Towards an Algebraic Network Information Theory: Part II. Simultaneous Decoding

Bobak Nazer
BU

Sung Hoon Lim — KIOST
Chen Feng — UBC
Adriano Pastore — CTTC
Michael Gastpar — EPFL

CISS 2018
March 23, 2018

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty-paper coding, and interference alignment.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty-paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty-paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty-paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.
- Most of the initial efforts have focused on Gaussian networks.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

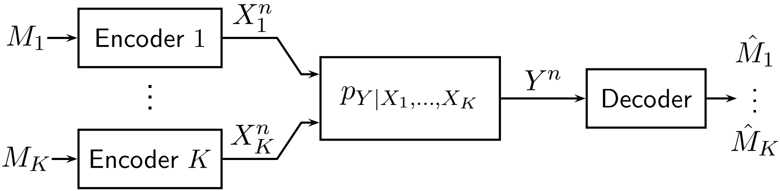
Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty-paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.
- Most of the initial efforts have focused on Gaussian networks.
- Are these just a collection of intriguing examples or elements of a more general theory?

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

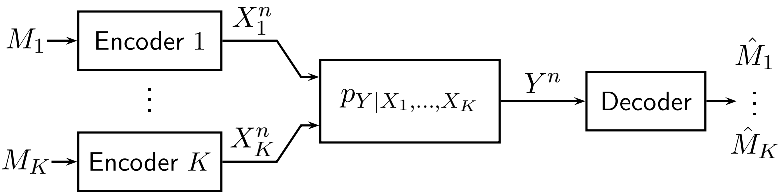
Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty-paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.
- Most of the initial efforts have focused on Gaussian networks.
- Are these just a collection of intriguing examples or elements of a more general theory?
- Recent efforts, starting with **Padakandla-Pradhan '13**, demonstrate that nested linear codes can be brought into the powerful framework of joint typicality encoding and decoding.



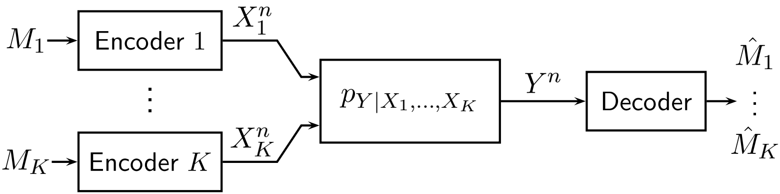
Problem Statement:

- Transmitter k has a message $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$



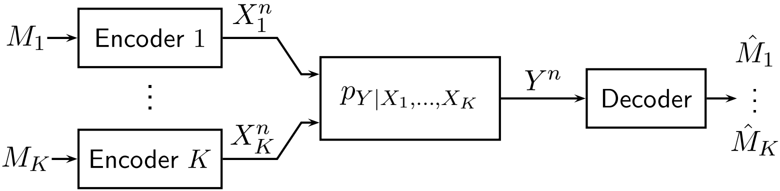
Problem Statement:

- Transmitter k has a message $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$



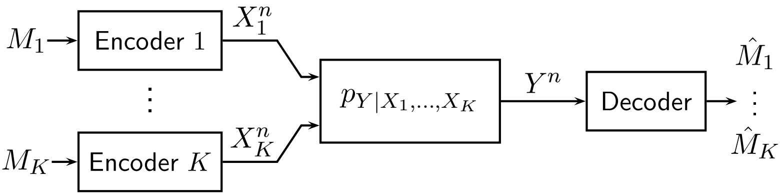
Problem Statement:

- Transmitter k has a message $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$
- R_k is the rate (in bits/channel use)



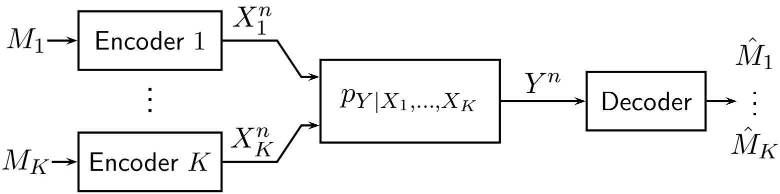
Problem Statement:

- Transmitter k has a message $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$
- R_k is the rate (in bits/channel use)
- Encoder k: assigns codeword $x_k^n(m_k) \in \mathcal{X}_k^n$ to each $m_k \in [2^{nR_k}]$



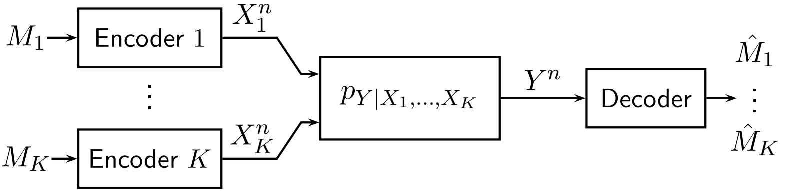
Problem Statement:

- Transmitter k has a message $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$
- R_k is the rate (in bits/channel use)
- Encoder k: assigns codeword $x_k^n(m_k) \in \mathcal{X}_k^n$ to each $m_k \in [2^{nR_k}]$
- Memoryless Channel: $p_{Y^n|X_1^n,X_2^n}(y^n|x_1^n, x_2^n) = \prod_{i=1}^{n} p_{Y|X_1,X_2}(y_i|x_{1,i}x_{2,i})$



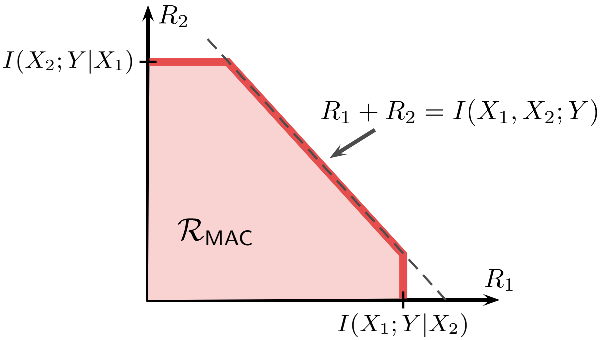
Problem Statement:

- Transmitter k has a message $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$
- R_k is the rate (in bits/channel use)
- Encoder k: assigns codeword $x_k^n(m_k) \in \mathcal{X}_k^n$ to each $m_k \in [2^{nR_k}]$
- Memoryless Channel: $p_{Y^n|X_1^n,X_2^n}(y^n|x_1^n, x_2^n) = \prod_{i=1}^{n} p_{Y|X_1,X_2}(y_i|x_{1,i}x_{2,i})$
- Decoder: assigns estimates $(\hat{m}_1, \hat{m}_2)$ to each $y^n \in \mathcal{Y}^n$



Problem Statement:

- Transmitter k has a message $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$
- R_k is the rate (in bits/channel use)
- Encoder k: assigns codeword $x_k^n(m_k) \in \mathcal{X}_k^n$ to each $m_k \in [2^{nR_k}]$
- Memoryless Channel: $p_{Y^n|X_1^n,X_2^n}(y^n|x_1^n, x_2^n) = \prod_{i=1}^{n} p_{Y|X_1,X_2}(y_i|x_{1,i}x_{2,i})$
- Decoder: assigns estimates $(\hat{m}_1, \hat{m}_2)$ to each $y^n \in \mathcal{Y}^n$
- Average probability of error is $\mathsf{P}\{(\hat{M}_1, \ldots, \hat{M}_K) \neq (M_1, \ldots, M_K)\}$ where $M_1, \ldots, M_K$ are drawn independently and uniformly.



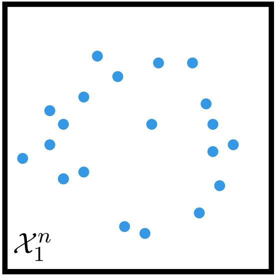
Theorem (Ahlswede '71, Liao '72)

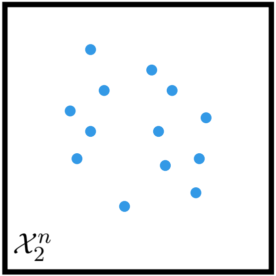
The multiple-access capacity region is the convex closure of all rate pairs (R_1, R_2) *satisfying*

$$R_1 < I(X_1;Y|X_2) \qquad R_2 < I(X_2;Y|X_1) \qquad R_1 + R_2 < I(X_1, X_2; Y)$$

for some $p_{X_1}(x_1)p_{X_2}(x_2)$.

MAC Achievability via I.I.D. Random Coding

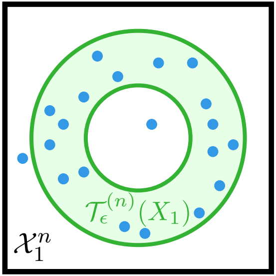


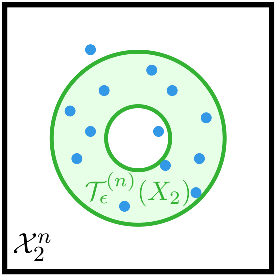


Code Construction:

- For each message $m_1 \in [2^{nR_1}]$, generate codeword $X_1^n(m_1)$ i.i.d. according to $p_{X_1}(x_1)$.
- For each message $m_2 \in [2^{nR_2}]$, generate codeword $X_2^n(m_2)$ i.i.d. according to $p_{X_2}(x_2)$.

MAC Achievability via I.I.D. Random Coding

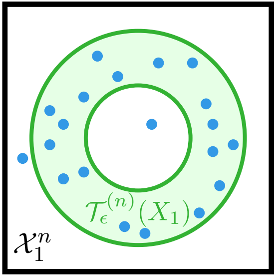


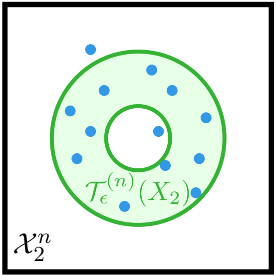


Code Construction:

- For each message $m_1 \in [2^{nR_1}]$, generate codeword $X_1^n(m_1)$ i.i.d. according to $p_{X_1}(x_1)$.
- For each message $m_2 \in [2^{nR_2}]$, generate codeword $X_2^n(m_2)$ i.i.d. according to $p_{X_2}(x_2)$.
- With high probability, codewords are typical.

MAC Achievability via I.I.D. Random Coding



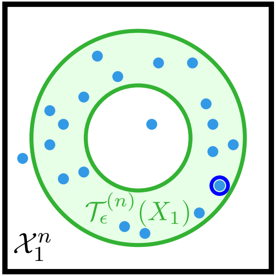


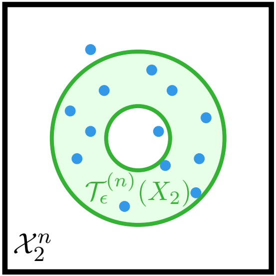
Code Construction:

- For each message $m_1 \in [2^{nR_1}]$, generate codeword $X_1^n(m_1)$ i.i.d. according to $p_{X_1}(x_1)$.
- For each message $m_2 \in [2^{nR_2}]$, generate codeword $X_2^n(m_2)$ i.i.d. according to $p_{X_2}(x_2)$.
- With high probability, codewords are typical.

Encoding:

MAC Achievability via I.I.D. Random Coding





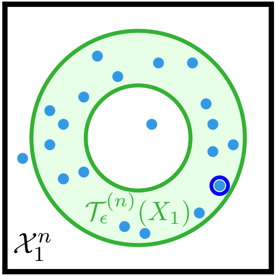
Code Construction:

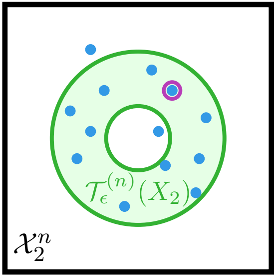
- For each message $m_1 \in [2^{nR_1}]$, generate codeword $X_1^n(m_1)$ i.i.d. according to $p_{X_1}(x_1)$.
- For each message $m_2 \in [2^{nR_2}]$, generate codeword $X_2^n(m_2)$ i.i.d. according to $p_{X_2}(x_2)$.
- With high probability, codewords are typical.

Encoding:

- User 1: Transmit $X_1^n(m_1)$.

MAC Achievability via I.I.D. Random Coding





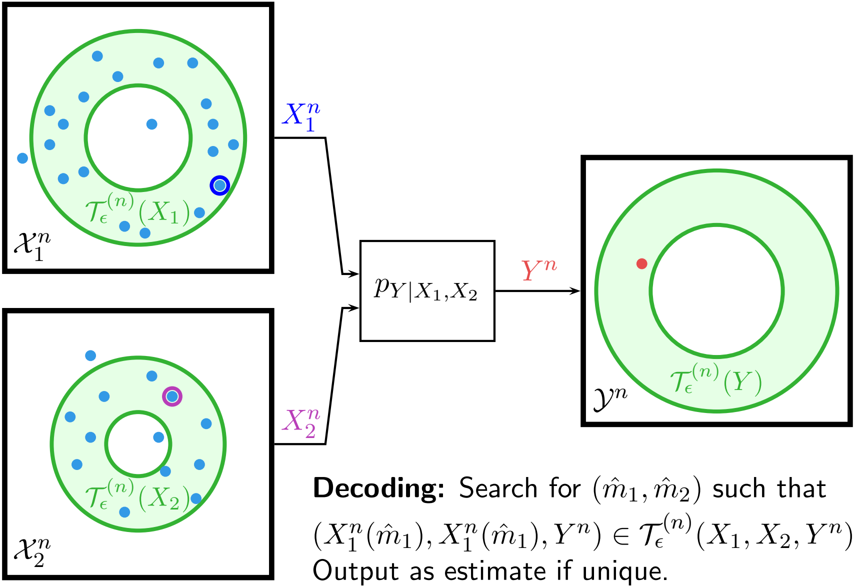
Code Construction:

- For each message $m_1 \in [2^{nR_1}]$, generate codeword $X_1^n(m_1)$ i.i.d. according to $p_{X_1}(x_1)$.
- For each message $m_2 \in [2^{nR_2}]$, generate codeword $X_2^n(m_2)$ i.i.d. according to $p_{X_2}(x_2)$.
- With high probability, codewords are typical.

Encoding:

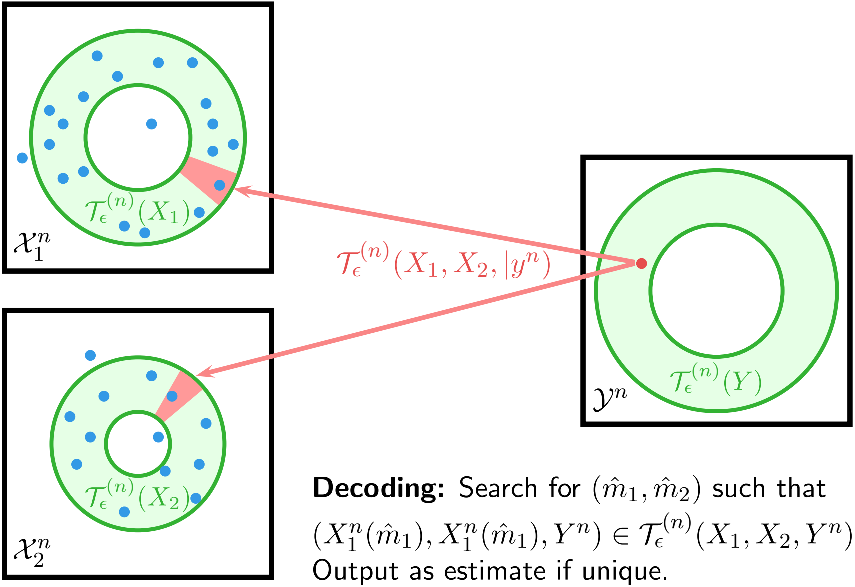
- User 1: Transmit $X_1^n(m_1)$.
- User 2: Transmit $X_2^n(m_2)$.

MAC Achievability via I.I.D. Random Coding



Decoding: Search for $(\hat{m}_1, \hat{m}_2)$ such that $(X_1^n(\hat{m}_1), X_1^n(\hat{m}_1), Y^n) \in \mathcal{T}_\epsilon^{(n)}(X_1, X_2, Y^n)$.
Output as estimate if unique.
Otherwise, declare an error.

MAC Achievability via I.I.D. Random Coding



Decoding: Search for $(\hat{m}_1, \hat{m}_2)$ such that
$(X_1^n(\hat{m}_1), X_1^n(\hat{m}_1), Y^n) \in \mathcal{T}_\epsilon^{(n)}(X_1, X_2, Y^n)$.
Output as estimate if unique.
Otherwise, declare an error.

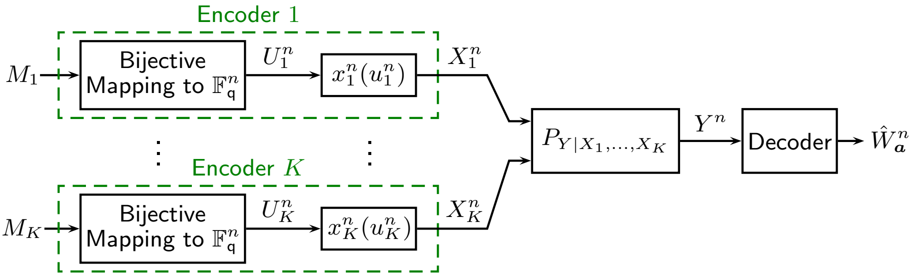
MAC Achievability via I.I.D. Random Coding

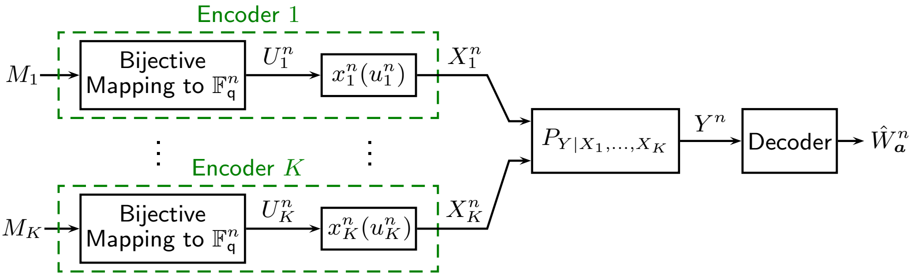
Error Analysis: Assume $m_1 = 0$, $m_2 = 0$ are selected messages.

$$\mathcal{E}_1 = \left\{(X_1^n(0), X_2^n(0), Y^n) \notin \mathcal{T}_\epsilon^{(n)}(X_1, X_2, Y)\right\}$$
$$\mathcal{E}_2 = \left\{(X_1^n(m_1), X_2^n(0), Y^n) \notin \mathcal{T}_\epsilon^{(n)}(X_1, X_2, Y) \text{ for some } m_1 \neq 0\right\}$$
$$\mathcal{E}_3 = \left\{(X_1^n(0), X_2^n(m_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}(X_1, X_2, Y) \text{ for some } m_2 \neq 0\right\}$$
$$\mathcal{E}_4 = \left\{(X_1^n(m_1), X_2^n(m_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}(X_1, X_2, Y) \text{ for some } m_1 \neq 0, m_2 \neq 0\right\}$$

- By the Weak Law of Large Numbers, $\mathsf{P}\{\mathcal{E}_1\} \to 0$.
- By the Packing Lemma, $\mathsf{P}\{\mathcal{E}_2\} \to 0$ if $R_1 < I(X_1; Y|X_2) - \delta(\epsilon)$.
- By the Packing Lemma, $\mathsf{P}\{\mathcal{E}_3\} \to 0$ if $R_2 < I(X_2; Y|X_1) - \delta(\epsilon)$.
- By the Packing Lemma, $\mathsf{P}\{\mathcal{E}_4\} \to 0$ if $R_1 + R_2 < I(X_1, X_2; Y) - \delta(\epsilon)$.

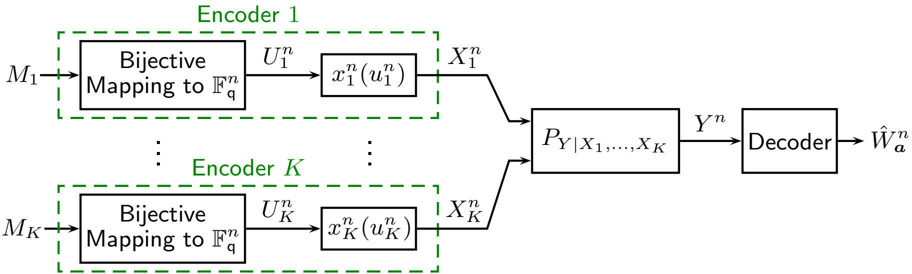
Compute–Forward





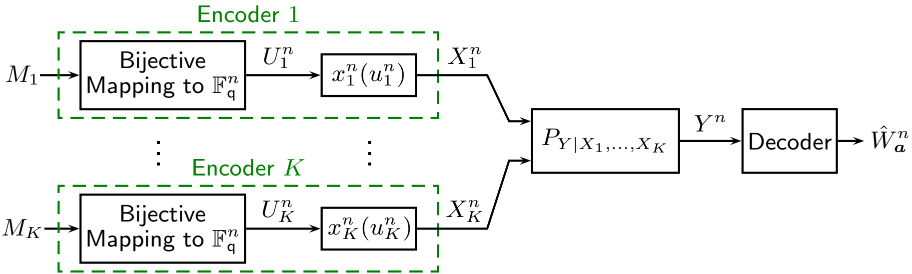
Problem Statement:

- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.



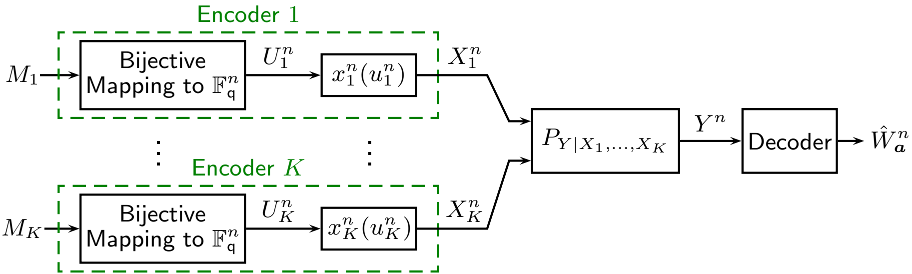
Problem Statement:

- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_q^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is bijective.



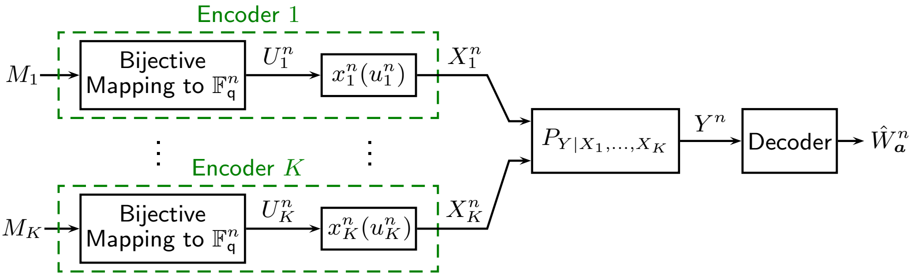
Problem Statement:

- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \dots, 2^{nR_k} - 1\}$, $k = 1, \dots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_q^n \times \mathcal{X}_k^n$, $k = 1, \dots, K$ such that $u_k^n(m_k)$ is bijective.
- Linear Combination: $w_{\boldsymbol{a}}^n \triangleq \bigoplus_k a_k u_k^n(m_k)$, $\boldsymbol{a} = [a_1 \ \cdots \ a_K] \in \mathbb{F}_q^K$



Problem Statement:

- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_q^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is bijective.
- Linear Combination: $w_{\boldsymbol{a}}^n \triangleq \bigoplus_k a_k u_k^n(m_k)$, $\boldsymbol{a} = [a_1 \ \cdots \ a_K] \in \mathbb{F}_q^K$
- Decoder: assigns an estimate $\hat{w}_{\boldsymbol{a}}^n \in \mathbb{F}_q^n$ to each $y^n \in \mathcal{Y}^n$.



Problem Statement:

- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_q^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is bijective.
- Linear Combination: $w_{\boldsymbol{a}}^n \triangleq \bigoplus_k a_k u_k^n(m_k)$, $\boldsymbol{a} = [a_1 \ \cdots \ a_K] \in \mathbb{F}_q^K$
- Decoder: assigns an estimate $\hat{w}_{\boldsymbol{a}}^n \in \mathbb{F}_q^n$ to each $y^n \in \mathcal{Y}^n$.
- Probability of Error: For uniformly distributed messages $M_1, \ldots, M_K$, want $\mathsf{P}\{\hat{W}_{\boldsymbol{a}}^n \neq W_{\boldsymbol{a}}^n\} \to 0$.

Theorem (Lim-Feng-Pastore-Nazer-Gastpar arXiv '16, ISIT '17)

Consider the case of $K = 2$ transmitters and a receiver that wants to recover a linear combination with coefficient vector $\boldsymbol{a} \in \mathbb{F}_{\mathsf{q}}^2$.
A rate pair is achievable if it is included in $\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a}) \cup \mathcal{R}_{\mathsf{LMAC}}$ for some pmfs $p_{U_1}(u_1)$, $p_{U_2}(u_2)$, symbol mappings $x_1(u_1)$, $x_2(u_2)$ where

Theorem (Lim-Feng-Pastore-Nazer-Gastpar arXiv '16, ISIT '17)

Consider the case of $K = 2$ transmitters and a receiver that wants to recover a linear combination with coefficient vector $\boldsymbol{a} \in \mathbb{F}_{\mathsf{q}}^2$.
A rate pair is achievable if it is included in $\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a}) \cup \mathcal{R}_{\mathsf{LMAC}}$ for some pmfs $p_{U_1}(u_1)$, $p_{U_2}(u_2)$, symbol mappings $x_1(u_1)$, $x_2(u_2)$ where

$$\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a}) \triangleq \left\{(R_1, R_2) : R_k < I_{\mathsf{CF},k}(\boldsymbol{a}) \triangleq H(U_k) - H(W_{\boldsymbol{a}}|Y),\ k = 1, 2\right\}$$

Two-User Compute–Forward

Theorem (Lim-Feng-Pastore-Nazer-Gastpar arXiv '16, ISIT '17)

Consider the case of $K = 2$ transmitters and a receiver that wants to recover a linear combination with coefficient vector $\boldsymbol{a} \in \mathbb{F}_{\mathsf{q}}^2$.
A rate pair is achievable if it is included in $\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a}) \cup \mathcal{R}_{\mathsf{LMAC}}$ for some pmfs $p_{U_1}(u_1)$, $p_{U_2}(u_2)$, symbol mappings $x_1(u_1)$, $x_2(u_2)$ where

$$\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a}) \triangleq \Big\{(R_1, R_2) : R_k < I_{\mathsf{CF},k}(\boldsymbol{a}) \triangleq H(U_k) - H(W_{\boldsymbol{a}}|Y),\ k = 1, 2\Big\}$$

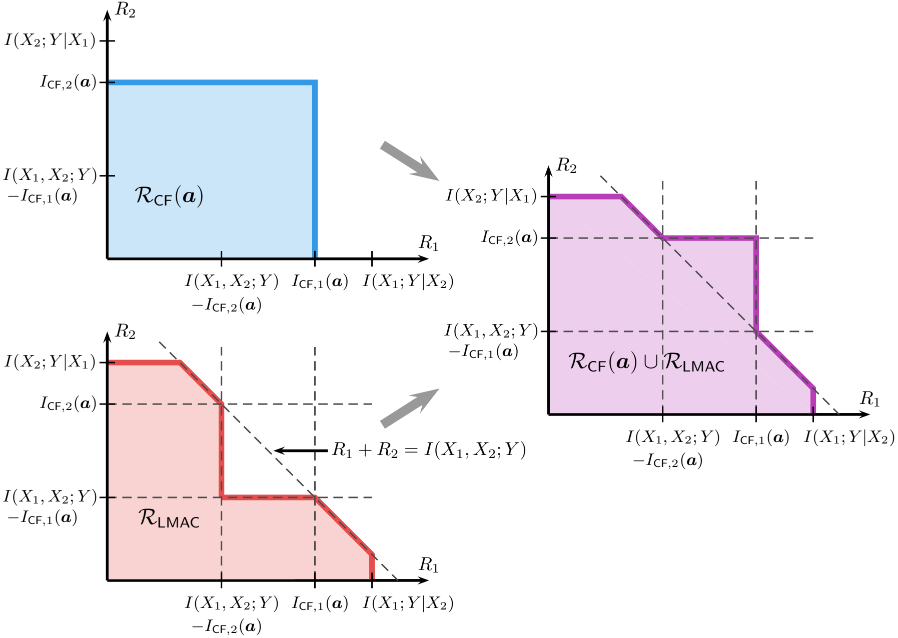
$$\mathcal{R}_{\mathsf{LMAC}} \triangleq \Big\{(R_1, R_2) : \max\{R_1, R_2\} < \min_{\boldsymbol{b} \in \mathbb{F}_{\mathsf{q}}^2 : b_k \neq 0} I(U_k; Y, W_{\boldsymbol{b}})$$

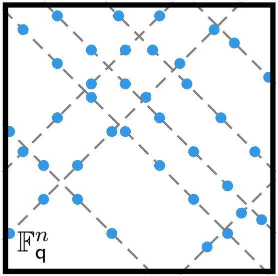
$$R_1 < I(U_1; Y|U_2),$$

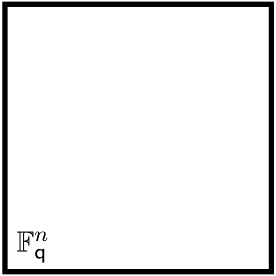
$$R_2 < I(U_2; Y|U_1),$$

$$R_1 + R_2 < I(U_1, U_2; Y)\Big\}$$

Two-User Compute–Forward



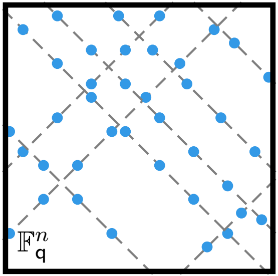


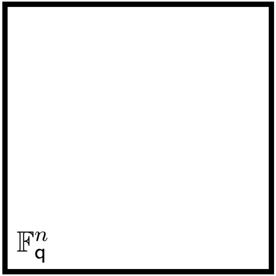


Code Construction:

- q-ary expansion $\mathbf{m}_k$ of message $m_k \in [2^{nR_k}]$.

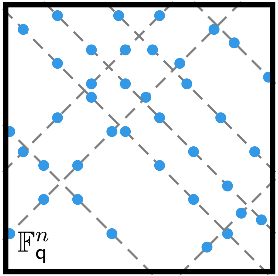
Compute–Forward Achievability via Linear Random Coding

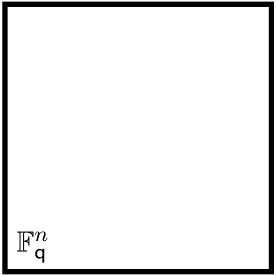




Code Construction:

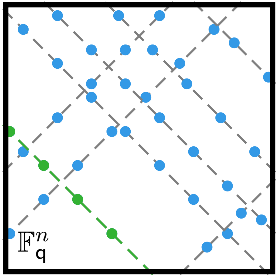
- q-ary expansion $\mathbf{m}_k$ of message $m_k \in [2^{nR_k}]$.
- Auxiliary index $l_k \in [2^{n\hat{R}_k}]$ with q-ary expansions $\mathbf{l}_k$.

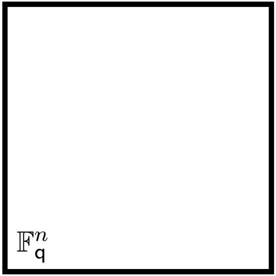




Code Construction:

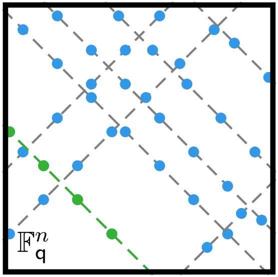
- q-ary expansion $\mathbf{m}_k$ of message $m_k \in [2^{nR_k}]$.
- Auxiliary index $l_k \in [2^{n\hat{R}_k}]$ with q-ary expansions $\mathbf{l}_k$.
- Draw generator matrix $\mathsf{G} \in \mathbb{F}_{\mathsf{q}}^{\kappa \times n}$ and dithers $\mathsf{d}_1^n, \mathsf{d}_2^n \in \mathbb{F}_{\mathsf{q}}^n$ i.i.d. Unif$(\mathbb{F}_{\mathsf{q}})$ where $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\}) / \log(\mathsf{q})$.

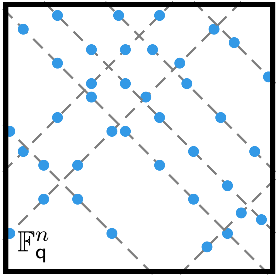




Code Construction:

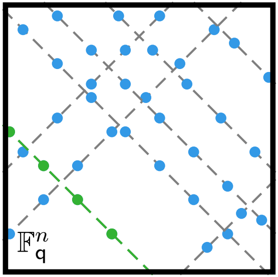
- q-ary expansion $\mathbf{m}_k$ of message $m_k \in [2^{nR_k}]$.
- Auxiliary index $l_k \in [2^{n\hat{R}_k}]$ with q-ary expansions $\mathbf{l}_k$.
- Draw generator matrix $\mathsf{G} \in \mathbb{F}_\mathsf{q}^{\kappa \times n}$ and dithers $\mathsf{d}_1^n, \mathsf{d}_2^n \in \mathbb{F}_\mathsf{q}^n$ i.i.d. Unif$(\mathbb{F}_\mathsf{q})$ where $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.
- Linear codewords: $u_1^n(m_1, l_1) = [\mathbf{m}_1\ \mathbf{l}_1]\mathsf{G} \oplus d_1^n$

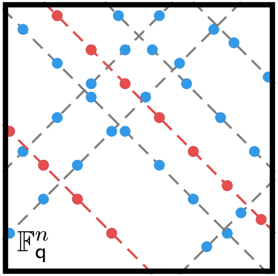




Code Construction:

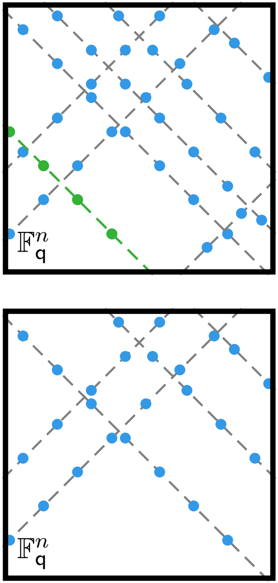
- q-ary expansion $\mathbf{m}_k$ of message $m_k \in [2^{nR_k}]$.
- Auxiliary index $l_k \in [2^{n\hat{R}_k}]$ with q-ary expansions $\mathbf{l}_k$.
- Draw generator matrix $\mathsf{G} \in \mathbb{F}_q^{\kappa \times n}$ and dithers $\mathsf{d}_1^n, \mathsf{d}_2^n \in \mathbb{F}_q^n$ i.i.d. Unif$(\mathbb{F}_q)$ where $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\}) / \log(q)$.
- Linear codewords:
  $u_1^n(m_1, l_1) = [\mathbf{m}_1\ \mathbf{l}_1]\mathsf{G} \oplus d_1^n$
  $u_2^n(m_2, l_2) = [\mathbf{m}_2\ \mathbf{l}_2\ \mathbf{0}]\mathsf{G} \oplus d_2^n$





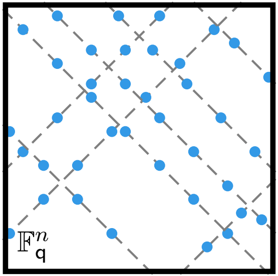
Code Construction:

- q-ary expansion $\mathbf{m}_k$ of message $m_k \in [2^{nR_k}]$.
- Auxiliary index $l_k \in [2^{n\hat{R}_k}]$ with q-ary expansions $\mathbf{l}_k$.
- Draw generator matrix $\mathsf{G} \in \mathbb{F}_q^{\kappa \times n}$ and dithers $\mathsf{d}_1^n, \mathsf{d}_2^n \in \mathbb{F}_q^n$ i.i.d. Unif($\mathbb{F}_q$) where $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.
- Linear codewords:
  $u_1^n(m_1, l_1) = [\mathbf{m}_1\ \mathbf{l}_1]\mathsf{G} \oplus d_1^n$
  $u_2^n(m_2, l_2) = [\mathbf{m}_2\ \mathbf{l}_2\ \mathbf{0}]\mathsf{G} \oplus d_2^n$

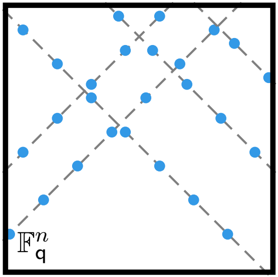


Code Construction:

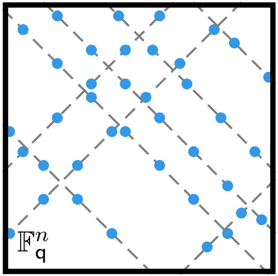
- q-ary expansion $\mathbf{m}_k$ of message $m_k \in [2^{nR_k}]$.
- Auxiliary index $l_k \in [2^{n\hat{R}_k}]$ with q-ary expansions $\mathbf{l}_k$.
- Draw generator matrix $\mathsf{G} \in \mathbb{F}_q^{\kappa \times n}$ and dithers $\mathsf{d}_1^n, \mathsf{d}_2^n \in \mathbb{F}_q^n$ i.i.d. $\mathrm{Unif}(\mathbb{F}_q)$ where $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.
- Linear codewords:
  $u_1^n(m_1, l_1) = [\mathbf{m}_1\ \mathbf{l}_1]\mathsf{G} \oplus d_1^n$
  $u_2^n(m_2, l_2) = [\mathbf{m}_2\ \mathbf{l}_2\ \mathbf{0}]\mathsf{G} \oplus d_2^n$

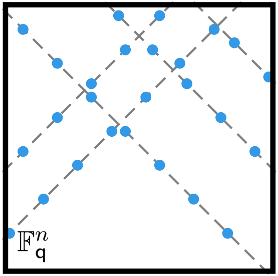


Encoding:



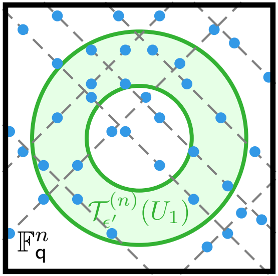
Compute–Forward Achievability via Linear Random Coding

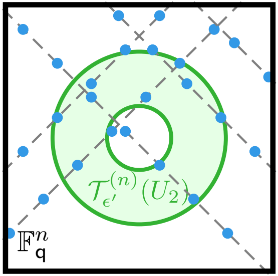




Encoding:

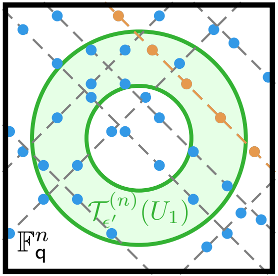
- Fix pmfs $p(u_1)$, $p(u_2)$,
  mappings $x_1(u_1)$, $x_2(u_2)$, and $0 < \epsilon' < \epsilon$.

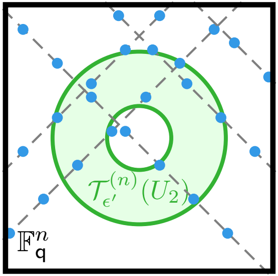




Encoding:

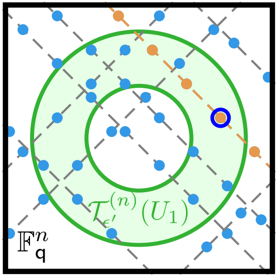
- Fix pmfs $p(u_1)$, $p(u_2)$, mappings $x_1(u_1)$, $x_2(u_2)$, and $0 < \epsilon' < \epsilon$.
- Multicoding:

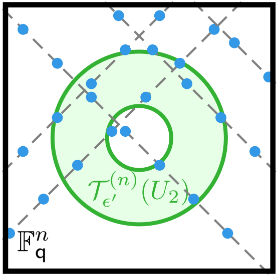




Encoding:

- Fix pmfs $p(u_1)$, $p(u_2)$, mappings $x_1(u_1)$, $x_2(u_2)$, and $0 < \epsilon' < \epsilon$.
- Multicoding: For message m_k,

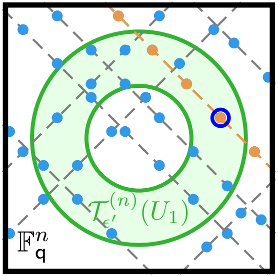


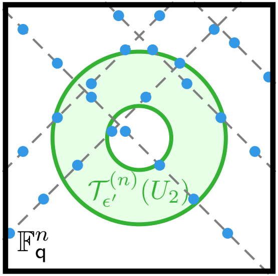


Encoding:

- Fix pmfs $p(u_1)$, $p(u_2)$, mappings $x_1(u_1)$, $x_2(u_2)$, and $0 < \epsilon' < \epsilon$.
- Multicoding: For message m_k, find index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$. (If no such l_k, pick one randomly.)

Compute–Forward Achievability via Linear Random Coding



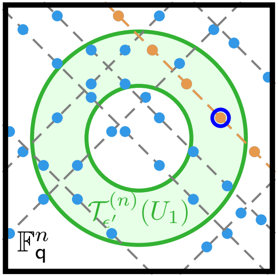


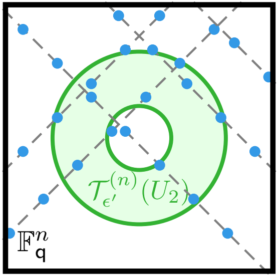
Encoding:

- Fix pmfs $p(u_1)$, $p(u_2)$, mappings $x_1(u_1)$, $x_2(u_2)$, and $0 < \epsilon' < \epsilon$.
- Multicoding: For message m_k, find index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$. (If no such l_k, pick one randomly.)
- Succeeds w.h.p. if

$$\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon')$$

by Mismatched Covering Lemma where $p_{\mathsf{q}} = \mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$.





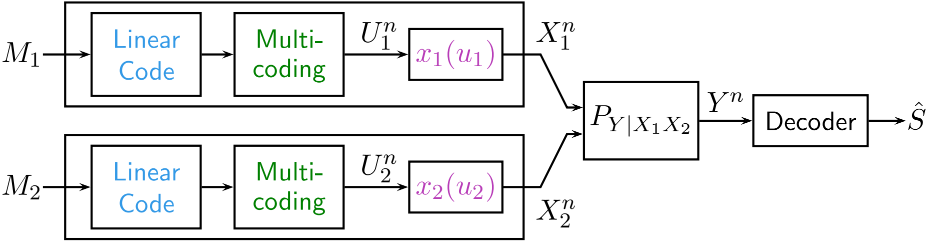
Encoding:

- Fix pmfs $p(u_1)$, $p(u_2)$, mappings $x_1(u_1)$, $x_2(u_2)$, and $0 < \epsilon' < \epsilon$.
- Multicoding: For message m_k, find index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$. (If no such l_k, pick one randomly.)
- Succeeds w.h.p. if

$$\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon')$$

by Mismatched Covering Lemma where $p_{\mathsf{q}} = \mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$.

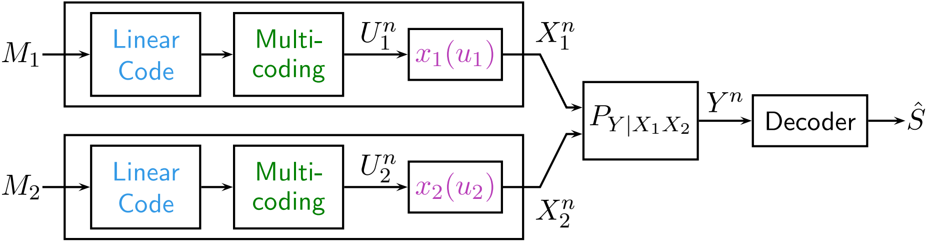
- At time i, transmit $x_{ki} = x_k\big(u_{ki}(m_k, l_k)\big)$.



- For $m_k \in [2^{nR_k}]$, $l_k \in [2^{n\hat{R}_k}]$, we can express the desired linear combination of codewords as

$$w_{\boldsymbol{a}}^n = a_1 u_1^n(m_1, l_1) \oplus a_2 u_2^n(m_2, l_2)$$

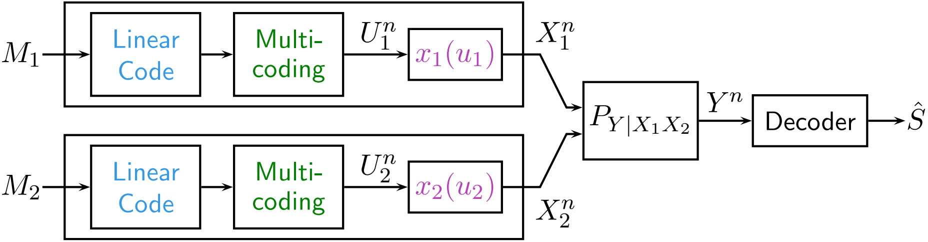
Compute–Forward Achievability via Linear Random Coding



- For $m_k \in [2^{nR_k}]$, $l_k \in [2^{n\hat{R}_k}]$, we can express the desired linear combination of codewords as

$$\begin{aligned} w_{\boldsymbol{a}}^n &= a_1 u_1^n(m_1, l_1) \oplus a_2 u_2^n(m_2, l_2) \\ &= \big[\, a_1[\mathbf{m}_1 \;\, \mathbf{l}_1] \oplus a_2[\mathbf{m}_2 \;\, \mathbf{l}_2 \;\, \mathbf{0}] \,\big] \mathsf{G} \oplus a_1 d_1^n \oplus a_2 d_2^n \end{aligned}$$

Compute–Forward Achievability via Linear Random Coding

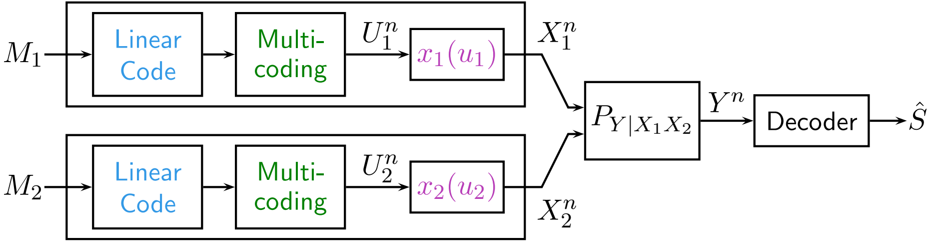


- For $m_k \in [2^{nR_k}]$, $l_k \in [2^{n\hat{R}_k}]$, we can express the desired linear combination of codewords as

$$\begin{aligned} w_{\boldsymbol{a}}^n &= a_1 u_1^n(m_1, l_1) \oplus a_2 u_2^n(m_2, l_2) \\ &= \big[\, a_1[\mathbf{m}_1 \ \mathbf{l}_1] \oplus a_2[\mathbf{m}_2 \ \mathbf{l}_2 \ \mathbf{0}] \,\big]\, \mathsf{G} \oplus a_1 d_1^n \oplus a_2 d_2^n \\ &= \mathbf{s}_{\boldsymbol{a}}\, \mathsf{G} \oplus a_1 d_1^n \oplus a_2 d_2^n \end{aligned}$$

where $s_{\boldsymbol{a}} \in [2^{n \max\{R_1+\hat{R}_1, R_2+\hat{R}_2\}}]$ is the linear combination index corresponding to q-ary expansion $\mathbf{s}_{\boldsymbol{a}}$.

Compute–Forward Achievability via Linear Random Coding

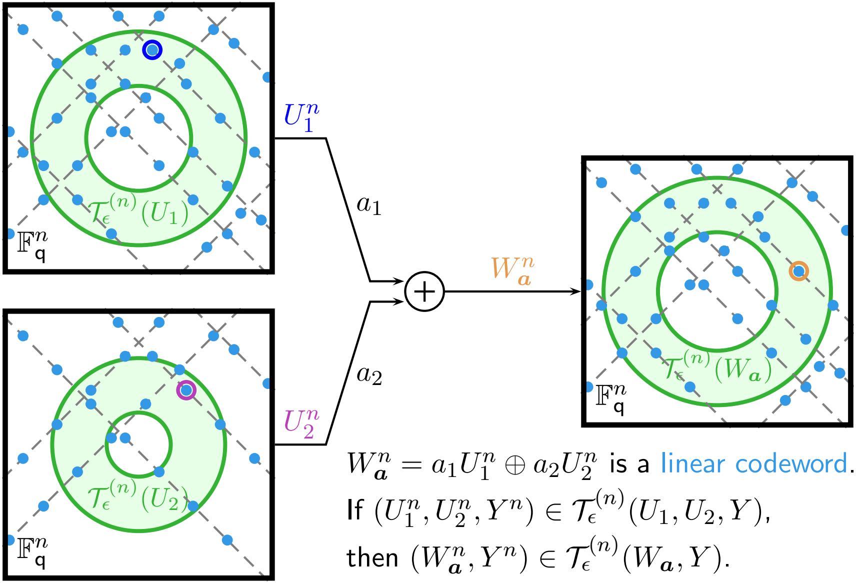


- For $m_k \in [2^{nR_k}]$, $l_k \in [2^{n\hat{R}_k}]$, we can express the desired linear combination of codewords as

$$\begin{aligned} w_{\boldsymbol{a}}^n &= a_1 u_1^n(m_1, l_1) \oplus a_2 u_2^n(m_2, l_2) \\ &= \big[\, a_1[\mathbf{m}_1\ \mathbf{l}_1] \oplus a_2[\mathbf{m}_2\ \mathbf{l}_2\ \mathbf{0}] \,\big] \mathsf{G} \oplus a_1 d_1^n \oplus a_2 d_2^n \\ &= \mathbf{s}_{\boldsymbol{a}}\, \mathsf{G} \oplus a_1 d_1^n \oplus a_2 d_2^n \end{aligned}$$

  where $s_{\boldsymbol{a}} \in \big[2^{n \max\{R_1+\hat{R}_1, R_2+\hat{R}_2\}}\big]$ is the linear combination index corresponding to q-ary expansion $\mathbf{s}_{\boldsymbol{a}}$.
- Can view $w_{\boldsymbol{a}}^n(s)$ as some linear codeword that belongs to $\mathcal{T}_{\epsilon'}^{(n)}(W_{\boldsymbol{a}})$.

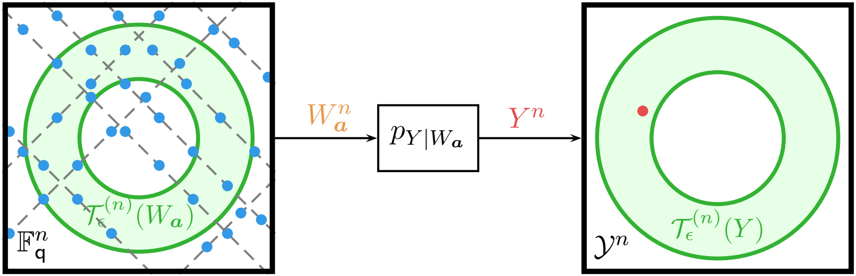
Compute–Forward Achievability via Linear Random Coding



$W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus a_2 U_2^n$ is a linear codeword.

If $(U_1^n, U_2^n, Y^n) \in \mathcal{T}_\epsilon^{(n)}(U_1, U_2, Y)$,

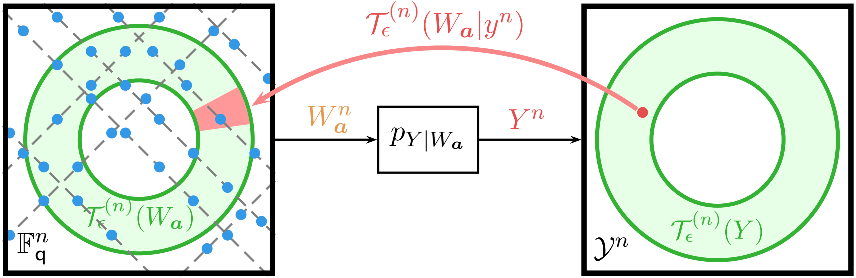
then $(W_{\boldsymbol{a}}^n, Y^n) \in \mathcal{T}_\epsilon^{(n)}(W_{\boldsymbol{a}}, Y)$.



Decoding:

- Search for index $\hat{s}_{\boldsymbol{a}}$ such that $(W_{\boldsymbol{a}}^n(\hat{s}_{\boldsymbol{a}}), Y^n) \in \mathcal{T}_\epsilon^{(n)}(W_{\boldsymbol{a}}, Y)$. Output as estimate if unique. Otherwise, declare an error.

Compute–Forward Achievability via Linear Random Coding



Decoding:

- Search for index $\hat{s}_{\boldsymbol{a}}$ such that $(W_{\boldsymbol{a}}^n(\hat{s}_{\boldsymbol{a}}), Y^n) \in \mathcal{T}_\epsilon^{(n)}(W_{\boldsymbol{a}}, Y)$. Output as estimate if unique. Otherwise, declare an error.
- Although the decoder searches for $W_{\boldsymbol{a}}^n$ over the full linear codebook, it ignores codewords that fall outside the typical set $\mathcal{T}_\epsilon^{(n)}(W_{\boldsymbol{a}})$.

Error Analysis: Assume $s_{\boldsymbol{a}} = 0$ is selected linear combination index.

$$\mathcal{E}_1 = \left\{ U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2 \right\}$$

$$\mathcal{E}_2 = \left\{ (U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)} \right\}$$

$$\mathcal{E}_3 = \left\{ (W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}(W_{\boldsymbol{a}}, Y) \text{ for some } s_{\boldsymbol{a}} \neq 0 \right\}$$

Error Analysis: Assume $s_{\boldsymbol{a}} = 0$ is selected linear combination index.

$$\mathcal{E}_1 = \left\{ U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2 \right\}$$

$$\mathcal{E}_2 = \left\{ (U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)} \right\}$$

$$\mathcal{E}_3 = \left\{ (W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}(W_{\boldsymbol{a}}, Y) \text{ for some } s_{\boldsymbol{a}} \neq 0 \right\}$$

- By the Mismatched Covering Lemma, $\mathsf{P}\{\mathcal{E}_1\} \to 0$ if

$$\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon').$$

Compute–Forward Achievability via Linear Random Coding

Error Analysis: Assume $s_{\boldsymbol{a}} = 0$ is selected linear combination index.

$$\mathcal{E}_1 = \left\{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\right\}$$

$$\mathcal{E}_2 = \left\{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}\right\}$$

$$\mathcal{E}_3 = \left\{(W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}(W_{\boldsymbol{a}}, Y) \text{ for some } s_{\boldsymbol{a}} \neq 0\right\}$$

- By the Mismatched Covering Lemma, $\mathsf{P}\{\mathcal{E}_1\} \to 0$ if

$$\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon').$$

- By the Markov Lemma for Nested Linear Codes, $\mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} \to 0$ if

$$\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon').$$

Subtle Issue: L_1 and L_2 are statistically dependent, since these multicoding indices are chosen with respect to the same linear codebook.

Error Analysis: Assume $s_{\boldsymbol{a}} = 0$ is selected linear combination index.

$$\mathcal{E}_1 = \left\{ U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2 \right\}$$

$$\mathcal{E}_2 = \left\{ (U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)} \right\}$$

$$\mathcal{E}_3 = \left\{ (W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}(W_{\boldsymbol{a}}, Y) \text{ for some } s_{\boldsymbol{a}} \neq 0 \right\}$$

- By the Mismatched Covering Lemma, $\mathsf{P}\{\mathcal{E}_1\} \to 0$ if

$$\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon').$$

- By the Markov Lemma for Nested Linear Codes, $\mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} \to 0$ if

$$\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon').$$

  Subtle Issue: L_1 and L_2 are statistically dependent, since these multicoding indices are chosen with respect to the same linear codebook.

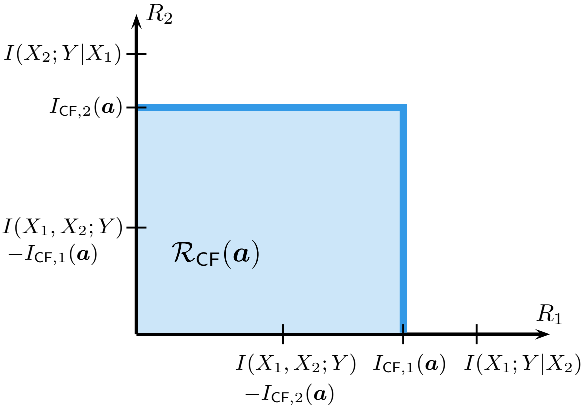
- By the Mismatched Packing Lemma, $\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \to 0$ if

$$R_1 + 2\hat{R}_1 + \hat{R}_2 < I(W_{\boldsymbol{a}}; Y) + D(p_{W_{\boldsymbol{a}}} \| p_{\mathsf{q}}) + D(p_{U_1} \| p_{\mathsf{q}}) + D(p_{U_2} \| p_{\mathsf{q}}) - 2\delta(\epsilon)$$

$$R_2 + \hat{R}_1 + 2\hat{R}_2 < I(W_{\boldsymbol{a}}; Y) + D(p_{W_{\boldsymbol{a}}} \| p_{\mathsf{q}}) + D(p_{U_1} \| p_{\mathsf{q}}) + D(p_{U_2} \| p_{\mathsf{q}}) - 2\delta(\epsilon)$$

- Setting $\hat{R}_k = D(p_{U_k} \| p_{\mathsf{q}}) + 2\delta(\epsilon')$, we find that a rate pair (R_1, R_2) is achievable if

$$R_1 < H(U_1) - H(W_{\boldsymbol{a}}|Y) \qquad R_2 < H(U_2) - H(W_{\boldsymbol{a}}|Y)$$



- What about the “multiple-access” rates, $\mathcal{R}_{\mathsf{LMAC}}$?

- What about the "multiple-access" rates, $\mathcal{R}_{\mathsf{LMAC}}$?
- Decoding $W^n_{\boldsymbol{a}}$ directly does not achieve this rate region.

- What about the "multiple-access" rates, $\mathcal{R}_{\mathsf{LMAC}}$?
- Decoding $W_{\boldsymbol{a}}^n$ directly does not achieve this rate region.
- Instead, we can first decode U_1^n and U_2^n by searching for a unique index tuple (m_1, l_1, m_2, l_2) such that

$$(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)}(U_1, U_2, Y)$$

and afterwards form $W_{\boldsymbol{a}}^n = a_1 U_1^n(m_1, l_1) \oplus a_2 U_2^n(m_2, l_2)$.

- What about the "multiple-access" rates, $\mathcal{R}_{\mathsf{LMAC}}$?
- Decoding $W_{\boldsymbol{a}}^n$ directly does not achieve this rate region.
- Instead, we can first decode U_1^n and U_2^n by searching for a unique index tuple (m_1, l_1, m_2, l_2) such that

$$(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)}(U_1, U_2, Y)$$

  and afterwards form $W_{\boldsymbol{a}}^n = a_1 U_1^n(m_1, l_1) \oplus a_2 U_2^n(m_2, l_2)$.
- Rather than applying two decoders, we can write down a single decoder, inspired by the simultaneous non-unique decoder of **Bandemer-El Gamal-Kim '15.**

Compute–Forward Achievability via Random Linear Codes

- What about the "multiple-access" rates, $\mathcal{R}_{\mathsf{LMAC}}$?
- Decoding $W_{\boldsymbol{a}}^n$ directly does not achieve this rate region.
- Instead, we can first decode U_1^n and U_2^n by searching for a unique index tuple (m_1, l_1, m_2, l_2) such that

$$(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)}(U_1, U_2, Y)$$

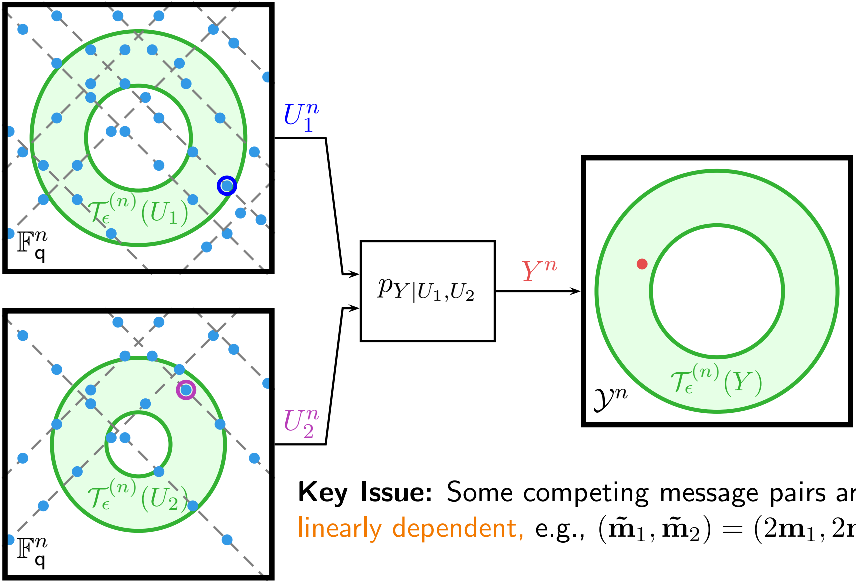
  and afterwards form $W_{\boldsymbol{a}}^n = a_1 U_1^n(m_1, l_1) \oplus a_2 U_2^n(m_2, l_2)$.

- Rather than applying two decoders, we can write down a single decoder, inspired by the simultaneous non-unique decoder of **Bandemer-El Gamal-Kim '15.**
- Specifically, we search for a unique index $s_{\boldsymbol{a}}$ such that, for some index tuple (m_1, l_1, m_2, l_2) whose q-ary expansions satisfy

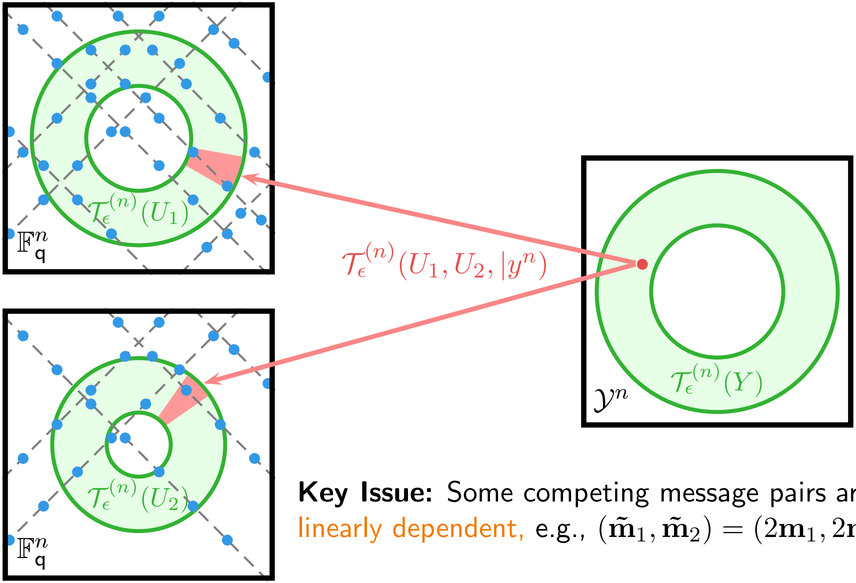
$$\mathbf{s}_{\boldsymbol{a}} = a_1[\mathbf{m}_1\ \mathbf{l}_1] \oplus a_2[\mathbf{m}_2\ \mathbf{l}_2\ \mathbf{0}],$$

  we have that $(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)}(U_1, U_2, Y)$.

LMAC Bound Figure



Key Issue: Some competing message pairs are linearly dependent, e.g., $(\tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2) = (2\mathbf{m}_1, 2\mathbf{m}_2)$.



Key Issue: Some competing message pairs are linearly dependent, e.g., $(\tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2) = (2\mathbf{m}_1, 2\mathbf{m}_2)$.

Error Analysis: Assume index tuple $(m_1, l_1, m_2, l_2) = (0, 0, 0, 0)$ is selected.

$$\begin{aligned}
\mathcal{E}_1 &= \big\{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\big\} \\
\mathcal{E}_2 &= \big\{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}\big\} \\
\mathcal{E}_3 &= \big\{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}(U_1, U_2, Y) \\
&\qquad\qquad \text{for some } (m_1, l_1, m_2, l_2) \neq (0, 0, 0, 0)\big\}
\end{aligned}$$

Error Analysis: Assume index tuple $(m_1, l_1, m_2, l_2) = (0, 0, 0, 0)$ is selected.

$$\begin{aligned}
\mathcal{E}_1 &= \big\{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\big\} \\
\mathcal{E}_2 &= \big\{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}\big\} \\
\mathcal{E}_3 &= \big\{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}(U_1, U_2, Y) \\
&\qquad\qquad \text{for some } (m_1, l_1, m_2, l_2) \neq (0, 0, 0, 0)\big\}
\end{aligned}$$

- We already dealt with $\mathsf{P}\{\mathcal{E}_1\}$ and $\mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\}$.

Error Analysis: Assume index tuple $(m_1, l_1, m_2, l_2) = (0, 0, 0, 0)$ is selected.

$$\begin{aligned}
\mathcal{E}_1 &= \big\{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\big\} \\
\mathcal{E}_2 &= \big\{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}\big\} \\
\mathcal{E}_3 &= \big\{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}(U_1, U_2, Y) \\
&\qquad\qquad\qquad \text{for some } (m_1, l_1, m_2, l_2) \neq (0, 0, 0, 0)\big\}
\end{aligned}$$

- We already dealt with $\mathsf{P}\{\mathcal{E}_1\}$ and $\mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\}$.
- We handle $\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\}$ with the Mismatched Packing Lemma and a careful partitioning of error events to capture linearly dependent competing codewords.

Compute–Forward Achievability via Linear Random Coding

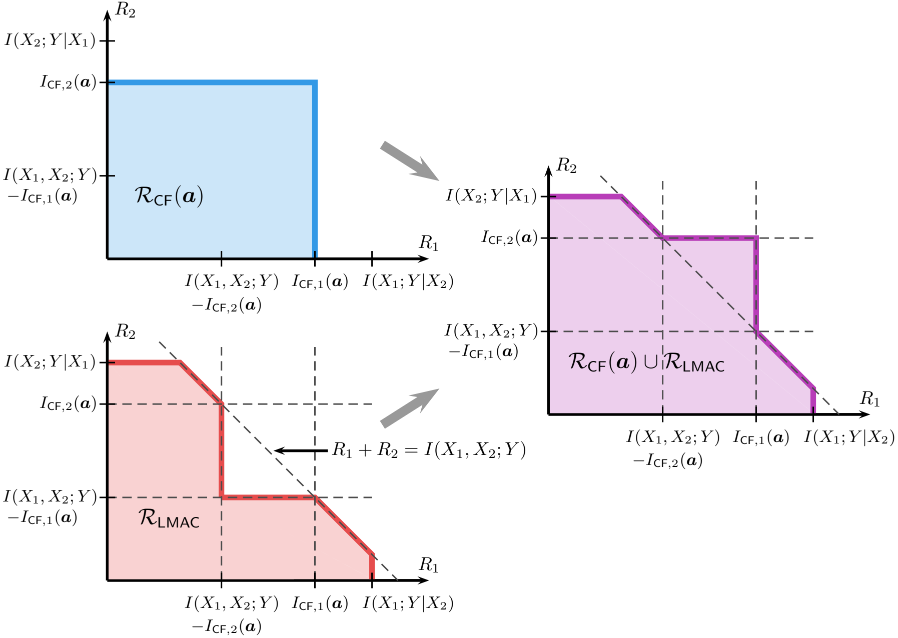
$$
\begin{aligned}
\mathcal{A} &= \{(m_1, l_1, m_2, l_2) : (m_1, l_1, m_2, l_2) \neq (0,0,0,0)\},\\
\mathcal{A}_1 &= \{(m_1, l_1, m_2, l_2) : (m_1, l_1) \neq (0,0), (m_2, l_2) = (0,0)\},\\
\mathcal{A}_2 &= \{(m_1, l_1, m_2, l_2) : (m_1, l_1) = (0,0), (m_2, l_2) \neq (0,0)\},\\
\mathcal{A}_{12} &= \{(m_1, l_1, m_2, l_2) : (m_1, l_1) \neq (0,0), (m_2, l_2) \neq (0,0)\},\\
\mathcal{L} &= \{(m_1, l_1, m_2, l_2) \in \mathcal{A}_{12} : [\mathbf{m}_1\ \mathbf{l_1}], [\mathbf{m}_2\ \mathbf{l}_2\ \mathbf{0}] \text{ are linearly dependent}\},\\
\mathcal{L}^c &= \{(m_1, l_1, m_2, l_2) \in \mathcal{A}_{12} : [\mathbf{m}_1\ \mathbf{l_1}], [\mathbf{m}_2\ \mathbf{l}_2\ \mathbf{0}] \text{ are linearly independent}\}
\end{aligned}
$$

Further, for some $\boldsymbol{b} \in \mathbb{F}_{\mathsf{q}}^2$ such that $\boldsymbol{b} \neq \mathbf{0}$, define

$$
\begin{aligned}
\mathcal{L}_1(\boldsymbol{b}) &= \{(m_1, l_1, m_2, l_2) \in \mathcal{L} : b_1[\mathbf{m}_1\ \mathbf{l_1}] \oplus b_2[\mathbf{m}_2\ \mathbf{l}_2\ \mathbf{0}] \neq \mathbf{0}\},\\
\mathcal{L}_2(\boldsymbol{b}) &= \{(m_1, l_1, m_2, l_2) \in \mathcal{L} : b_1[\mathbf{m}_1\ \mathbf{l_1}] \oplus b_2[\mathbf{m}_2\ \mathbf{l}_2\ \mathbf{0}] = \mathbf{0}\}.
\end{aligned}
$$

Simplifying, we find that any rate $(R_1, R_2) \in \mathcal{R}_{\mathsf{LMAC}}$ is achievable via "multiple-access" decoding.

Rate Region



R_2
$I(X_2;Y|X_1)$
$I_{\mathsf{CF},2}(\boldsymbol{a})$
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},1}(\boldsymbol{a})$
$\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a})$
R_1
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$I_{\mathsf{CF},1}(\boldsymbol{a})$
$I(X_1;Y|X_2)$
$\mathcal{R}_{\mathsf{LMAC}}$
$R_1+R_2=I(X_1,X_2;Y)$
$\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a})\cup\mathcal{R}_{\mathsf{LMAC}}$

- Can we use these discrete memoryless results to recover the Gaussian compute–forward region from **Nazer - Gastpar '11**?

Gaussian Compute–Forward via Discretization

- Can we use these discrete memoryless results to recover the Gaussian compute–forward region from **Nazer - Gastpar '11**?

- Yes! However, the proof requires some new ingredients, since the region is in terms of entropies, rather than mutual informations.

- Can we use these discrete memoryless results to recover the Gaussian compute–forward region from **Nazer - Gastpar '11**?
- Yes! However, the proof requires some new ingredients, since the region is in terms of entropies, rather than mutual informations.
- How about from 2 to K users, i.e., recovering L linear combinations out of K users?

Gaussian Compute–Forward via Discretization

- Can we use these discrete memoryless results to recover the Gaussian compute–forward region from **Nazer - Gastpar '11**?

- Yes! However, the proof requires some new ingredients, since the region is in terms of entropies, rather than mutual informations.

- How about from 2 to K users, i.e., recovering L linear combinations out of K users?

- Yes!

K-User Compute–Forward

- For $\mathsf{A} \in \mathbb{F}_{\mathsf{q}}^{L \times K}$, want to compute

$$W_{\mathsf{A}}^n = \mathsf{A} \begin{bmatrix} U_1^n \\ \vdots \\ U_K^n \end{bmatrix}$$

- For some full rank matrices $\mathsf{B} \in \mathbb{F}_{\mathsf{q}}^{L_{\mathsf{B}} \times K}$, $\mathsf{C} \in \mathbb{F}_{\mathsf{q}}^{L_{\mathsf{C}} \times L_{\mathsf{B}}}$, $0 \leq L_{\mathsf{C}} < L_{\mathsf{B}} \leq K$ (with ranks L_{B} and L_{C}, respectively) and sets $\mathcal{S}, \mathcal{T} \subseteq [1:K]$, define $\mathscr{R}_{\mathrm{D}}(\mathsf{B}, \mathsf{C}, \mathcal{S}, \mathcal{T})$ as the set of rate tuples satisfying the inequality

$$\sum_{k \in \mathcal{T}} R_k < H(U(\mathcal{T})) - H(W_{\mathsf{B}(\mathcal{S})} | Y, W_{\mathsf{CB}}).$$

where $W_{\mathsf{B}} = \mathsf{B}[U_1, \ldots, U_K]^T$.

Theorem

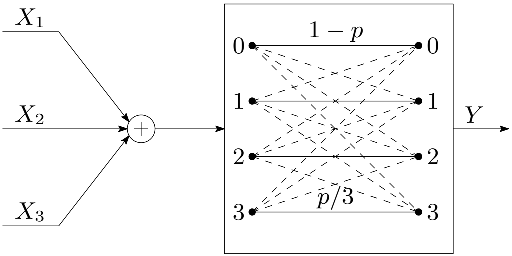
A rate tuple $(R_1, \ldots, R_K)$ is achievable for computing the A*-linear combinations if it is contained in*

$$\bigcup_{\mathsf{B}} \bigcap_{\mathsf{C}} \bigcup_{\mathcal{S}} \bigcap_{\mathcal{T}} \mathscr{R}_{\mathrm{D}}(\mathsf{B}, \mathsf{C}, \mathcal{S}, \mathcal{T})$$

for some $\prod_{k=1}^{K} p(u_k)$ *and mappings* $x_k(u_k)$, $k \in [1:K]$. *The set operations are over all tuples* $(\mathsf{B}, \mathsf{C}, \mathcal{S}, \mathcal{T})$ *with the following constraints:*

1. $\mathsf{B} \in \mathbb{F}_{\mathsf{q}}^{L_{\mathsf{B}} \times K}$ *are full rank matrices satisfying* $\mathrm{span}(\mathsf{A}) \subseteq \mathrm{span}(\mathsf{B})$,
2. $\mathsf{C} \in \mathbb{F}_{\mathsf{q}}^{L_{\mathsf{C}} \times L_{\mathsf{B}}}$ *are full rank matrices (including empty matrices), where* $0 \leq L_{\mathsf{C}} < L_{\mathsf{B}}$,
3. $\mathcal{S} \subseteq [1 : L_{\mathsf{B}}]$ *are sets of size* $|\mathcal{S}| = L_{\mathsf{B}} - L_{\mathsf{C}}$ *such that* $\mathrm{rank}\left(\begin{bmatrix} \mathsf{C} \\ \mathsf{I}(\mathcal{S}) \end{bmatrix}\right) = L_{\mathsf{B}}$,
4. $\mathcal{T} \subseteq \mathcal{K}$ *are sets of size* $|\mathcal{T}| = L_{\mathsf{B}} - L_{\mathsf{C}}$ *such that* $\mathrm{rank}\left(\begin{bmatrix} \mathsf{B}(\mathcal{S}) \\ \mathsf{I}(\mathcal{K} \setminus \mathcal{T}) \end{bmatrix}\right) = K$.

Example: Noisy Additive Channel

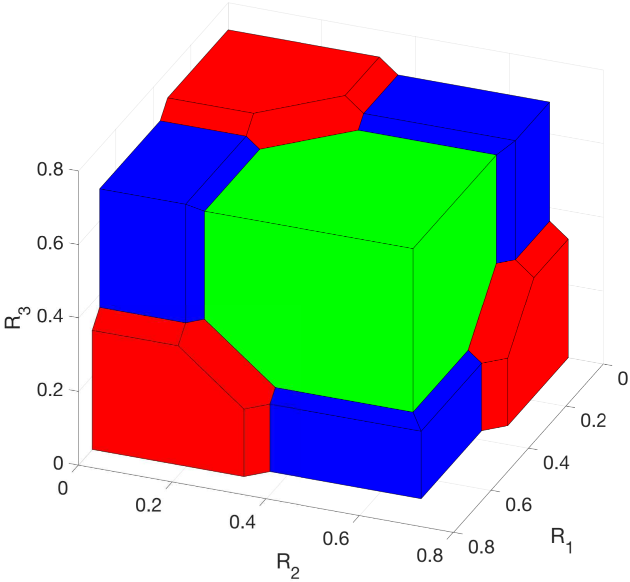


- $\mathcal{X}_k = \{0, 1\}$, $\mathcal{Y} = \{0, 1, 2, 3\}$
- Y is the sum of X_1, X_2, X_3 passed through quaternary symmetric channel
- Fix $p(x_k) \sim \mathrm{Bern}(1/2)$, $U_k = X_k$
- Crossover probability $p = 0.1$

General A-Computation Example

- Compute A $= [1,\ 1,\ 1]$
  - Rank 1: B $=$ A
  - Rank 2: B $= \begin{bmatrix} 1, 1, 0 \\ 0, 0, 1 \end{bmatrix}$, B $= \begin{bmatrix} 1, 0, 1 \\ 0, 1, 0 \end{bmatrix}$, B $= \begin{bmatrix} 0, 1, 1 \\ 1, 0, 0 \end{bmatrix}$,
  - Rank 3: B $= \mathbf{I}$

General A-Computation Example



Example: Gaussian Channel

- Consider a $K = 3$ user Gaussian MAC with channel gain

$$\mathbf{H} = \begin{bmatrix} 1 & 1.5 & 0.75 \\ 0.75 & 1 & 1.5 \\ 1.5 & 0.75 & 1 \end{bmatrix},$$

- $P = 2$, and $\mathsf{A} = [1,\ 1,\ 1]$
- Compare with sequential decoding points $\mathsf{B} = [1,\ 1,\ 1]$ and

$$\mathsf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

Example: Gaussian Channel