average delay: N<K

- Skip $\mathbf{X_{246}} = A_{46} \oplus B_{26} \oplus C_{24}$
- Focus on user 2 (needs $\mathbf{A_{46}}$)

- Have transmitted:
  - $X_{146} = A_{46} \oplus B_{16} \oplus C_{14}$
  - $X_{145} = A_{45} \oplus B_{15} \oplus C_{14}$
  - $X_{136} = A_{36} \oplus B_{16} \oplus C_{13}$
  - $X_{135} = A_{35} \oplus B_{15} \oplus C_{13}$

- Adding up:
  - $X_{146} \oplus X_{145} \oplus X_{136} \oplus X_{135} = A_{46} \oplus \mathbf{A_{45}} \oplus \mathbf{A_{36}} \oplus \mathbf{A_{35}}$

- $A_{45} \leftarrow X_{245}$
- $A_{36} \leftarrow X_{236}$
- $A_{35} \leftarrow X_{235}$

Example: Yu et al.

Toward average delay: N<K

- Skip $\mathbf{X_{246}} = A_{46} \oplus B_{26} \oplus C_{24}$
- Focus on user 2 (needs $\mathbf{A_{46}}$)

- Have transmitted:
  - $X_{146} = A_{46} \oplus B_{16} \oplus C_{14}$
  - $X_{145} = A_{45} \oplus B_{15} \oplus C_{14}$
  - $X_{136} = A_{36} \oplus B_{16} \oplus C_{13}$
  - $X_{135} = A_{35} \oplus B_{15} \oplus C_{13}$

- Adding up:
  - $X_{146} \oplus X_{145} \oplus X_{136} \oplus X_{135} = A_{46} \oplus \mathbf{A_{45}} \oplus \mathbf{A_{36}} \oplus \mathbf{A_{35}}$

- $A_{45} \leftarrow X_{245} = A_{45} \oplus B_{25} \oplus C_{24}$
- $A_{36} \leftarrow X_{236} = A_{36} \oplus B_{26} \oplus C_{23}$
- $A_{35} \leftarrow X_{235} = A_{35} \oplus B_{25} \oplus C_{23}$

- ➔ Decode $A_{\{4,6\}}$ (➔ similarly B_{26} for user 4, C_{24} for user 6).

Example: Yu et al.

Extensions wide open

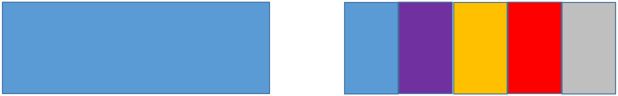
- Done for very special case of the single-stream BC

Extensions wide open

- Done for very special case of the single-stream BC
- Approach requires new ‘tricks’ when setting changes
    - Shared caches (Hachem, Karamchandrani et Diggavi)
    - Multiple file demands (Wei-Ulukus)
    - Multi-layer files (high-def, low-def) (Yang-Gündüz)

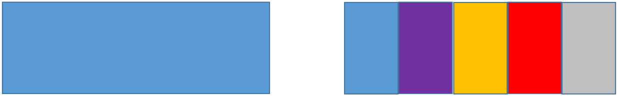
Extensions wide open

- Done for very special case of the single-stream BC
- Approach requires new 'tricks' when setting changes
  - Shared caches (Hachem, Karamchandrani et Diggavi)
  - Multiple file demands (Wei-Ulukus)
  - Multi-layer files (high-def, low-def) (Yang-Gündüz)
- Need methods needed that preserve some asymmetry



Extensions wide open

- Done for very special case of the single-stream BC
- Approach requires new 'tricks' when setting changes
  - Shared caches (Hachem, Karamchandrani et Diggavi)
  - Multiple file demands (Wei-Ulukus)
  - Multi-layer files (high-def, low-def) (Yang-Gündüz)
- Need methods needed that preserve some asymmetry



- Need methods that preserve topology
- Need methods reflecting multiple-senders/multiple-antennas.