Linear Coded Caching Schemes

Minquan Cheng

Guangxi Key Lab of Multi-source Information Mining & Security
Guangxi Normal University

Joint work with: Jie Li, Xiaohu Tang, Ruizhong Wei

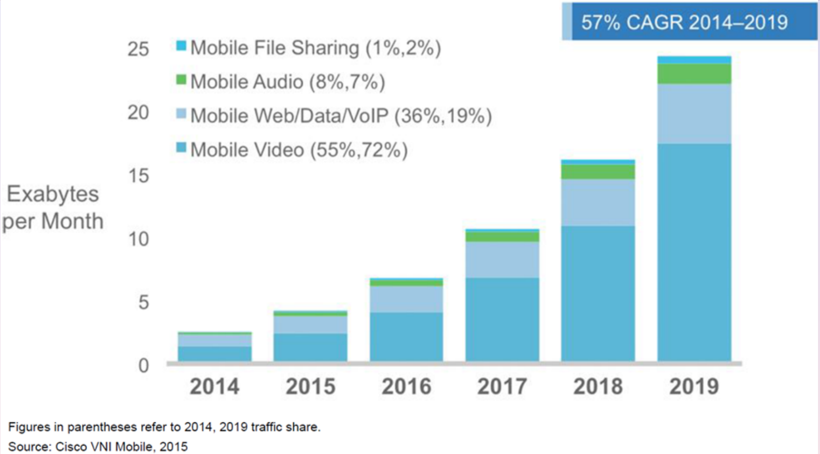
July 5, 2018

1. Background
2. Coded caching scheme
3. Linear constructions

1. Background
2. Coded caching scheme
3. Linear constructions

1. Background
2. Coded caching scheme
3. Linear constructions

The amount of transmission in wireless network

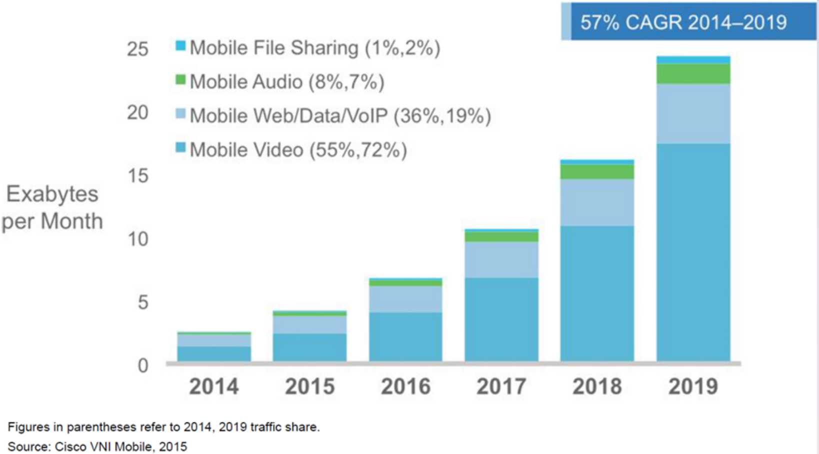


- Video-on-demand drives wireless traffic growth
- The available bandwidth is finite

---

[1] Cisco Visual Networking Index, Global Mobile Data Traffic Forecast Update, 2014-2019, White Paper, 2015.

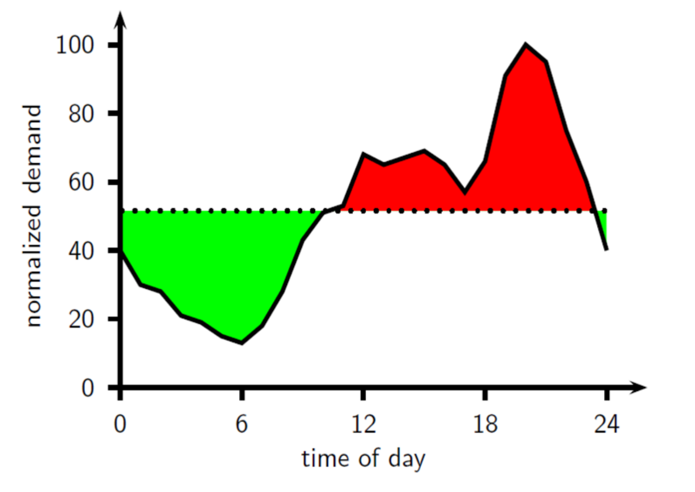
The amount of transmission in wireless network



- Video-on-demand drives wireless traffic growth
- The available bandwidth is finite

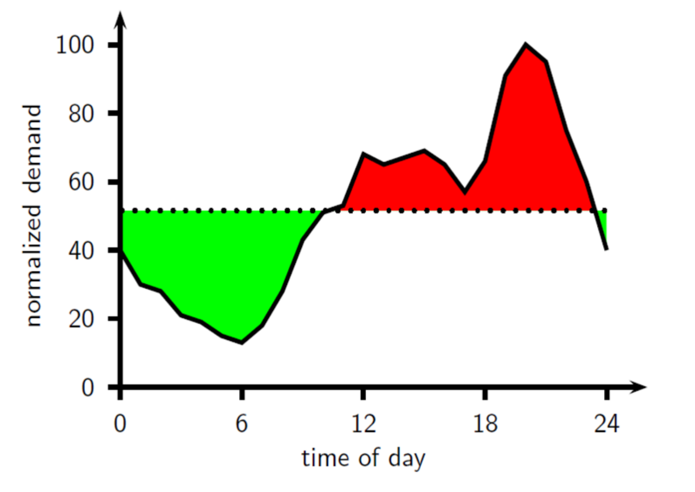
[1] Cisco Visual Networking Index, Global Mobile Data Traffic Forecast Update, 2014-2019, White Paper, 2015.

The high temporal variability of network traffic



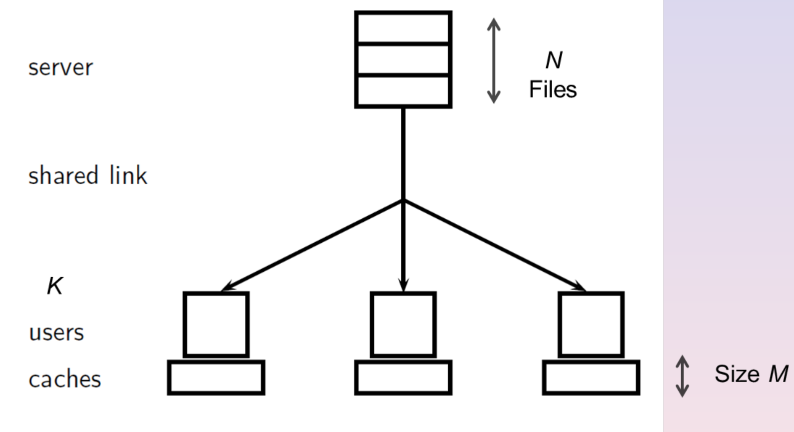
- Off peak traffic times: $0 \sim 12$h
- Peak traffic times: $12 \sim 24$h

The high temporal variability of network traffic



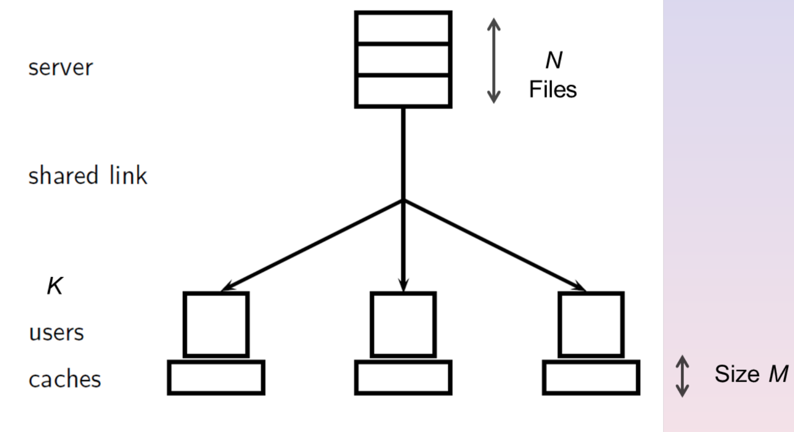
- Off peak traffic times: $0 \sim 12$h
- Peak traffic times: $12 \sim 24$h

Centralized wireless network



- A server stores N files, each of size F packets ($N = 3$)
- K users, each access a cache of size MF packets ($K = 3$)

Centralized wireless network



- A server stores N files, each of size F packets ($N = 3$)
- K users, each access a cache of size MF packets ($K = 3$)

Caching system

- Placement Phase
  - parts of each file is partially cached at each user
  - without the knowledge of user's demands
- Delivery Phase

  requested file with help its cached contents.

$$R = \max\left\{ R_{\mathbf{d}} \mid \forall \mathbf{d} \in [0, N)^K \right\}.$$

  R is always called the rate of a coded caching scheme.
- **Objective 1:** R is as small as possible.

Caching system

- Placement Phase
  - parts of each file is partially cached at each user
  - without the knowledge of user's demands
- Delivery Phase
  - Server sends $R_{\mathbf{d}}$ files to users such that each user decodes its requested file with help its cached contents.

$$R = \max\left\{ R_{\mathbf{d}} \mid \forall \mathbf{d} \in [0, N)^K \right\}.$$

R is always called the rate of a coded caching scheme.

- **Objective 1:** R is as small as possible.

Caching system

- Placement Phase
  - parts of each file is partially cached at each user
  - without the knowledge of user's demands
- Delivery Phase
  - Assume that user k requests the d_kth file. Denote all the request file numbers by $\mathbf{d} = (d_0, d_1, \ldots, d_{K-1})$.
  - Server sends $R_{\mathbf{d}}$ files to users such that each user decodes its requested file with help its cached contents.

$$R = \max\left\{R_{\mathbf{d}} \mid \forall \mathbf{d} \in [0, N)^K\right\}.$$

R is always called the rate of a coded caching scheme.

- **Objective 1:** R is as small as possible.

Caching system

- Placement Phase
  - parts of each file is partially cached at each user
  - without the knowledge of user's demands
- Delivery Phase
  - Assume that user k requests the d_kth file. Denote all the request file numbers by $\mathbf{d} = (d_0, d_1, \ldots, d_{K-1})$.
  - Server sends $R_{\mathbf{d}}$ files to users such that each user decodes its requested file with help its cached contents.

$$R = \max\left\{R_{\mathbf{d}} \mid \forall \mathbf{d} \in [0, N)^K\right\}.$$

R is always called the rate of a coded caching scheme.

- **Objective 1:** R is as small as possible.

- Placement Phase
  - parts of each file is partially cached at each user
  - without the knowledge of user's demands
- Delivery Phase
  - Assume that user k requests the d_kth file. Denote all the request file numbers by $\mathbf{d} = (d_0, d_1, \ldots, d_{K-1})$.
  - Server sends $R_{\mathbf{d}}$ files to users such that each user decodes its requested file with help its cached contents.

$$R = \max\left\{R_{\mathbf{d}} \mid \forall \mathbf{d} \in [0, N)^K\right\}.$$

R is always called the rate of a coded caching scheme.

- **Objective 1:** R is as small as possible.

- Placement Phase
  - parts of each file is partially cached at each user
  - without the knowledge of user's demands
- Delivery Phase
  - Assume that user k requests the d_kth file. Denote all the request file numbers by $\mathbf{d} = (d_0, d_1, \ldots, d_{K-1})$.
  - Server sends $R_{\mathbf{d}}$ files to users such that each user decodes its requested file with help its cached contents.

$$R = \max\left\{R_{\mathbf{d}} \mid \forall \mathbf{d} \in [0, N)^K\right\}.$$

R is always called the rate of a coded caching scheme.

- **Objective 1:** R is as small as possible.

- Placement Phase
  - parts of each file is partially cached at each user
  - without the knowledge of user's demands
- Delivery Phase
  - Assume that user k requests the d_kth file. Denote all the request file numbers by $\mathbf{d} = (d_0, d_1, \ldots, d_{K-1})$.
  - Server sends $R_{\mathbf{d}}$ files to users such that each user decodes its requested file with help its cached contents.

$$R = \max\left\{R_{\mathbf{d}} \mid \forall \mathbf{d} \in [0, N)^K\right\}.$$

R is always called the rate of a coded caching scheme.

- **Objective 1:** R is as small as possible.

MN scheme

In 2014, Ali and Niesen proposed that

coded caching scheme can be used to further reduce R.

- The rate of MN scheme is ordered optimal
- It is widely used in heterogeneous wireless network, such as D2D, hierarchical network and so on.
- Best paper award of IEEE IT
- Cited by 680 times

[2] M. A. Maddah-Ali and U. Niesen, IEEE Trans. Inf. Theory, 60(5), (2014).

In 2014, Ali and Niesen proposed that

coded caching scheme can be used to further reduce R.

- The rate of MN scheme is ordered optimal
- It is widely used in heterogeneous wireless network, such as D2D, hierarchical network and so on.
- Best paper award of IEEE IT
- Cited by 680 times

---

[2] M. A. Maddah-Ali and U. Niesen, IEEE Trans. Inf. Theory, 60(5), (2014).

In 2014, Ali and Niesen proposed that

coded caching scheme can be used to further reduce R.

- The rate of MN scheme is ordered optimal
- It is widely used in heterogeneous wireless network, such as D2D, hierarchical network and so on.
- Best paper award of IEEE IT
- Cited by 680 times

[2] M. A. Maddah-Ali and U. Niesen, IEEE Trans. Inf. Theory, 60(5), (2014).

In 2014, Ali and Niesen proposed that

coded caching scheme can be used to further reduce R.

- The rate of MN scheme is ordered optimal
- It is widely used in heterogeneous wireless network, such as D2D, hierarchical network and so on.
- Best paper award of IEEE IT
- Cited by 680 times

[2] M. A. Maddah-Ali and U. Niesen, IEEE Trans. Inf. Theory, 60(5), (2014).

In 2014, Ali and Niesen proposed that

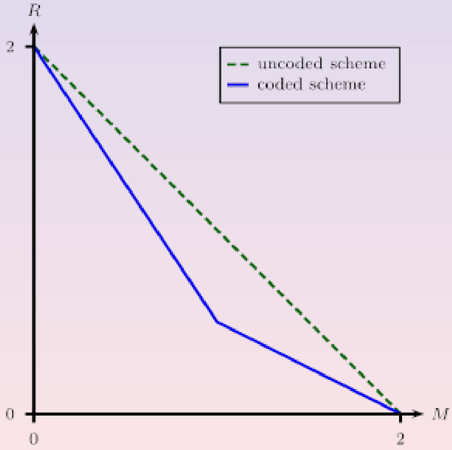
coded caching scheme can be used to further reduce R.

- The rate of MN scheme is ordered optimal
- It is widely used in heterogeneous wireless network, such as D2D, hierarchical network and so on.
- Best paper award of IEEE IT
- Cited by 680 times

[2] M. A. Maddah-Ali and U. Niesen, IEEE Trans. Inf. Theory, 60(5), (2014).

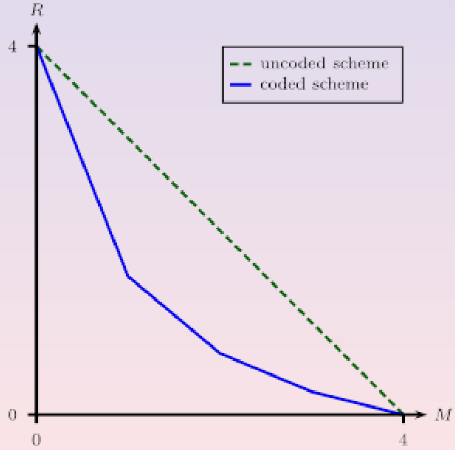
Comparison: Uncoded VS Coded

N files, $K = N$ users, Cache size M files



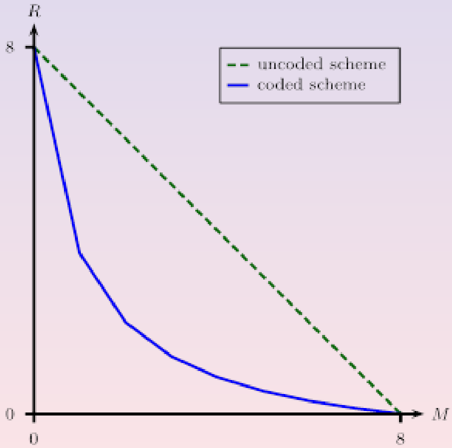
Comparison: Uncoded VS Coded

N files, $K = N$ users, Cache size M files



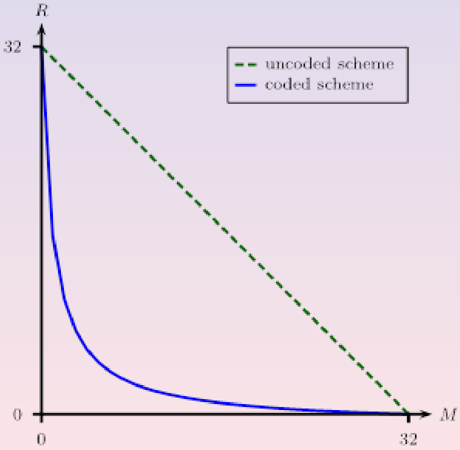
Comparison: Uncoded VS Coded

N files, $K = N$ users, Cache size M files



Comparison: Uncoded VS Coded

N files, $K = N$ users, Cache size M files



The second objective

- **Disadvantage:** MN scheme is unfeasible when K is large.
- **Objective 2:** F is as small as possible for the fixed R.

[2] M. A. Maddah-Ali and U. Niesen, IEEE Trans. Inf. Theory, 60(5), (2014).

The second objective

- **Disadvantage:** MN scheme is unfeasible when K is large.
- **Objective 2:** F is as small as possible for the fixed R.

[2] M. A. Maddah-Ali and U. Niesen, IEEE Trans. Inf. Theory, 60(5), (2014).

Table: Previously known constructions

results	M/N	R	F
MN scheme	$\frac{1}{q}$	$\frac{K(q-1)}{q+K}$	$\sim \frac{q}{\sqrt{2\pi K(q-1)}} \cdot e^{\frac{K}{q}\left(\ln q+(q-1)\ln \frac{q}{q-1}\right)}$
The scheme	$\frac{1}{q}$	$q-1$	$e^{(\frac{K}{q}-1)\ln q}$

[3] Q. Yan, M. Cheng, X. Tang, and Q. Chen, IEEE Trans. Inform. Theory, 63(9), 2017.

Placement phase:

- User k caches

$$\mathcal{Z}_k = \{\phi_k(W_n) \mid n \in [0, N)\}$$

where $\phi_k(x)$ (*caching function*) has the size of $\frac{FM}{N}$ packets.

Delivery phase:

- Given a request $\mathbf{d} = (d_0, d_1, \cdots, d_{K-1})$, sever sends

$$\mathbf{X_d} = \psi_0(W_{d_0}) + \psi_1(W_{d_1}) + \ldots + \psi_{K-1}(W_{d_{K-1}})$$

to users where ψ_k(*coding function*) has the size of RF packets.

Linear characterization

Placement phase:

- User k caches

$$\mathcal{Z}_k = \{\phi_k(W_n) \mid n \in [0, N)\}$$

where $\phi_k(x)$ (*caching function*) has the size of $\frac{FM}{N}$ packets.

Delivery phase:

- Given a request $\mathbf{d} = (d_0, d_1, \cdots, d_{K-1})$, sever sends

$$\mathbf{X_d} = \psi_0(W_{d_0}) + \psi_1(W_{d_1}) + \ldots + \psi_{K-1}(W_{d_{K-1}})$$

to users where ψ_k(*coding function*) has the size of RF packets.

Placement phase:

- User k caches

$$\mathcal{Z}_k = \{\phi_k(W_n) \mid n \in [0, N)\}$$

where $\phi_k(x)$ (*caching function*) has the size of $\frac{FM}{N}$ packets.

Delivery phase:

- Given a request $\mathbf{d} = (d_0, d_1, \cdots, d_{K-1})$, sever sends

$$\mathbf{X_d} = \psi_0(W_{d_0}) + \psi_1(W_{d_1}) + \ldots + \psi_{K-1}(W_{d_{K-1}})$$

to users where ψ_k(*coding function*) has the size of RF packets.

Linear characterization

- User k obtains
$$W'_{d_k} = \chi_k(\mathbf{X_d})$$
where χ_k(*decoding function*) has the size of $F(1-\frac{M}{N})$ packets.
- Our goal: W_{d_k} can be obtained by $\phi_k(W_{d_k})$ and W'_{d_k}.

- User k obtains
$$W'_{d_k} = \chi_k(\mathbf{X}_\mathbf{d})$$
where χ_k(*decoding function*) has the size of $F(1-\frac{M}{N})$ packets.
- Our goal: W_{d_k} can be obtained by $\phi_k(W_{d_k})$ and W'_{d_k}.

Linear characterization

- Placement phase: User k caches

$$\mathcal{Z}_k = \{\mathbf{S}_k W_n \mid n \in [0, N)\}$$

where $\mathbf{S}_i$ (*caching matrix*) is an $\frac{FM}{N} \times F$ matrix.

- Delivery phase: Given a request $\mathbf{d} = (d_0, d_1, \cdots, d_{K-1})$, sever sends

$$\mathbf{X}_\mathbf{d} = \mathbf{A}_0 W_{d_0} + \mathbf{A}_1 W_{d_1} + \ldots + \mathbf{A}_{K-1} W_{d_{K-1}}$$

to users where $\mathbf{A}_k$(*coding matrix*) is a $RF \times F$ matrix.

- User k obtains

$$W'_{d_k} = \mathbf{S}'_k \mathbf{X}_\mathbf{d}$$

where $\mathbf{S}'_k$(*decoding matrix*) is an $F(1 - \frac{M}{N}) \times RF$ matrix.

- Placement phase: User k caches

$$\mathcal{Z}_k = \{\mathbf{S}_k W_n \mid n \in [0, N)\}$$

where $\mathbf{S}_i$ (*caching matrix*) is an $\frac{FM}{N} \times F$ matrix.

- Delivery phase: Given a request $\mathbf{d} = (d_0, d_1, \cdots, d_{K-1})$, sever sends

$$\mathbf{X}_{\mathbf{d}} = \mathbf{A}_0 W_{d_0} + \mathbf{A}_1 W_{d_1} + \ldots + \mathbf{A}_{K-1} W_{d_{K-1}}$$

to users where $\mathbf{A}_k$(*coding matrix*) is a $RF \times F$ matrix.

- User k obtains

$$W'_{d_k} = \mathbf{S}'_k \mathbf{X}_{\mathbf{d}}$$

where $\mathbf{S}'_k$(*decoding matrix*) is an $F(1 - \frac{M}{N}) \times RF$ matrix.

Linear characterization

- Placement phase: User k caches

$$\mathcal{Z}_k = \{\mathbf{S}_k W_n \mid n \in [0, N)\}$$

where $\mathbf{S}_i$ (*caching matrix*) is an $\frac{FM}{N} \times F$ matrix.

- Delivery phase: Given a request $\mathbf{d} = (d_0, d_1, \cdots, d_{K-1})$, sever sends

$$\mathbf{X}_{\mathbf{d}} = \mathbf{A}_0 W_{d_0} + \mathbf{A}_1 W_{d_1} + \ldots + \mathbf{A}_{K-1} W_{d_{K-1}}$$

to users where $\mathbf{A}_k$(*coding matrix*) is a $RF \times F$ matrix.

- User k obtains

$$W'_{d_k} = \mathbf{S}'_k \mathbf{X}_{\mathbf{d}}$$

where $\mathbf{S}'_k$(*decoding matrix*) is an $F(1 - \frac{M}{N}) \times RF$ matrix.

Theorem

For any request $\mathbf{d}$ and the related signals $\mathbf{X}_{\mathbf{d}}$, user k can obtain the required W_{d_k} if and only if the matrices satisfy the following conditions.

$$\operatorname{rank}\begin{pmatrix} \mathbf{S}_k \\ \mathbf{S}'_k \mathbf{A}_{k'} \end{pmatrix} = \begin{cases} F, & \text{if } \ k = k' \\ FM/N, & \text{otherwise} \end{cases} \tag{1}$$

for all $0 \leq k' < K$.

Remark

- Formula (1) is exactly one condition of an minimum storage regenerating code

Theorem

For any request $\mathbf{d}$ and the related signals $\mathbf{X}_{\mathbf{d}}$, user k can obtain the required W_{d_k} if and only if the matrices satisfy the following conditions.

$$\operatorname{rank}\begin{pmatrix} \mathbf{S}_k \\ \mathbf{S}'_k \mathbf{A}_{k'} \end{pmatrix} = \begin{cases} F, & \text{if } \ k = k' \\ FM/N, & \text{otherwise} \end{cases} \tag{1}$$

for all $0 \le k' < K$.

Remark

- Formula (1) is exactly one condition of an minimum storage regenerating code

Theorem

For any request $\mathbf{d}$ and the related signals $\mathbf{X}_{\mathbf{d}}$, user k can obtain the required W_{d_k} if and only if the matrices satisfy the following conditions.

$$\operatorname{rank}\begin{pmatrix} \mathbf{S}_k \\ \mathbf{S}'_k\mathbf{A}_{k'} \end{pmatrix} = \begin{cases} F, & \text{if } \ k = k' \\ FM/N, & \text{otherwise} \end{cases} \tag{1}$$

for all $0 \le k' < K$.

Remark

- Formula (1) is exactly one condition of an minimum storage regenerating code

- Minimum storage regenerating (MSR) code was introduced in [4] for distributed storage systems.
- Assume that a file of size $\mathcal{M} = k\alpha$ denoted by the column vector $\mathbf{W} \in \mathbf{F}_p^{k\alpha}$ is partitioned in k parts

$$\mathbf{W} = \{\mathbf{W}_0, \mathbf{W}_1, \cdots, \mathbf{W}_{k-1}\}$$

each of size α, where p is a primepower.
- We encode $\mathbf{W}$ using an $(n = k + r, k)$ MDS code and store it across k systematic and r parity storage nodes.

---

[4] A. G. Dimakis et al., IEEE Trans. Inf. Theory, 56(9): 4539-551, 2010.

- Minimum storage regenerating (MSR) code was introduced in [4] for distributed storage systems.
- Assume that a file of size $\mathcal{M} = k\alpha$ denoted by the column vector $\mathbf{W} \in \mathbf{F}_p^{k\alpha}$ is partitioned in k parts

$$\mathbf{W} = \{\mathbf{W}_0, \mathbf{W}_1, \cdots, \mathbf{W}_{k-1}\}$$

each of size α, where p is a primepower.

- We encode $\mathbf{W}$ using an $(n = k + r, k)$ MDS code and store it across k systematic and r parity storage nodes.

[4] A. G. Dimakis et al., IEEE Trans. Inf. Theory, 56(9): 4539-551, 2010.

- Minimum storage regenerating (MSR) code was introduced in [4] for distributed storage systems.
- Assume that a file of size $\mathcal{M} = k\alpha$ denoted by the column vector $\mathbf{W} \in \mathbf{F}_p^{k\alpha}$ is partitioned in k parts

$$\mathbf{W} = \{\mathbf{W}_0, \mathbf{W}_1, \cdots, \mathbf{W}_{k-1}\}$$

  each of size α, where p is a primepower.
- We encode $\mathbf{W}$ using an $(n = k + r, k)$ MDS code and store it across k systematic and r parity storage nodes.

[4] A. G. Dimakis et al., IEEE Trans. Inf. Theory, 56(9): 4539-551, 2010.

Theorem

If an $(n = K + r, K)$ MSR code has optimal repairing bandwidth, then a (K, M, N) caching scheme with

$$F = \alpha \qquad \frac{M}{N} = \frac{1}{r} \qquad R = r - 1$$

can be obtained.

(1) holds if and only if the following formula holds.

$$\begin{cases} \mathbf{S}'_k \mathbf{A}_{k'} \subseteq \mathbf{S}_k & \text{if} \quad k \neq k' \\ \mathbf{S}_k + \mathbf{S}'_k \mathbf{A}_{k'} = \mathbb{F}_2^F & \text{if} \quad k = k' \end{cases} \qquad k, k' \in [1, K)$$

Here the sum of two subspace $\mathcal{U}$, $\mathcal{V}$ of $\mathbb{F}^F$ is defined as

$$\mathcal{U} + \mathcal{V} = \{\mathbf{u} + \mathbf{v} \mid \mathbf{u} \in \mathcal{U}, \mathbf{v} \in \mathcal{V}\}.$$

Notations

- Given an integer $s \in [0, q^m)$ where $m \in \mathbb{N}^+$, with

$$s = \sum_{l=0}^{m-1} s_l q^l$$

for integers $s_l \in [0, q)$, we refer to

$$s = (s_{m-1}, \cdots, s_0)_q$$

as the *q-ary representation* of s.

- For each integer s let

$$\mathbf{e}_s = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$$

be a q^m length vector where the sth entry is 1 and other entries are 0.

Notations

- Given an integer $s \in [0, q^m)$ where $m \in \mathbb{N}^+$, with

$$s = \sum_{l=0}^{m-1} s_l q^l$$

for integers $s_l \in [0, q)$, we refer to

$$s = (s_{m-1}, \cdots, s_0)_q$$

as the *q-ary representation* of s.

- For each integer s let

$$\mathbf{e}_s = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$$

be a q^m length vector where the sth entry is 1 and other entries are 0.

Notations

- There are m partitions of $[0, q^m)$, *i.e.*, for $u = 0, 1, \cdots, m-1$,

$$\mathcal{V}_{u,v} = \{(s_{n-1}, \cdots, s_0)_q \mid s_u = v\}, \ 0 \leq v < q.$$

Example ($q = 3$, $m = 2$)

$$\mathcal{V}_{0,0} = \{(0,0),\ (1,0),\ (2,0)\} = \{0,\ 3,\ 6\}$$
$$\mathcal{V}_{0,1} = \{(0,1),\ (1,1),\ (2,1)\} = \{1,\ 4,\ 7\}$$
$$\mathcal{V}_{0,2} = \{(0,2),\ (1,2),\ (2,2)\} = \{2,\ 5,\ 8\}$$
$$\mathcal{V}_{1,0} = \{(0,0),\ (0,1),\ (0,2)\} = \{0,\ 1,\ 2\}$$
$$\mathcal{V}_{1,1} = \{(1,0),\ (1,1),\ (1,2)\} = \{3,\ 4,\ 5\}$$
$$\mathcal{V}_{1,2} = \{(2,0),\ (2,1),\ (2,2)\} = \{6,\ 7,\ 8\}$$

Notations

- There are m partitions of $[0, q^m)$, *i.e.*, for $u = 0, 1, \cdots, m-1$,

$$\mathcal{V}_{u,v} = \{(s_{n-1}, \cdots, s_0)_q \mid s_u = v\}, \ 0 \leq v < q.$$

Example ($q = 3$, $m = 2$)

$$\begin{aligned}
\mathcal{V}_{0,0} &= \{(0,0),\ (1,0),\ (2,0)\} = \{0,\ 3,\ 6\} \\
\mathcal{V}_{0,1} &= \{(0,1),\ (1,1),\ (2,1)\} = \{1,\ 4,\ 7\} \\
\mathcal{V}_{0,2} &= \{(0,2),\ (1,2),\ (2,2)\} = \{2,\ 5,\ 8\} \\
\mathcal{V}_{1,0} &= \{(0,0),\ (0,1),\ (0,2)\} = \{0,\ 1,\ 2\} \\
\mathcal{V}_{1,1} &= \{(1,0),\ (1,1),\ (1,2)\} = \{3,\ 4,\ 5\} \\
\mathcal{V}_{1,2} &= \{(2,0),\ (2,1),\ (2,2)\} = \{6,\ 7,\ 8\}
\end{aligned}$$

- For each u and v, define

$$\mathcal{E}_{u,v} = \{e_s \mid s \in \mathcal{V}_{u,v}\}$$

and

$$\mathcal{Q}_u = \left\{ \sum_{s_u=0}^{q-1} e_{(s_{m-1},\cdots,s_0)} \;\middle|\; s_j \in \{0,1\}, j \neq u \right\}$$

where the sum is performed under modulo q.

Example ($q = 3$, $m = 2$)

$$\mathcal{E}_{0,0} = \{\mathbf{e}_0,\ \mathbf{e}_3,\ \mathbf{e}_6\} \quad \mathcal{E}_{0,1} = \{\mathbf{e}_1,\ \mathbf{e}_4,\ \mathbf{e}_7\} \quad \mathcal{E}_{0,2} = \{\mathbf{e}_2,\ \mathbf{e}_5,\ \mathbf{e}_8\}$$
$$\mathcal{E}_{1,0} = \{\mathbf{e}_0,\ \mathbf{e}_1,\ \mathbf{e}_2\} \quad \mathcal{E}_{1,1} = \{\mathbf{e}_3,\ \mathbf{e}_4,\ \mathbf{e}_5\} \quad \mathcal{E}_{1,2} = \{\mathbf{e}_6,\ \mathbf{e}_7,\ \mathbf{e}_8\}$$

$$\mathcal{Q}_0 = \{\mathbf{e}_0 + \mathbf{e}_1 + \mathbf{e}_2,\ \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5,\ \mathbf{e}_6 + \mathbf{e}_7 + \mathbf{e}_8\},$$
$$\mathcal{Q}_1 = \{\mathbf{e}_0 + \mathbf{e}_3 + \mathbf{e}_6,\ \mathbf{e}_1 + \mathbf{e}_4 + \mathbf{e}_7,\ \mathbf{e}_2 + \mathbf{e}_5 + \mathbf{e}_8\}.$$

- For each u and v, define

$$\mathcal{E}_{u,v} = \{e_s \mid s \in \mathcal{V}_{u,v}\}$$

and

$$\mathcal{Q}_u = \left\{ \sum_{s_u=0}^{q-1} e_{(s_{m-1},\cdots,s_0)} \;\middle|\; s_j \in \{0,1\}, j \neq u \right\}$$

where the sum is performed under modulo q.

Example ($q = 3$, $m = 2$)

$$\mathcal{E}_{0,0} = \{\mathbf{e}_0,\ \mathbf{e}_3,\ \mathbf{e}_6\} \quad \mathcal{E}_{0,1} = \{\mathbf{e}_1,\ \mathbf{e}_4,\ \mathbf{e}_7\} \quad \mathcal{E}_{0,2} = \{\mathbf{e}_2,\ \mathbf{e}_5,\ \mathbf{e}_8\}$$
$$\mathcal{E}_{1,0} = \{\mathbf{e}_0,\ \mathbf{e}_1,\ \mathbf{e}_2\} \quad \mathcal{E}_{1,1} = \{\mathbf{e}_3,\ \mathbf{e}_4,\ \mathbf{e}_5\} \quad \mathcal{E}_{1,2} = \{\mathbf{e}_6,\ \mathbf{e}_7,\ \mathbf{e}_8\}$$

$$\mathcal{Q}_0 = \{\mathbf{e}_0 + \mathbf{e}_1 + \mathbf{e}_2,\ \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5,\ \mathbf{e}_6 + \mathbf{e}_7 + \mathbf{e}_8\},$$
$$\mathcal{Q}_1 = \{\mathbf{e}_0 + \mathbf{e}_3 + \mathbf{e}_6,\ \mathbf{e}_1 + \mathbf{e}_4 + \mathbf{e}_7,\ \mathbf{e}_2 + \mathbf{e}_5 + \mathbf{e}_8\}.$$

Notations

- $$\phi_{u,v}(s) = \begin{cases} 1 & \text{if } s \in \mathcal{V}_{u,v} \\ 0 & \text{otherwise} \end{cases}$$
$$\varphi_{u,v}(s) = (s_0, \ldots, s_{u-1}, v, s_{u+1}, \ldots, s_{m-1}).$$

- $$\mathbf{C}_{u,v,v'} = \begin{pmatrix} \phi_{u,v}(0)\mathbf{e}_{\varphi_{u,v'}(0)} \\ \phi_{u,v}(1)\mathbf{e}_{\varphi_{u,v'}(1)} \\ \cdots \\ \phi_{u,v}(q^m-1)\mathbf{e}_{\varphi_{u,v'}(q^m-1)} \end{pmatrix} + \begin{pmatrix} \phi_{u,v'}(0)\mathbf{e}_0 \\ \phi_{u,v'}(1)\mathbf{e}_1 \\ \cdots \\ \phi_{u,v'}(q^m-1)\mathbf{e}_{q^m-1} \end{pmatrix}$$

- $$\mathbf{C}_{u,q,v} = \begin{pmatrix} \phi_{u,v}(0)\mathbf{e}_0 \\ \phi_{u,v}(1)\mathbf{e}_1 \\ \cdots \\ \phi_{u,v}(q^m-1)\mathbf{e}_{q^m-1} \end{pmatrix}$$

Notations

- $$\phi_{u,v}(s) = \begin{cases} 1 & \text{if } s \in \mathcal{V}_{u,v} \\ 0 & \text{otherwise} \end{cases}$$
  $$\varphi_{u,v}(s) = (s_0, \ldots, s_{u-1}, v, s_{u+1}, \ldots, s_{m-1}).$$

- $$\mathbf{C}_{u,v,v'} = \begin{pmatrix} \phi_{u,v}(0)\mathbf{e}_{\varphi_{u,v'}(0)} \\ \phi_{u,v}(1)\mathbf{e}_{\varphi_{u,v'}(1)} \\ \ldots \\ \phi_{u,v}(q^m-1)\mathbf{e}_{\varphi_{u,v'}(q^m-1)} \end{pmatrix} + \begin{pmatrix} \phi_{u,v'}(0)\mathbf{e}_0 \\ \phi_{u,v'}(1)\mathbf{e}_1 \\ \ldots \\ \phi_{u,v'}(q^m-1)\mathbf{e}_{q^m-1} \end{pmatrix}$$

- $$\mathbf{C}_{u,q,v} = \begin{pmatrix} \phi_{u,v}(0)\mathbf{e}_0 \\ \phi_{u,v}(1)\mathbf{e}_1 \\ \ldots \\ \phi_{u,v}(q^m-1)\mathbf{e}_{q^m-1} \end{pmatrix}$$

Notations

- $$\phi_{u,v}(s) = \begin{cases} 1 & \text{if } s \in \mathcal{V}_{u,v} \\ 0 & \text{otherwise} \end{cases}$$
  $$\varphi_{u,v}(s) = (s_0, \ldots, s_{u-1}, v, s_{u+1}, \ldots, s_{m-1}).$$

- $$\mathbf{C}_{u,v,v'} = \begin{pmatrix} \phi_{u,v}(0)\mathbf{e}_{\varphi_{u,v'}(0)} \\ \phi_{u,v}(1)\mathbf{e}_{\varphi_{u,v'}(1)} \\ \cdots \\ \phi_{u,v}(q^m-1)\mathbf{e}_{\varphi_{u,v'}(q^m-1)} \end{pmatrix} + \begin{pmatrix} \phi_{u,v'}(0)\mathbf{e}_0 \\ \phi_{u,v'}(1)\mathbf{e}_1 \\ \cdots \\ \phi_{u,v'}(q^m-1)\mathbf{e}_{q^m-1} \end{pmatrix}$$

- $$\mathbf{C}_{u,q,v} = \begin{pmatrix} \phi_{u,v}(0)\mathbf{e}_0 \\ \phi_{u,v}(1)\mathbf{e}_1 \\ \cdots \\ \phi_{u,v}(q^m-1)\mathbf{e}_{q^m-1} \end{pmatrix}$$

Example ($q = 3,\ m = 2$)

$$\mathbf{C}_{0,0,1} = \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_1 \\ 0 \\ \mathbf{e}_4 \\ \mathbf{e}_4 \\ 0 \\ \mathbf{e}_7 \\ \mathbf{e}_7 \\ 0 \end{pmatrix}, \mathbf{C}_{0,0,2} = \begin{pmatrix} \mathbf{e}_2 \\ 0 \\ \mathbf{e}_2 \\ \mathbf{e}_5 \\ 0 \\ \mathbf{e}_5 \\ \mathbf{e}_8 \\ 0 \\ \mathbf{e}_8 \end{pmatrix}, \mathbf{C}_{0,1,0} = \begin{pmatrix} \mathbf{e}_0 \\ \mathbf{e}_0 \\ 0 \\ \mathbf{e}_3 \\ \mathbf{e}_3 \\ 0 \\ \mathbf{e}_6 \\ \mathbf{e}_6 \\ 0 \end{pmatrix}, \mathbf{C}_{1,1,2} = \begin{pmatrix} 0 \\ \mathbf{e}_2 \\ \mathbf{e}_2 \\ 0 \\ \mathbf{e}_5 \\ \mathbf{e}_5 \\ 0 \\ \mathbf{e}_8 \\ \mathbf{e}_8 \end{pmatrix},$$

$$\mathbf{C}_{0,2,0} = \begin{pmatrix} \mathbf{e}_0 \\ 0 \\ \mathbf{e}_0 \\ \mathbf{e}_3 \\ 0 \\ \mathbf{e}_3 \\ \mathbf{e}_6 \\ 0 \\ \mathbf{e}_6 \end{pmatrix}, \mathbf{C}_{0,2,1} = \begin{pmatrix} 0 \\ \mathbf{e}_1 \\ \mathbf{e}_1 \\ 0 \\ \mathbf{e}_4 \\ \mathbf{e}_4 \\ 0 \\ \mathbf{e}_7 \\ \mathbf{e}_7 \end{pmatrix}, \mathbf{C}_{0,3,0} = \begin{pmatrix} \mathbf{e}_0 \\ 0 \\ 0 \\ \mathbf{e}_3 \\ 0 \\ 0 \\ \mathbf{e}_6 \\ 0 \\ 0 \end{pmatrix}, \mathbf{C}_{0,3,1} = \begin{pmatrix} 0 \\ \mathbf{e}_1 \\ 0 \\ 0 \\ \mathbf{e}_4 \\ 0 \\ 0 \\ \mathbf{e}_7 \\ 0 \end{pmatrix}$$

- Caching matrices

$$\mathbf{S}_{u,v} = \mathcal{E}_{u,v}, \mathbf{S}_{u,q} = \mathcal{Q}_u$$

- Coding matrices:

$$\mathbf{A}_{u,v} = \begin{pmatrix} \mathbf{C}_{u,v,v_0} \\ \mathbf{C}_{u,v,v_1} \\ \vdots \\ \mathbf{C}_{u,v,v_{q-1}} \end{pmatrix}, v_i \in [0,q) \setminus \{v\} \text{ and } \mathbf{A}_{u,q} = \begin{pmatrix} \mathbf{C}_{u,q,0} \\ \mathbf{C}_{u,q,1} \\ \vdots \\ \mathbf{C}_{u,q,q-2} \end{pmatrix}$$

- Decoding matrices

$$\mathbf{S}'_{u,v} = \begin{pmatrix} \mathbf{E}_{u,v} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{E}_{u,v} \end{pmatrix}$$

- Caching matrices

$$\mathbf{S}_{u,v} = \mathcal{E}_{u,v}, \mathbf{S}_{u,q} = \mathcal{Q}_u$$

- Coding matrices:

$$\mathbf{A}_{u,v} = \begin{pmatrix} \mathbf{C}_{u,v,v_0} \\ \mathbf{C}_{u,v,v_1} \\ \vdots \\ \mathbf{C}_{u,v,v_{q-1}} \end{pmatrix}, v_i \in [0,q) \setminus \{v\} \text{ and } \mathbf{A}_{u,q} = \begin{pmatrix} \mathbf{C}_{u,q,0} \\ \mathbf{C}_{u,q,1} \\ \vdots \\ \mathbf{C}_{u,q,q-2} \end{pmatrix}$$

- Decoding matrices

$$\mathbf{S}'_{u,v} = \begin{pmatrix} \mathbf{E}_{u,v} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{E}_{u,v} \end{pmatrix}$$

- Caching matrices

$$\mathbf{S}_{u,v} = \mathcal{E}_{u,v}, \mathbf{S}_{u,q} = \mathcal{Q}_u$$

- Coding matrices:

$$\mathbf{A}_{u,v} = \begin{pmatrix} \mathbf{C}_{u,v,v_0} \\ \mathbf{C}_{u,v,v_1} \\ \vdots \\ \mathbf{C}_{u,v,v_{q-1}} \end{pmatrix}, v_i \in [0,q) \setminus \{v\} \text{ and } \mathbf{A}_{u,q} = \begin{pmatrix} \mathbf{C}_{u,q,0} \\ \mathbf{C}_{u,q,1} \\ \vdots \\ \mathbf{C}_{u,q,q-2} \end{pmatrix}$$

- Decoding matrices

$$\mathbf{S}'_{u,v} = \begin{pmatrix} \mathbf{E}_{u,v} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{E}_{u,v} \end{pmatrix}$$

Table: Comparisons

results	M/N	R	F
MN scheme	$\frac{1}{q}$	$\frac{K(q-1)}{q+K}$	$\sim \frac{q}{\sqrt{2\pi K(q-1)}} \cdot e^{\frac{K}{q}\left(\ln q+(q-1)\ln\frac{q}{q-1}\right)}$
Known scheme	$\frac{1}{q}$	$q-1$	$e^{(\frac{K}{q}-1)\ln q}$
New Scheme	$\frac{1}{q}$	$q-1$	$e^{\frac{K}{q+1}\ln q}$

Theorem

For any positive integers q, z, m with $q \geq 2$ and $z < q$,there exists a coded caching scheme with parameters

$$K = m(q+1)\left\lfloor \frac{q-1}{q-z} \right\rfloor, \quad \frac{M}{N} = \frac{z}{q}, \quad R = q - z, \quad F = q^m.$$

The operation is over the finite filed $\mathbb{F}_2$.

- We proposed a new viewpoint to study coded caching scheme and constructed a new class of coded caching schemes over $\mathbb{F}_2$.

- For the fixed parameters K, $\frac{M}{N}$ and R, could we further reduce the value of F by increasing the size of filed?

- We proposed a new viewpoint to study coded caching scheme and constructed a new class of coded caching schemes over $\mathbb{F}_2$.

- For the fixed parameters K, $\frac{M}{N}$ and R, could we further reduce the value of F by increasing the size of filed?

Thank You!!!