Compute-and-Forward:
An Explicit Link between Finite Field and Gaussian Interference Networks

Bobak Nazer
Boston University

Netcod 2013
June 8, 2013
Multi-User Wireless Networks

- Must cope with interference, fading, and noise.
Multi-User Wireless Networks

- Must cope with interference, fading, and noise.
- Receivers observe noisy linear combinations of transmitted signals:

\[ y = \sum_{\ell} h_{\ell} x_{\ell} + z \]
Multi-User Wireless Networks

- Must cope with interference, fading, and noise.
- Receivers observe noisy linear combinations of transmitted signals:
  \[ y = \sum_{\ell} h_{\ell} x_{\ell} + z \]
- How should we deal with interference?
Possible Coding Strategies:

- Avoid interference / orthogonalize.
Possible Coding Strategies:

- Avoid interference / orthogonalize.
Possible Coding Strategies:

- Avoid interference / orthogonalize.
- Treat interference as noise.
Multi-User Wireless Networks

Possible Coding Strategies:

- Avoid interference / orthogonalize.
- Treat interference as noise.
- Decode interfering codewords.
Possible Coding Strategies:

- Avoid interference / orthogonalize.
- Treat interference as noise.
- Decode interfering codewords.
- Work with the analog channel output (or a quantized version).
**Multi-User Wireless Networks**

\[ y = q_1 w_1 \oplus q_2 w_2 \]

**Possible Coding Strategies:**
- Avoid interference / orthogonalize.
- Treat interference as noise.
- Decode interfering codewords.
- Work with the analog channel output (or a quantized version).
- Decode linear combinations of codewords.

**Conventional Approach:** First, eliminate *interference* and then remove *noise*. 
Multi-User Wireless Networks

Possible Coding Strategies:
- Avoid interference / orthogonalize.
- Treat interference as noise.
- Decode interfering codewords.
- Work with the analog channel output (or a quantized version).
- Decode linear combinations of codewords.

- **Conventional Approach:** First, eliminate interference and then remove noise.

- **This Talk:** First, remove noise and then eliminate interference.
**Goal:** Convert noisy Gaussian networks into noiseless finite field ones.
Goal: Convert noisy Gaussian networks into noiseless finite field ones.
Compute-and-Forward

**Goal:** Convert noisy Gaussian networks into noiseless finite field ones.
**Goal:** Convert noisy Gaussian networks into noiseless finite field ones.
**Compute-and-Forward**

**Goal:** Convert noisy Gaussian networks into noiseless finite field ones.
**Compute-and-Forward**

**Goal:** Convert noisy Gaussian networks into noiseless finite field ones.
**Compute-and-Forward**

**Goal:** Convert noisy Gaussian networks into noiseless finite field ones.

- Which linear combinations can be sent over a given channel?
**Goal:** Convert noisy Gaussian networks into noiseless finite field ones.

- Which linear combinations can be sent over a given channel?
- Where can this help us?
Deterministic Model from Avestimehr-Diggavi-Tse ’11:
Top-Down vs. Bottom-Up

Deterministic Model from Avestimehr-Diggavi-Tse ’11:

\[ \hat{u} = q_1 w_1 \oplus q_2 w_2 \]

Compute-and-Forward from Nazer-Gastpar ’11:
Top-Down vs. Bottom-Up

Deterministic Model from Avestimehr-Diggavi-Tse ’11:
• Top-down approach.

Compute-and-Forward from Nazer-Gastpar ’11:
• Bottom-up approach.
Top-Down vs. Bottom-Up

**Deterministic Model** from Avestimehr-Diggavi-Tse ’11:
- Top-down approach.
- Proofs guided by finite-field models but no explicit mapping.

**Compute-and-Forward** from Nazer-Gastpar ’11:
- Bottom-up approach.
- Explicit mapping from Gaussian to finite-field models.
**Top-Down vs. Bottom-Up**

**Deterministic Model** from *Avestimehr-Diggavi-Tse ’11:*
- Top-down approach.
- Proofs guided by finite-field models but no explicit mapping.
- Clear interpretation of achievable rate region as a constant gap from cut-set upper bounds for many interesting topologies.

**Compute-and-Forward** from *Nazer-Gastpar ’11:*
- Bottom-up approach.
- Explicit mapping from Gaussian to finite-field models.
- More challenging to connect achievable rate region to upper bounds.
**Deterministic Model** from Avestimehr-Diggavi-Tse ’11:
- Top-down approach.
- Proofs guided by finite-field models but no explicit mapping.
- Clear interpretation of achievable rate region as a constant gap from cut-set upper bounds for many interesting topologies.
- More challenging to capture physical-layer phenomena such as MIMO and interference alignment.

**Compute-and-Forward** from Nazer-Gastpar ’11:
- Bottom-up approach.
- Explicit mapping from Gaussian to finite-field models.
- More challenging to connect achievable rate region to upper bounds.
- Clear interpretation of physical-layer phenomena such as MIMO and interference alignment.
Road Map

• Compute-and-Forward:
Road Map

• Compute-and-Forward:
  • Achievability results for Gaussian networks.
  • Proof ideas.
Road Map

- Compute-and-Forward:
  - Achievability results for Gaussian networks.
  - Proof ideas.

- Applications to communication across single-hop Gaussian networks.
Road Map

- **Compute-and-Forward:**
  - Achievability results for Gaussian networks.
  - Proof ideas.

- Applications to communication across single-hop Gaussian networks.
Road Map

- Compute-and-Forward:
  - Achievability results for Gaussian networks.
  - Proof ideas.

- Applications to communication across single-hop Gaussian networks.

MIMO Channels

Multiple-Access Channels

1

2

\cdots

K
Compute-and-Forward:
- Achievability results for Gaussian networks.
- Proof ideas.

Applications to communication across single-hop Gaussian networks.
Compute-and-Forward: Problem Statement

- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $x_\ell, y \in \mathbb{R}^n$.
- Power constraint, $\|x_\ell\|^2 \leq np$.
- Gaussian noise, $z \sim \mathcal{N}(0, I)$.
- Equal rates: $R = \frac{k}{n} \log_2 p$.
Compute-and-Forward: Problem Statement

- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $\mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n$.
- Power constraint, $\|\mathbf{x}_\ell\|^2 \leq nP$.
- Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Equal rates: $R = \frac{k}{n} \log_2 p$

- Decoder wants $M$ linear combinations of the messages with vanishing probability of error 
  $$\lim_{n \to \infty} \mathbb{P}\left(\left\{\hat{\mathbf{u}}_1 \neq \mathbf{u}_1\right\} \cup \cdots \cup \left\{\hat{\mathbf{u}}_M \neq \mathbf{u}_M\right\}\right) = 0.$$
Compute-and-Forward: Problem Statement

- Messages are finite field vectors, \( w_\ell \in \mathbb{F}_p^k \).
- Real-valued inputs and outputs, \( x_\ell, y \in \mathbb{R}^n \).
- Power constraint, \( \|x_\ell\|^2 \leq nP \).
- Gaussian noise, \( z \sim \mathcal{N}(0, I) \).
- Equal rates: \( R = \frac{k}{n} \log_2 p \)

- Decoder wants \( M \) linear combinations of the messages with vanishing probability of error
  \( \lim_{n \to \infty} \mathbb{P}
  \left(\{\hat{u}_1 \neq u_1\} \cup \cdots \cup \{\hat{u}_M \neq \hat{u}_M\}\right) = 0 \).

- Receiver can use its channel state information (CSI) to match the linear combination coefficients \( q_{m\ell} \in \mathbb{F}_p \) to the channel coefficients \( h_\ell \in \mathbb{R} \). Transmitters do not require CSI.
Compute-and-Forward: Problem Statement

- Messages are finite field vectors, \( \mathbf{w}_\ell \in \mathbb{F}_p^k \).
- Real-valued inputs and outputs, \( \mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n \).
- Power constraint, \( \|\mathbf{x}_\ell\|^2 \leq nP \).
- Gaussian noise, \( \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \).
- Equal rates: \( R = \frac{k}{n} \log_2 p \)

- Decoder wants \( M \) linear combinations of the messages with vanishing probability of error \( \lim_{n \to \infty} P \left( \{\hat{\mathbf{u}}_1 \neq \mathbf{u}_1\} \cup \cdots \cup \{\hat{\mathbf{u}}_M \neq \hat{\mathbf{u}}_M\} \right) = 0 \).

- Receiver can use its channel state information (CSI) to match the linear combination coefficients \( q_{m\ell} \in \mathbb{F}_p \) to the channel coefficients \( h_\ell \in \mathbb{R} \). Transmitters do not require CSI.

- What rates are achievable as a function of \( h_\ell \) and \( q_{m\ell} \)?
Computation Rate

- Want to characterize achievable rates as a function of $h_\ell$ and $q_m \ell$. 
Computation Rate

- Want to characterize achievable rates as a function of $h_\ell$ and $q_{m\ell}$.
- Easier to think about integer rather than finite field coefficients.
**Computation Rate**

- Want to characterize achievable rates as a function of $h_\ell$ and $q_{m\ell}$.
- Easier to think about integer rather than finite field coefficients.
- The **linear combination** with integer coefficient vector $a_m = [a_{m1} \ a_{m2} \ \cdots \ a_{mL}]^T \in \mathbb{Z}^L$ corresponds to

$$u_m = \bigoplus_{\ell=1}^{L} q_{m\ell} w_\ell \quad \text{where} \quad q_{m\ell} = g^{-1}\left(\left[a_{m\ell}\right] \mod p\right)$$

where $g(\cdot)$ is the natural mapping between $\mathbb{F}_p$ and $\mathbb{Z}$. 
Want to characterize achievable rates as a function of $h_\ell$ and $q_{m\ell}$.

Easier to think about integer rather than finite field coefficients.

The linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1} \ a_{m2} \ \cdots \ a_{mL}]^T \in \mathbb{Z}^L$ corresponds to

$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell} \mathbf{w}_\ell$$

where $q_{m\ell} = g^{-1}\left(\left[a_{m\ell}\right] \mod p\right)$

where $g(\cdot)$ is the natural mapping between $\mathbb{F}_p$ and $\mathbb{Z}$.

Key Definition: The computation rate region described by $R_{\text{comp}}(h, \mathbf{a})$ is achievable if, for any $\epsilon > 0$ and $n, p$ large enough, a receiver can decode any linear combinations with integer coefficient vectors $\mathbf{a}_1, \ldots, \mathbf{a}_M \in \mathbb{Z}^L$ for which the message rate $R$ satisfies

$$R < \min_{m} R_{\text{comp}}(h, \mathbf{a}_m)$$
The computation rate region described by

\[ R_{\text{comp}}(h, a) = \max_{\alpha \in \mathbb{R}} \frac{1}{2} \log^+ \left( \frac{P}{\alpha^2 + P\|\alpha h - a\|^2} \right) \]

is achievable.
Theorem (Nazer-Gastpar ’11)

The computation rate region described by

\[ R_{\text{comp}}(h, a) = \frac{1}{2} \log^+ \left( \frac{P}{a^\top (P^{-1}I + hh^\top)^{-1}a} \right) \]

is achievable.
The computation rate region described by

$$R_{comp}(h, a) = \frac{1}{2} \log^+ \left( \frac{P}{a^\top (P^{-1}I + hh^\top)^{-1}a} \right)$$

is achievable.

if \( R < \min_m R_{comp}(h, a_m) \) for some \( a_m \in \mathbb{Z}^L \) satisfying \( g^{-1}([a_m] \mod p) = q_m \).
Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

\[ R_{\text{comp}}(h, a) = \frac{1}{2} \log^+ \left( \frac{P}{a^T (P^{-1}I + hh^T)^{-1} a} \right) \]

is achievable.

Special Cases:

- Perfect Match: \( R_{\text{comp}}(a, a) = \frac{1}{2} \log^+ \left( \frac{1}{\|a\|^2} + P \right) \)
The computation rate region described by

\[ R_{\text{comp}}(h, a) = \frac{1}{2} \log^+ \left( \frac{P}{a^T (P^{-1} I + hh^T)^{-1} a} \right) \]

is achievable.

**Special Cases:**

- **Perfect Match:** \( R_{\text{comp}}(a, a) = \frac{1}{2} \log^+ \left( \frac{1}{\|a\|^2} + P \right) \)

- **Decode a Message:**

\[ R_{\text{comp}}\left( h, [\underbrace{0 \cdots 0}_{m-1 \text{ zeros}} \ 1 \ 0 \cdots 0]^T \right) = \frac{1}{2} \log \left( 1 + \frac{h_m^2 P}{1 + P \sum_{\ell \neq m} h_\ell^2} \right) \]
\[ y = \sum_{\ell=1}^{L} h_{\ell} x_{\ell} + z \]

\[ = \sum_{\ell=1}^{L} a_{\ell} x_{\ell} + \sum_{\ell=1}^{L} (h_{\ell} - a_{\ell}) x_{\ell} + z \]

**Desired Codebook:**

- Closed under integer linear combinations \(\Rightarrow\) lattice codebook.
Compute-and-Forward: Effective Noise

\[ y = \sum_{\ell=1}^{L} h_{\ell} x_{\ell} + z \]

\[ = \sum_{\ell=1}^{L} a_{\ell} x_{\ell} + \sum_{\ell=1}^{L} (h_{\ell} - a_{\ell}) x_{\ell} + z \]

Effective Noise

**Desired Codebook:**

- Closed under integer linear combinations \( \implies \) lattice codebook.
- Independent effective noise \( \implies \) dithering.
**Compute-and-Forward: Effective Noise**

\[ y = \sum_{\ell=1}^{L} h_\ell x_\ell + z \]

\[ = \sum_{\ell=1}^{L} a_\ell x_\ell + \sum_{\ell=1}^{L} (h_\ell - a_\ell) x_\ell + z \]

Effective Noise

Desired Codebook:
- Closed under integer linear combinations \( \Rightarrow \) lattice codebook.
- Independent effective noise \( \Rightarrow \) dithering.
- Isomorphic to \( \mathbb{F}_p^k \) \( \Rightarrow \) nested lattice codebook.
Nested Lattices

- A lattice is a discrete subgroup of $\mathbb{R}^n$. 
Nested Lattices

- A lattice is a discrete subgroup of $\mathbb{R}^n$.

- Nearest neighbor quantizer:

\[
Q_{\Lambda}(x) = \arg \min_{\lambda \in \Lambda} ||x - \lambda||_2
\]
Nested Lattices

- A lattice is a discrete subgroup of $\mathbb{R}^n$.

- Nearest neighbor quantizer:

  $$Q_\Lambda(x) = \arg\min_{\lambda \in \Lambda} ||x - \lambda||_2$$

- Two lattices $\Lambda$ and $\Lambda_{\text{FINE}}$ are nested if $\Lambda \subset \Lambda_{\text{FINE}}$
Nested Lattices

- A lattice is a discrete subgroup of $\mathbb{R}^n$.

- Nearest neighbor quantizer:
  \[ Q_{\Lambda}(x) = \arg \min_{\lambda \in \Lambda} \| x - \lambda \|_2 \]

- Two lattices $\Lambda$ and $\Lambda_{\text{FINE}}$ are nested if $\Lambda \subset \Lambda_{\text{FINE}}$
**Nested Lattices**

- A lattice is a discrete subgroup of $\mathbb{R}^n$.

- Nearest neighbor quantizer:
  
  $$Q_\Lambda(x) = \arg\min_{\lambda \in \Lambda} \|x - \lambda\|_2$$

- Two lattices $\Lambda$ and $\Lambda_{\text{FINE}}$ are nested if $\Lambda \subset \Lambda_{\text{FINE}}$

- Quantization error serves as modulo operation:
  
  $$[x] \mod \Lambda = x - Q_\Lambda(x).$$

**Distributive Law:**

$$[x_1 + a[x_2] \mod \Lambda] \mod \Lambda = [x_1 + ax_2] \mod \Lambda$$ for all $a \in \mathbb{Z}$. 

Nested Lattice Codes

- **Nested Lattice Code:** Formed by taking all elements of $\Lambda_{\text{FINE}}$ that lie in the fundamental Voronoi region of $\Lambda$. 
**Nested Lattice Codes**

- **Nested Lattice Code:** Formed by taking all elements of $\Lambda_{\text{FINE}}$ that lie in the fundamental Voronoi region of $\Lambda$.

- Fine lattice $\Lambda_{\text{FINE}}$ protects against noise.
Nested Lattice Codes

- **Nested Lattice Code**: Formed by taking all elements of $\Lambda_{\text{FINE}}$ that lie in the fundamental Voronoi region of $\Lambda$.

- Fine lattice $\Lambda_{\text{FINE}}$ protects against noise.

- Coarse lattice $\Lambda$ enforces the power constraint.

$B(0, \sqrt{nP})$
Nested Lattice Codes

- **Nested Lattice Code**: Formed by taking all elements of $\Lambda_{\text{FINE}}$ that lie in the fundamental Voronoi region of $\Lambda$.

- Fine lattice $\Lambda_{\text{FINE}}$ protects against noise.

- Coarse lattice $\Lambda$ enforces the power constraint.

- Existence of good nested lattice codes: Loeliger '97, Forney-Trott-Chung '00, Erez-Litsyn-Zamir '05, Ordentlich-Erez '13.

- **Erez-Zamir '04**: Nested lattice codes can achieve the point-to-point Gaussian capacity.
Construction A: Lattice Codes from Linear Codes

- Map elements \( \{0, 1, 2, \ldots, p - 1\} \) to equally spaced points on \([0, 1)\).

- Choose generator matrix \( G \in \mathbb{F}_p^{n \times k} \) and place its codewords into the unit cube \([0, 1)^n\).
Construction A: Lattice Codes from Linear Codes

- Map elements \( \{0, 1, 2, \ldots, p - 1\} \) to equally spaced points on \([0, 1)\).

- Choose generator matrix \( \mathbf{G} \in \mathbb{F}_p^{n \times k} \) and place its codewords into the unit cube \([0, 1)^n\).

- Enables us to map between nested lattice codewords \( t_\ell \) and finite field messages \( w_\ell \):

Encoding: \( t_\ell = \phi(w_\ell) \)

Decoding: \( \phi^{-1}\left(\sum_{\ell=1}^{L} a_\ell t_\ell \mod \Lambda\right) = \bigoplus_{\ell=1}^{L} q_\ell w_\ell \)

where \( q_\ell = g^{-1}([a_\ell] \mod p) \).
Dithering

- The **effective noise** will be a mixture of the channel inputs $x_\ell$ and the noise $z$.

- **Dithering** can make the effective noise look independent from the desired lattice codeword.
Dithering

- The **effective noise** will be a mixture of the channel inputs $x_\ell$ and the noise $z$.

- **Dithering** can make the effective noise look independent from the desired lattice codeword.

- Map message $w_\ell$ to a **lattice codeword** $t_\ell$. 

![Diagram of lattice codeword placement](image-url)
Dithering

- The **effective noise** will be a mixture of the channel inputs $x_\ell$ and the noise $z$.

- Dithering can make the **effective noise** look independent from the desired lattice codeword.

- Map message $w_\ell$ to a lattice codeword $t_\ell$.

- Generate a **random dither vector** $d_\ell$ uniformly over Voronoi region of $\Lambda$. 
• The effective noise will be a mixture of the channel inputs $x_\ell$ and the noise $z$.

• Dithering can make the effective noise look independent from the desired lattice codeword.

• Map message $w_\ell$ to a lattice codeword $t_\ell$.

• Generate a random dither vector $d_\ell$ uniformly over Voronoi region of $\Lambda$.

• Transmitter sends a dithered codeword:

$$x_\ell = [t_\ell + d_\ell] \mod \Lambda$$
• The **effective noise** will be a mixture of the channel inputs $x_\ell$ and the noise $z$.

• **Dithering** can make the effective noise look independent from the desired lattice codeword.

• Map message $w_\ell$ to a lattice codeword $t_\ell$.

• Generate a random dither vector $d_\ell$ uniformly over Voronoi region of $\Lambda$.

• Transmitter sends a dithered codeword:

$$x_\ell = [t_\ell + d_\ell] \mod \Lambda$$
Dithering

- The **effective noise** will be a mixture of the channel inputs $x_\ell$ and the noise $z$.

- **Dithering** can make the effective noise look independent from the desired lattice codeword.

- Map message $w_\ell$ to a *lattice codeword* $t_\ell$.

- Generate a *random dither vector* $d_\ell$ uniformly over Voronoi region of $\Lambda$.

- Transmitter sends a *dithered* codeword:

  $$ x_\ell = [t_\ell + d_\ell] \mod \Lambda $$

- $x_\ell$ is now independent of the codeword $t_\ell$. 
Compute-and-Forward: Encoding and Decoding

- Map messages to lattice points, $t_\ell = \phi(w_\ell)$. 
Compute-and-Forward: Encoding and Decoding

- Map messages to lattice points, $t_\ell = \phi(w_\ell)$.
- Transmit dithered codewords, $x_\ell = [t_\ell + d_\ell] \mod \Lambda$
Compute-and-Forward: Encoding and Decoding

- Map messages to lattice points, \( t_\ell = \phi(w_\ell) \).
- Transmit dithered codewords, \( x_\ell = [t_\ell + d_\ell] \mod \Lambda \).
- Receiver scales its observation by \( \alpha \), removes dithers, and decodes an integer-linear combination \( \left[ \sum a_\ell t_\ell \right] \mod \Lambda \).

\[
\tilde{y} = \left[ \alpha \left( \sum_{\ell=1}^{L} h_\ell x_\ell + z \right) - \sum_{\ell=1}^{L} a_\ell d_\ell \right] \mod \Lambda \\
= \left[ \sum_{\ell=1}^{L} a_\ell (x_\ell - d_\ell) + \sum_{\ell=1}^{L} (\alpha h_\ell - a_\ell) x_\ell + \alpha z \right] \mod \Lambda \\
= \left[ \sum_{\ell=1}^{L} a_\ell t_\ell \right] \mod \Lambda + z_{\text{effec}}(h, a) \mod \Lambda
\]
Compute-and-Forward: Encoding and Decoding

- Map messages to lattice points, \( t_\ell = \phi(w_\ell) \).
- Transmit dithered codewords, \( x_\ell = [t_\ell + d_\ell] \mod \Lambda \).
- Receiver scales its observation by \( \alpha \), removes dithers, and decodes an integer-linear combination \( \sum a_\ell t_\ell \mod \Lambda \).

\[
\tilde{y} = \left[ \alpha \left( \sum_{\ell=1}^{L} h_\ell x_\ell + z \right) - \sum_{\ell=1}^{L} a_\ell d_\ell \right] \mod \Lambda \\
= \left[ \sum_{\ell=1}^{L} a_\ell (x_\ell - d_\ell) + \sum_{\ell=1}^{L} (\alpha h_\ell - a_\ell) x_\ell + \alpha z \right] \mod \Lambda \\
= \left[ \sum_{\ell=1}^{L} a_\ell t_\ell \right] \mod \Lambda + z_{\text{effec}}(h, a) \mod \Lambda
\]
Compute-and-Forward: Encoding and Decoding

• Map messages to lattice points, $t_\ell = \phi(w_\ell)$.

• Transmit dithered codewords, $x_\ell = [t_\ell + d_\ell] \mod \Lambda$.

• Receiver scales its observation by $\alpha$, removes dithers, and decodes an integer-linear combination $[\sum a_\ell t_\ell] \mod \Lambda$.

$$
\tilde{y} = \left[ \alpha \left( \sum_{\ell=1}^{L} h_\ell x_\ell + z \right) - \sum_{\ell=1}^{L} a_\ell d_\ell \right] \mod \Lambda
$$

$$
= \left[ \sum_{\ell=1}^{L} a_\ell (x_\ell - d_\ell) + \sum_{\ell=1}^{L} (\alpha h_\ell - a_\ell) x_\ell + \alpha z \right] \mod \Lambda
$$

$$
= \left[ \sum_{\ell=1}^{L} a_\ell t_\ell \right] \mod \Lambda + z_{\text{effec}}(h, a) \mod \Lambda
$$

• Apply inverse map $\phi^{-1}$ to integer-linear combination to obtain linear combination $\bigoplus_\ell q_\ell w_\ell$ with coefficients $q_\ell = g^{-1}([a_\ell] \mod p)$. 
All users pick the same nested lattice code:
Choose message vectors over finite field $w_\ell \in \mathbb{F}_p^k$: 

\begin{align*}
\textbf{w}_1 & \rightarrow \begin{array}{c}
\text{Hexagonal grid}
\end{array} \\
\textbf{w}_2 & \rightarrow \begin{array}{c}
\text{Hexagonal grid}
\end{array}
\end{align*}
Compute-and-Forward: Illustration

Map $w_\ell$ to lattice point $t_\ell = \phi(w_\ell)$:
Transmit lattice points over the channel:

\[ h = \begin{bmatrix} 1.4 & 2.1 \end{bmatrix} \]

\[ a = \begin{bmatrix} 2 & 3 \end{bmatrix} \]
Transmit lattice points over the channel:

\[ w_1 \]

\[ w_2 \]

\[ h = \begin{bmatrix} 1.4 & 2.1 \end{bmatrix} \]

\[ a = \begin{bmatrix} 2 & 3 \end{bmatrix} \]
Lattice codewords are scaled by channel coefficients:

\[ h = \begin{bmatrix} 1.4 & 2.1 \end{bmatrix} \]

\[ a = \begin{bmatrix} 2 & 3 \end{bmatrix} \]
Scaled codewords added together plus noise:

\[ w_2 \rightarrow \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array}\]

\[ w_1 \rightarrow \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array}\]

\[ w_2 \rightarrow \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array}\]

\[ \mathbf{h} = \begin{bmatrix} 1.4 & 2.1 \end{bmatrix} \]

\[ \mathbf{a} = \begin{bmatrix} 2 & 3 \end{bmatrix} \]
Scaled codewords added together plus noise:

\[ w_1 \xrightarrow{} w_1 \]
\[ w_2 \xrightarrow{} w_2 \]

\[ x_1 \xrightarrow{} h_1 \]
\[ x_2 \xrightarrow{} h_2 \]

\[ h = [\begin{array}{cc} 1.4 & 2.1 \end{array}] \]
\[ a = [\begin{array}{cc} 2 & 3 \end{array}] \]
Extra noise penalty for non-integer channel coefficients:

\[ h = \begin{bmatrix} 1.4 & 2.1 \end{bmatrix} \]

\[ a = \begin{bmatrix} 2 & 3 \end{bmatrix} \]

Effective noise: \( 1 + P \| h - a \|^2 \)
Compute-and-Forward: Illustration

Scale output by $\alpha$ to reduce non-integer noise penalty:

\[
\alpha \mathbf{h} = \begin{bmatrix} \alpha 1.4 & \alpha 2.1 \end{bmatrix}
\]

\[
\mathbf{a} = \begin{bmatrix} 2 & 3 \end{bmatrix}
\]

Effective noise: $\alpha^2 + P\|\alpha \mathbf{h} - \mathbf{a}\|^2$
Scale output by $\alpha$ to reduce non-integer noise penalty:

\[ \alpha \mathbf{h} = \begin{bmatrix} \alpha 1.4 & \alpha 2.1 \end{bmatrix} \]

\[ \mathbf{a} = \begin{bmatrix} 2 & 3 \end{bmatrix} \]

Effective noise: $\alpha^2 + P \| \alpha \mathbf{h} - \mathbf{a} \|^2$
Decode to the closest lattice point:

\[
\alpha \mathbf{h} = \begin{bmatrix} \alpha 1.4 & \alpha 2.1 \end{bmatrix}
\]

\[
\mathbf{a} = \begin{bmatrix} 2 & 3 \end{bmatrix}
\]

Effective noise: \(\alpha^2 + P ||\alpha \mathbf{h} - \mathbf{a}||^2\)
Recover integer linear combination modulo $\Lambda$:

\[ \alpha \mathbf{h} = \begin{bmatrix} \alpha 1.4 & \alpha 2.1 \end{bmatrix} \]

\[ \mathbf{a} = \begin{bmatrix} 2 & 3 \end{bmatrix} \]

Effective noise: $\alpha^2 + P \|\alpha \mathbf{h} - \mathbf{a}\|^2$
Map back to linear combination of the messages:

\[ \alpha \mathbf{h} = \begin{bmatrix} \alpha_{1.4} & \alpha_{2.1} \end{bmatrix} \]

\[ \mathbf{a} = \begin{bmatrix} 2 & 3 \end{bmatrix} \]

Effective noise: \( \alpha^2 + P \| \alpha \mathbf{h} - \mathbf{a} \|^2 \)
Random i.i.d. codes are not good for computation.

\[ 2^{nR_1} \text{ codewords} \]

\[ 2^{nR_2} \text{ codewords} \]

\[ x_1 \]

\[ x_2 \]

\[ y \]

\[ 2^{n(R_1 + R_2)} \text{ codewords} \]
Random i.i.d. codes are not good for computation.
• Usually fight interference and convert to network of bit pipes.
Physical-Layer Network Coding

- Usually fight interference and convert to network of bit pipes.
• Usually fight interference and convert to network of bit pipes.
• Usually fight interference and convert to network of bit pipes.

• **Physical-layer network coding**: exploiting the wireless medium for network coding. Independently and concurrently proposed by Zhang-Liew-Lam ’06, Popovski-Yomo ’06, Nazer-Gastpar ’06.

• Compute-and-forward is an information-theoretic approach.

• Usually fight interference and convert to network of bit pipes.

• Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by Zhang-Liew-Lam ’06, Popovski-Yomo ’06, Nazer-Gastpar ’06.

• Compute-and-forward is an information-theoretic approach.

• Usually fight interference and convert to network of bit pipes.

• Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by Zhang-Liew-Lam ’06, Popovski-Yomo ’06, Nazer-Gastpar ’06.

• Compute-and-forward is an information-theoretic approach.

Physical-Layer Network Coding

- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by Zhang-Liew-Lam ’06, Popovski-Yomo ’06, Nazer-Gastpar ’06.
- Compute-and-forward is an information-theoretic approach.
• Usually fight interference and convert to network of bit pipes.

• **Physical-layer network coding:** exploiting the wireless medium for network coding. Independently and concurrently proposed by Zhang-Liew-Lam ’06, Popovski-Yomo ’06, Nazer-Gastpar ’06.

• Compute-and-forward is an information-theoretic approach.

• Recent surveys: **Liew-Zhang-Lu ’11, Nazer-Gastpar Proc. IEEE ’11.**
Physical-Layer Network Coding

- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by Zhang-Liew-Lam ’06, Popovski-Yomo ’06, Nazer-Gastpar ’06.
- Compute-and-forward is an information-theoretic approach.
Physical-Layer Network Coding

- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.
- Compute-and-forward is an information-theoretic approach.
Road Map

- Compute-and-Forward:
  - Achievability results for Gaussian networks.
  - Proof ideas.

- Applications to communication across single-hop Gaussian networks.

Joint work with Jiening Zhan, Uri Erez, and Michael Gastpar.
Each antenna encodes an independent data stream of rate $R$ (e.g., V-BLAST setting, cellular uplink).

Channel model: $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}$ where $\mathbf{Z}$ is elementwise i.i.d. $\mathcal{N}(0, 1)$.

Probability of error: $\mathbb{P}(\{\hat{\mathbf{w}}_1 \neq \mathbf{w}_1\} \cup \cdots \cup \{\hat{\mathbf{w}}_M \neq \mathbf{w}_M\}) < \epsilon$

Joint maximum likelihood decoding is optimal but has high implementation complexity.
• Project the received signal, $\tilde{Y} = BY$, to **eliminate interference** between data streams.

• **Zero-Forcing**: Set $B = H^{-1}$ to obtain $\tilde{Y} = X + H^{-1}Z$.

• Significantly reduces complexity at the expense of performance.
Zero-Forcing Linear Receivers

- Project the received signal, $\tilde{Y} = BY$, to eliminate interference between data streams.
- **Zero-Forcing:** Set $B = H^{-1}$ to obtain $\tilde{Y} = X + H^{-1}Z$.
- Significantly reduces complexity at the expense of performance.
- Slight improvement possible by using an MMSE projection instead.
Project the received signal, $\tilde{Y} = BY$, to create an integer-valued effective channel matrix.

**Integer-Forcing:** Set $B = AH^{-1}$ to obtain $\tilde{Y} = AX + AH^{-1}Z$. Solve resulting equations, Zhan-Nazer-Erez-Gastpar ’12.

Optimize over $A \in \mathbb{Z}^{M \times M}$ to minimize effective noise.
• Project the received signal, $\tilde{Y} = B Y$, to create an integer-valued effective channel matrix.

• **Integer-Forcing:** Set $B = A H^{-1}$ to obtain $\tilde{Y} = A X + A H^{-1} Z$. Solve resulting equations, *Zhan-Nazer-Erez-Gastpar '12.*

• Optimize over $A \in \mathbb{Z}^{M \times M}$ to minimize effective noise.

• Includes **zero-forcing** by setting $A = I$. 

\[ u_m = \bigoplus_{\ell} q_{m\ell} w_\ell \]
**Integer-Forcing Linear Receivers**

- Project the received signal, $\tilde{Y} = BY$, to create an integer-valued effective channel matrix.

- **Integer-Forcing:** Set $B = AH^{-1}$ to obtain $\tilde{Y} = AX + AH^{-1}Z$. Solve resulting equations, Zhan-Nazer-Erez-Gastpar ’12.

- Optimize over $A \in \mathbb{Z}^{M \times M}$ to minimize effective noise.

- Includes **zero-forcing** by setting $A = I$.

- Slight improvement possible by using an MMSE projection instead.

\[
\begin{align*}
    w_1 & \xrightarrow{\text{SISO Encoder}} x_1 \\
    w_2 & \xrightarrow{\text{SISO Encoder}} x_2 \\
    \vdots \\
    w_M & \xrightarrow{\text{SISO Encoder}} x_M \\
    y_1 & \xrightarrow{\text{Linear Equalizer}} \tilde{y}_1 \\
    y_2 & \xrightarrow{\text{SISO Decoder}} \hat{u}_1 \\
    \vdots \\
    y_M & \xrightarrow{\text{SISO Decoder}} \hat{u}_M \\
    \hat{w}_1 \\
    \hat{w}_2 \\
    \vdots \\
    \hat{w}_M
\end{align*}
\]
• Receiver has an $M$-dimensional observation of each transmitted symbol:

$$Y = \sum_{m=1}^{M} h_m x_m^T + Z$$

$$= HX + Z$$
MIMO Compute-and-Forward

- Receiver has an $M$-dimensional observation of each transmitted symbol:
  \[ Y = \sum_{m=1}^{M} h_m x_m^T + Z = HX + Z \]

- To recover the linear combination with integer coefficient vector $a \in \mathbb{Z}^L$, the receiver projects its observation:
  \[ b^T Y = a^T X + (b^T H - a^T)X + b^T Z \]
MIMO Compute-and-Forward

- Receiver has an $M$-dimensional observation of each transmitted symbol:
  \[
  Y = \sum_{m=1}^{M} h_m x_m^T + Z = HX + Z
  \]

- To recover the linear combination with integer coefficient vector $a \in \mathbb{Z}^L$, the receiver projects its observation:
  \[
  b^T Y = a^T X + (b^T H - a^T)X + b^T Z
  \]

**Theorem (Zhan-Nazer-Erez-Gastpar ’12)**

The computation rate region described by

\[
R_{\text{comp}}(H, a) = \max_{b \in \mathbb{R}^M} \frac{1}{2} \log^+ \left( \frac{P}{\|b\|^2 + P\|H^T b - a\|^2} \right)
\]

is achievable.
MIMO Compute-and-Forward

- Receiver has an $M$-dimensional observation of each transmitted symbol:
  \[
  Y = \sum_{m=1}^{M} h_m x_m^T + Z = HX + Z
  \]

- To recover the linear combination with integer coefficient vector $a \in \mathbb{Z}^L$, the receiver projects its observation:
  \[
  b^T Y = a^T X + (b^T H - a^T)X + b^T Z
  \]

**Theorem (Zhan-Nazer-Erez-Gastpar '12)**

The computation rate region described by

\[
R_{\text{comp}}(H, a) = \frac{1}{2} \log^+ \left( \frac{P}{a^T(P^{-1}I + HH^T)^{-1}a} \right)
\]

is achievable.
**Comparison**

**Integer-Forcing:**

\[ Y = HX + Z \quad \xrightarrow{\text{Project}} \quad AH^{-1}Y = AX + AH^{-1}Z \quad \xrightarrow{\text{Decode}} \quad AX \]
**Comparison**

**Integer-Forcing:**

\[ Y = HX + Z \quad \xrightarrow{\text{Project}} \quad AH^{-1}Y = AX + AH^{-1}Z \quad \xrightarrow{\text{Decode}} \quad AX \]

- Achievable rate: \( R_{IF}(H) = M \max_{A \in \mathbb{Z}^{M \times M}} \min_{m} R_{\text{comp}}(H, a_m) \)
Comparison

**Integer-Forcing:**

\[
Y = HX + Z \xrightarrow{\text{Project}} AH^{-1}Y = AX + AH^{-1}Z \xrightarrow{\text{Decode}} AX
\]

- Achievable rate: \( R_{IF}(H) = \max_{A \in \mathbb{Z}^{M \times M}} \min_{m} R_{\text{comp}}(H, a_m) \)
  \[\text{rank}(A) = M\]
- Only need to search over vectors satisfying \( \|a_m\|^2 \leq 1 + P \lambda_{\text{max}}^2(H) \).
Comparison

**Integer-Forcing:**

\[ Y = HX + Z \quad \xrightarrow{\text{Project}} \quad AH^{-1}Y = AX + AH^{-1}Z \quad \xrightarrow{\text{Decode}} \quad AX \]

- Achievable rate: \( R_{IF}(H) = M \max_{A \in \mathbb{Z}^{M \times M}} \min_m R_{\text{comp}}(H, a_m) \)
- Only need to search over vectors satisfying \( \|a_m\|^2 \leq 1 + P \lambda_{\text{max}}^2(H) \).
- Faster search: Apply LLL algorithm to \( F = \left( P^{-1}I + HH^T \right)^{-1/2} \).
**Comparison**

**Integer-Forcing:**

\[ Y = HX + Z \quad \xrightarrow{\text{Project}} \quad AH^{-1}Y = AX + AH^{-1}Z \quad \xrightarrow{\text{Decode}} \quad AX \]

- Achievable rate: \( R_{IF}(H) = M \max_{A \in \mathbb{Z}^{M \times M}} \min_{m} R_{\text{comp}}(H, a_m) \) \( m \)
  \( \text{rank}(A) = M \)

- Only need to search over vectors satisfying \( \|a_m\|^2 \leq 1 + P\lambda_{\text{max}}^2(H) \).

- Faster search: Apply LLL algorithm to \( F = (P^{-1}I + HH^T)^{-1/2} \).

**Zero-Forcing:**

\[ Y = HX + Z \quad \xrightarrow{\text{Project}} \quad H^{-1}Y = X + H^{-1}Z \quad \xrightarrow{\text{Decode}} \quad X \]
Comparison

**Integer-Forcing:**

\[ Y = HX + Z \quad \xrightarrow{\text{Project}} \quad AH^{-1}Y = AX + AH^{-1}Z \quad \xrightarrow{\text{Decode}} \quad AX \]

- Achievable rate: \( R_{IF}(H) = M \max_{A \in \mathbb{Z}^{M \times M}} \min_{m} R_{\text{comp}}(H, a_m) \)

- Only need to search over vectors satisfying \( \|a_m\|^2 \leq 1 + P\lambda_{\text{max}}^2(H) \).

- Faster search: Apply LLL algorithm to \( F = (P^{-1}I + HH^T)^{-1/2} \).

**Zero-Forcing:**

\[ Y = HX + Z \quad \xrightarrow{\text{Project}} \quad H^{-1}Y = X + H^{-1}Z \quad \xrightarrow{\text{Decode}} \quad X \]

- Achievable rate: \( R_{ZF}(H) = M \min_{m} R_{\text{comp}}(H, [0 \cdots 0 \ 1 \ 0 \cdots 0]^T) \)

  \( m-1 \) zeros
Channel matrix $\mathbf{H}$ is i.i.d. Rayleigh, only known at the receiver.

For a fixed probability $\rho$, the outage rate is defined to be

$$R_{\text{OUT}}(\rho) = \sup \left\{ R : \Pr (R(\mathbf{H}) < R) \leq \rho \right\}.$$
• Channel matrix $\mathbf{H}$ is i.i.d. Rayleigh, **only known at the receiver**.

• For a fixed probability $\rho$, the **outage rate** is defined to be

$$R_{OUT}(\rho) = \sup \left\{ R : \Pr(R(H) < R) \leq \rho \right\}.$$
- Channel matrix $\mathbf{H}$ is i.i.d. Rayleigh, only known at the receiver.
- For a fixed probability $\rho$, the outage rate is defined to be
  \[ R_{OUT}(\rho) = \sup \left\{ R : \Pr (R(\mathbf{H}) < R) \leq \rho \right\}. \]
- Integer-forcing beats zero-forcing (even if it is augmented with successive interference cancellation (SIC)).
• **Integer-forcing** can adapt to the channel by choosing a basis (of integer vectors) close to the maximum singular vector.
• **Integer-forcing** can adapt to the channel by choosing a basis (of integer vectors) close to the maximum singular vector.

• **Zero-forcing** implicitly decodes using the standard basis.
Zheng-Tse ’03: A family of codes is said to achieve spatial multiplexing gain $r$ and diversity gain $d$ if the total data rate and the average probability of error satisfy

$$\lim_{\text{SNR} \to \infty} \frac{R(\text{SNR})}{\log \text{SNR}} \geq r$$

$$\lim_{\text{SNR} \to \infty} \frac{\log p_{\text{error}}(\text{SNR})}{\log \text{SNR}} \leq -d.$$
Diversity-Multiplexing Tradeoff

- **Zheng-Tse '03:** A family of codes is said to achieve spatial multiplexing gain $r$ and diversity gain $d$ if the total data rate and the average probability of error satisfy

$$\lim_{\text{SNR} \to \infty} \frac{R(\text{SNR})}{\log \text{SNR}} \geq r$$

$$\lim_{\text{SNR} \to \infty} \frac{\log p_{\text{error}}(\text{SNR})}{\log \text{SNR}} \leq -d.$$

- **Zhan-Nazer-Erez-Gastpar '12:** Integer-forcing can attain the optimal DMT while conventional linear receivers cannot.
Diversity-Multiplexing Tradeoff

- **Zheng-Tse ’03**: A family of codes is said to achieve spatial multiplexing gain $r$ and diversity gain $d$ if the total data rate and the average probability of error satisfy

\[
\lim_{\text{SNR} \to \infty} \frac{R(\text{SNR})}{\log \text{SNR}} \geq r
\]

\[
\lim_{\text{SNR} \to \infty} \frac{\log p_{\text{error}}(\text{SNR})}{\log \text{SNR}} \leq -d.
\]

- **Zhan-Nazer-Erez-Gastpar ’12**: Integer-forcing can attain the optimal DMT while conventional linear receivers cannot.

- What about space-time coding at the transmitter?
Diversity-Multiplexing Tradeoff

- **Zheng-Tse ’03**: A family of codes is said to achieve spatial multiplexing gain $r$ and diversity gain $d$ if the total data rate and the average probability of error satisfy

$$\lim_{\text{SNR} \to \infty} \frac{R(\text{SNR})}{\log \text{SNR}} \geq r$$

$$\lim_{\text{SNR} \to \infty} \frac{\log p_{\text{error}}(\text{SNR})}{\log \text{SNR}} \leq -d.$$ 

- **Zhan-Nazer-Erez-Gastpar ’12**: Integer-forcing can attain the optimal DMT while conventional linear receivers cannot.

- **What about space-time coding at the transmitter?**

- **Ordentlich-Erez ’13**: Linear dispersion codes + integer-forcing achieves the MIMO capacity universally to within a constant gap. Includes optimal DMT as a special case.
Road Map

- Compute-and-Forward:
  - Achievability results for Gaussian networks.
  - Proof ideas.

- Applications to communication across single-hop Gaussian networks.

Joint work with Or Ordentlich and Uri Erez.
Gaussian Multiple-Access Channel

\[ \| x_\ell \|^2 \leq nP, \quad z \sim \mathcal{N}(0, I). \]
The capacity region is the set of all rate pairs \((R_1, R_2)\) satisfying:

\[
R_1 < \frac{1}{2} \log(1 + h_1^2 P) \quad R_2 < \frac{1}{2} \log(1 + h_2^2 P)
\]

\[
R_1 + R_2 < \frac{1}{2} \log(1 + \|h\|^2 P)
\]
Gaussian Multiple-Access Channel

The capacity region is the set of all rate pairs \((R_1, R_2)\) satisfying:

\[
R_1 < \frac{1}{2} \log(1 + h_1^2 P) \quad R_2 < \frac{1}{2} \log(1 + h_2^2 P)
\]

\[
R_1 + R_2 < \frac{1}{2} \log(1 + \|h\|^2 P)
\]

Achievable via joint decoding.
\[
\|x_\ell\|^2 \leq nP, \quad z \sim \mathcal{N}(0, I).
\]
Successive Cancellation

\[ \| x_\ell \| ^2 \leq nP, \quad z \sim \mathcal{N}(0, I). \]

- Treat \( x_2 \) as noise and decode \( x_1 \), \( R_1 < \frac{1}{2} \log \left( 1 + \frac{h_1^2 P}{1 + h_2^2 P} \right) \).
Successive Cancellation

\[ \|x_\ell\|^2 \leq nP, \; \; z \sim \mathcal{N}(0, I). \]

- Treat \( x_2 \) as noise and decode \( x_1 \), \( R_1 < \frac{1}{2} \log \left( 1 + \frac{h_1^2 P}{1 + h_2^2 P} \right) \).
- Cancel \( x_1 \) and decode \( x_2 \), \( R_2 < \frac{1}{2} \log (1 + h_2^2 P) \).
\[ \|x_\ell\|^2 \leq nP, \quad z \sim \mathcal{N}(0, I). \]

- Treat \(x_2\) as noise and decode \(x_1\), \( R_1 < \frac{1}{2} \log \left( 1 + \frac{h_1^2P}{1 + h_2^2P} \right) \).
- Cancel \(x_1\) and decode \(x_2\), \( R_2 < \frac{1}{2} \log \left( 1 + h_2^2P \right) \).
- Switch decoding order for the other corner point.
\[
\|x_\ell\|^2 \leq nP, \quad z \sim \mathcal{N}(0, I).
\]

- Treat \(x_2\) as noise and decode \(x_1\), \(R_1 < \frac{1}{2} \log \left(1 + \frac{h_1^2P}{1 + h_2^2P}\right)\).

- Cancel \(x_1\) and decode \(x_2\), \(R_2 < \frac{1}{2} \log (1 + h_2^2P)\).

- Switch decoding order for the other corner point.

- Achieves capacity when combined with time-sharing or rate-splitting (Rimoldi-Urbanke ’96).
Two Linear Combinations

\[ w_1 \rightarrow \mathcal{E}_1 \rightarrow \mathcal{E}_2 \rightarrow w_2 \]

\[ x_1 \rightarrow h_1 \rightarrow z \rightarrow y \rightarrow D \rightarrow \hat{u}_1 \hat{u}_2 \]

- Decode two linearly independent equations.

\[ u_1 = a_{11}w_1 \oplus a_{12}w_2 \]
\[ u_2 = a_{21}w_1 \oplus a_{22}w_2 \]
Two Linear Combinations

\[ w_1 \rightarrow E_1 \xrightarrow{h_1} z \rightarrow y \rightarrow D \xrightarrow{\hat{u}_1} \hat{u}_2 \]
\[ w_2 \rightarrow E_2 \xrightarrow{h_2} \]

- Decode two linearly independent equations.

\[ u_1 = a_{11}w_1 \oplus a_{12}w_2 \]
\[ u_2 = a_{21}w_1 \oplus a_{22}w_2 \]
Two Linear Combinations

\[ w_1 \rightarrow \mathcal{E}_1 \rightarrow x_1 \rightarrow h_1 \rightarrow z \rightarrow y \rightarrow D \rightarrow \hat{u}_1 \hat{u}_2 \]

\[ w_2 \rightarrow \mathcal{E}_2 \rightarrow x_2 \rightarrow h_2 \rightarrow \]

- Decode two linearly independent equations.

\[ u_1 = a_{11} w_1 \oplus a_{12} w_2 \]

\[ u_2 = a_{21} w_1 \oplus a_{22} w_2 \]
Two Linear Combinations

\[ w_1 \rightarrow E_1 \rightarrow x_1, h_1, z \rightarrow y \rightarrow D \rightarrow \hat{u}_1, \hat{u}_2 \]

\[ w_2 \rightarrow E_2 \rightarrow x_2, h_2 \]

- Decode two linearly independent equations.

\[ u_1 = a_{11}w_1 \oplus a_{12}w_2 \]

\[ u_2 = a_{21}w_1 \oplus a_{22}w_2 \]
Two Linear Combinations

\[ w_1 \rightarrow \mathcal{E}_1 \rightarrow x_1 \rightarrow h_1 \rightarrow z \rightarrow y \rightarrow \mathcal{D} \rightarrow \hat{u}_1 \rightarrow \hat{u}_2 \]

\[ w_2 \rightarrow \mathcal{E}_2 \rightarrow x_2 \rightarrow h_2 \rightarrow y \rightarrow \mathcal{D} \]

\[ u_1 = a_{11} w_1 \oplus a_{12} w_2 \]

\[ u_2 = a_{21} w_1 \oplus a_{22} w_2 \]

- Decode two linearly independent equations.

![Graph showing normalized computation rate vs. crossgain g with black, blue, and red lines representing Sum Rate, 1st Equation, and 2nd Equation respectively.](image_url)
Two Linear Combinations

\[ w_1 \xrightarrow{\mathcal{E}_1} x_1 \xrightarrow{h_1} z \]
\[ w_2 \xrightarrow{\mathcal{E}_2} x_2 \xrightarrow{h_2} y \]
\[ u_1 = a_{11}w_1 \oplus a_{12}w_2 \]
\[ u_2 = a_{21}w_1 \oplus a_{22}w_2 \]

- Decode two linearly independent equations.

![Graph showing normalized computation rate against crossgain g](image-url)
Sum of Computation Rates

- Looks as if the sum of computation rates is nearly equal to the MAC sum capacity. Why is this happening?

- Let $F = (P^{-1/2}I + hh^T)^{-1/2}$. Then, each computation rate can be written as
  $$R_{\text{comp}}(h, a_k) = \frac{1}{2} \log^+ \left( \frac{P}{\|F a_k\|^2} \right).$$

- Thus, decoding the best linear combinations is the same as finding the successive minima $\lambda_k(F)$ for the lattice $\Lambda(F) = F\mathbb{Z}^K$:
  $$\lambda_k(F) \triangleq \inf \left\{ r : \dim \left( \text{span} \left( \Lambda(F) \cap B(0, r) \right) \right) \geq k \right\}.$$
Successive Minima
Successive Minima
Successive Minima
Successive Minima
Successive Minima
### Theorem (Minkowski)

Let $\Lambda(\mathbf{F})$ be a lattice spanned by a full-rank $K \times K$ matrix $\mathbf{F}$. Its successive minima $\lambda_k(\mathbf{F})$ satisfy

$$
\prod_{k=1}^{K} \lambda_k^2(\mathbf{F}) \leq K^K |\det(\mathbf{F})|^2 .
$$
Minkowski’s Theorem on Successive Minima

**Theorem (Minkowski)**

Let $\Lambda(F)$ be a lattice spanned by a full-rank $K \times K$ matrix $F$. Its successive minima $\lambda_k(F)$ satisfy

$$\prod_{k=1}^{K} \lambda_k^2(F) \leq K^K |\text{det}(F)|^2.$$ 

**Theorem (Ordentlich-Erez-Nazer ’12)**

The sum of the $K$ best linearly independent computation rates satisfies

$$\sum_{k=1}^{K} R_{\text{comp}}(h, a_k) \geq \frac{1}{2} \log(1 + \|h\|^2 \text{SNR}) - \frac{K}{2} \log K.$$
Operational Interpretation: Multiple-Access

- Associate each computation rate to a message.

\[ u_1 = a_{11}w_1 \oplus a_{12}w_2 \]
\[ u_2 = a_{21}w_1 \oplus a_{22}w_2 \]
Operational Interpretation: Multiple-Access

- Associate each computation rate to a message.

\[ u_1 = a_{11}w_1 \oplus a_{12}w_2 \]
\[ u_2 = a_{21}w_1 \oplus a_{22}w_2 \]

- Decoding the first equation succeeds since

\[ \max(R_1, R_2) < R_{\text{comp}}(h, a_1) \]
Operational Interpretation: Multiple-Access

\[ w_1 \rightarrow E_1 \quad x_1 \quad h_1 \quad z \quad y \quad D \quad \hat{u}_1 \quad \hat{u}_2 \]

\[ w_2 \rightarrow E_2 \quad x_2 \quad h_2 \quad u_1 = a_{11} w_1 \oplus a_{12} w_2 \]
\[ u_2 = a_{21} w_1 \oplus a_{22} w_2 \]

- Associate each computation rate to a message.

- Decoding the first equation succeeds since
  \[ \max(R_1, R_2) < R_{\text{comp}}(h, a_1) . \]

- Decoding the second equation runs into an issue:
  \[ \max(R_1, R_2) > R_{\text{comp}}(h, a_2) . \]
After decoding the first equation, the receiver knows

\[ v_1 = \left[ a_{11} t_1 + a_{12} t_2 \right] \mod \Lambda .\]
• After decoding the first equation, the receiver knows

\[ v_1 = [a_{11}t_1 + a_{12}t_2] \mod \Lambda. \]

• The effective channel for the second equation is

\[ \tilde{y}_2 = [a_{21}t_1 + a_{22}t_2 + z_{\text{eff}}(h, a_2)] \mod \Lambda. \]
(Algebraic) Successive Cancellation

- After decoding the first equation, the receiver knows

\[ v_1 = [a_{11}t_1 + a_{12}t_2] \mod \Lambda. \]

- The effective channel for the second equation is

\[ \tilde{y}_2 = [a_{21}t_1 + a_{22}t_2 + z_{\text{effec}}(h, a_2)] \mod \Lambda. \]

- Using \( v_1 \) we can cancel out \( t_1 \) from \( \tilde{y}_2 \) without changing the effective noise.

\[ \tilde{y}_2^{SI} = [s_2 - b_1v_1] \mod \Lambda \\
= \left[(a_{22} - b_1a_{12})t_2 + z_{\text{effec}}(h, a_2)\right] \mod \Lambda. \]
After decoding the first equation, the receiver knows

\[ v_1 = [a_{11} t_1 + a_{12} t_2] \mod \Lambda. \]

The effective channel for the second equation is

\[ \tilde{y}_2 = [a_{21} t_1 + a_{22} t_2 + z_{\text{effec}}(h, a_2)] \mod \Lambda. \]

Using \( v_1 \) we can cancel out \( t_1 \) from \( \tilde{y}_2 \) without changing the effective noise.

\[
\tilde{y}^{SI}_2 = \left[ s_2 - b_1 v_1 \right] \mod \Lambda \\
= \left[ (a_{22} - b_1 a_{12}) t_2 + z_{\text{effec}}(h, a_2) \right] \mod \Lambda.
\]

Now, the receiver can decode since \( R_2 < R_{\text{comp}}(h, a_2) \).
Multiple-Access via Computation

\[ w_1 \rightarrow \mathcal{E}_1 \rightarrow x_1 \rightarrow h_1 \rightarrow z \rightarrow y \rightarrow D \rightarrow \hat{w}_1 \]

\[ w_2 \rightarrow \mathcal{E}_2 \rightarrow x_2 \rightarrow h_2 \rightarrow y \rightarrow \hat{w}_2 \]
Multiple-Access via Computation

Successive cancellation (without time-sharing or rate-splitting) achieves corner points.
Multiple-Access via Computation

- **Successive cancellation** (without time-sharing or rate-splitting) achieves corner points.
Multiple-Access via Computation

- **Successive cancellation** (without time-sharing or rate-splitting) achieves corner points.
• **Successive cancellation** (without time-sharing or rate-splitting) achieves corner points.

• **Compute-and-forward** achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.
Multiple-Access via Computation

- **Successive cancellation** (without time-sharing or rate-splitting) achieves corner points.

- **Compute-and-forward** achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.
Multiple-Access via Computation

- **Successive cancellation** (without time-sharing or rate-splitting) achieves corner points.

- **Compute-and-forward** achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.

- **Successive compute-and-forward** can attain the exact sum rate. Key Idea: Perform an MMSE projection between each decoding step. 

Nazer ’12, Ordentlich-Erez-Nazer ’13.
• **Successive cancellation** (without time-sharing or rate-splitting) achieves corner points.

• **Compute-and-forward** achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.

• **Successive compute-and-forward** can attain the exact sum rate. Key Idea: Perform an MMSE projection between each decoding step Nazer ’12, Ordentlich-Erez-Nazer ’13.
Road Map

- Decoding linear combinations.
  - Achievability results for Gaussian networks.

- Applications to communication across single-hop Gaussian networks.

Joint work with:
Symmetric case: Or Ordentlich and Uri Erez.
Interference-Free Capacity
Interference-Free Capacity
Time Division
Time Division
Time Division
Time Division

1

2

K

1

2

K
• **Cadambe-Jafar ’08**: Alignment can achieve $K/2$ degrees-of-freedom for the $K$-user interference channel.

• **Birk-Kol ’98**: Alignment for index coding. **Maddah-Ali - Motahari - Khandani ’08**: Alignment for the MIMO X channel. See Jafar ’11 monograph (or recent e-book) for a richer history.
• **Cadambe-Jafar ’08**: Alignment can achieve $K/2$ degrees-of-freedom for the $K$-user interference channel.

• **Birk-Kol ’98**: Alignment for index coding. **Maddah-Ali - Motahari - Khandani ’08**: Alignment for the MIMO X channel. See **Jafar ’11** monograph (or recent e-book) for a richer history.
- **Cadambe-Jafar ’08**: Alignment can achieve $K/2$ degrees-of-freedom for the $K$-user interference channel.

• **Cadambe-Jafar ’08:** Alignment can achieve $K/2$ degrees-of-freedom for the $K$-user interference channel.

• **Birk-Kol ’98:** Alignment for index coding. Maddah-Ali - Motahari - Khandani ’08: Alignment for the MIMO X channel. See Jafar ’11 monograph (or recent e-book) for a richer history.
• **Cadambe-Jafar ’08:** Alignment can achieve $K/2$ degrees-of-freedom for the $K$-user interference channel.

• **Birk-Kol ’98:** Alignment for index coding. Maddah-Ali - Motahari - Khandani ’08: Alignment for the MIMO X channel. See Jafar ’11 monograph (or recent e-book) for a richer history.
• **Cadambe-Jafar ’08**: Alignment can achieve \( K/2 \) degrees-of-freedom for the \( K \)-user interference channel.

• **Birk-Kol ’98**: Alignment for index coding. **Maddah-Ali - Motahari - Khandani ’08**: Alignment for the MIMO X channel. See **Jafar ’11** monograph (or recent e-book) for a richer history.
• **Cadambe-Jafar ’08**: Alignment can achieve $K/2$ degrees-of-freedom for the $K$-user interference channel.

• **Birk-Kol ’98**: Alignment for index coding. Maddah-Ali - Motahari - Khandani ’08: Alignment for the MIMO X channel. See Jafar ’11 monograph (or recent e-book) for a richer history.
• **Cadambe-Jafar ’08**: Alignment can achieve $K/2$ degrees-of-freedom for the $K$-user interference channel.

• **Birk-Kol ’98**: Alignment for index coding. **Maddah-Ali - Motahari - Khandani ’08**: Alignment for the MIMO X channel. See Jafar ’11 monograph (or recent e-book) for a richer history.
What do we need to get half the cake? (so far)

- Very large channel diversity (e.g., time extensions, frequency bands):
What do we need to get half the cake? (so far)

- Very large channel diversity (e.g., time extensions, frequency bands):

- Very high SNR:
What do we need to get half the cake? (so far)

- **Very large channel diversity** (e.g., time extensions, frequency bands):
  - **Cadambe, Jafar ’08**: Asymptotic alignment. Achieves $\frac{K}{2}$ DoF across roughly $2^{K^2}$ channel realizations. **Signal space alignment**.

- **Very high SNR**:
What do we need to get half the cake? (so far)

- **Very large channel diversity** (e.g., time extensions, frequency bands):
  - **Cadambe, Jafar '08**: Asymptotic alignment. Achieves $\frac{K}{2}$ DoF across roughly $2^{K^2}$ channel realizations. *Signal space alignment.*
  - **Nazer, Gastpar, Jafar, Vishwanath ’12**: Ergodic alignment. Achieves $\frac{K}{2} \log(1 + 2\text{SNR})$ across roughly $(K \text{SNR})^{K^2/2}$ channel realizations. *Signal space alignment.*

- **Very high SNR:**
What do we need to get half the cake? (so far)

- **Very large channel diversity** (e.g., time extensions, frequency bands):
  - **Cadambe, Jafar ’08**: Asymptotic alignment. Achieves $\frac{K}{2}$ DoF across roughly $2^{K^2}$ channel realizations. *Signal space alignment.*
  - **Nazer, Gastpar, Jafar, Vishwanath ’12**: Ergodic alignment. Achieves $\frac{K}{2} \log(1 + 2\text{SNR})$ across roughly $(K\text{SNR})^{K^2/2}$ channel realizations. *Signal space alignment.*

- **Very high SNR**:
  - **Motahari, Gharan, Maddah-Ali, Khandani ’09**: Real alignment. Achieves $\frac{K}{2}$ DoF over one channel realization using roughly $2^{K^2}$ codeword layers. *Signal scale alignment.*
Ntranos-Cadambe-Nazer-Caire ’13:

- Proposed a new framework, **integer-forcing interference alignment**, that can simultaneously exploit **signal space** and **signal scale** alignment.
Ntranos-Cadambe-Nazer-Caire ’13:

- Proposed a new framework, **integer-forcing interference alignment**, that can simultaneously exploit **signal space** and **signal scale** alignment.

- Aimed at scenarios with **finite channel diversity** (e.g., a few independent fading realizations) and **finite SNR**.
Ntranos-Cadambe-Nazer-Caire ’13:

- Proposed a new framework, **integer-forcing interference alignment**, that can simultaneously exploit **signal space** and **signal scale** alignment.

- Aimed at scenarios with **finite channel diversity** (e.g., a few independent fading realizations) and **finite SNR**.

- Yields an achievable rate region for any scenario which employs “stream-by-stream” alignment.
Proposed a new framework, integer-forcing interference alignment, that can simultaneously exploit signal space and signal scale alignment.

Aimed at scenarios with finite channel diversity (e.g., a few independent fading realizations) and finite SNR.

Yields an achievable rate region for any scenario which employs “stream-by-stream” alignment.

First, we need to develop a finer understanding of signal scale alignment which so far is only well-understood in the high SNR regime.
Symmetric $K$-User Gaussian Interference Channel

\[ \begin{align*}
E_1 & \xrightarrow{w_1} x_1 & \xrightarrow{H} & \xrightarrow{D_1} \hat{w}_1 \\
E_2 & \xrightarrow{w_2} x_2 & \xrightarrow{H} & \xrightarrow{D_2} \hat{w}_2 \\
\vdots & \quad \vdots & & \quad \vdots \\
E_K & \xrightarrow{w_K} x_K & \xrightarrow{H} & \xrightarrow{D_K} \hat{w}_K
\end{align*} \]

- **Signal space** alignment (e.g., beamforming) is infeasible.
- **Signal scale** alignment attains $K/2$ degrees-of-freedom for almost all channel gains, Motahari et al. ’09, Wu-Shamai-Verdu ’11.
- At finite SNR, the approximate capacity known in some special cases: two-user Etkin-Tse-Wang ’08, many-to-one and one-to-many Bresler-Parekh-Tse ’10, cyclic Zhou-Yu ’10.
- **Signal space** alignment (e.g., beamforming) is infeasible.
- **Signal scale** alignment attains $K/2$ degrees-of-freedom for almost all channel gains, Motahari et al. ’09, Wu-Shamai-Verdu ’11.
- At finite SNR, the approximate capacity known in some special cases: two-user Etkin-Tse-Wang ’08, many-to-one and one-to-many Bresler-Parekh-Tse ’10, cyclic Zhou-Yu ’10.
- We will approximate the sum capacity for the **symmetric** case.
Effective Multiple-Access Channel

- **Lattice codes** can enable **signal scale** alignment.

- Each receiver sees an effective two-user multiple-access channel,
  \[
  y_k = x_k + g \sum_{\ell \neq k} x_\ell + z_k .
  \]
Lattice codes can enable signal scale alignment.

Each receiver sees an effective two-user multiple-access channel,

\[ y_k = x_k + g \sum_{\ell \neq k} x_\ell + z_k. \]

Successive cancellation: Decode and subtract interference \( \sum_{\ell \neq k} x_\ell \) before going after desired message.

Only optimal when the interference is very strong, Sridharan et al. ’08.
Lattice codes can enable signal scale alignment.

Each receiver sees an effective two-user multiple-access channel,

\[ y_k = x_k + g \sum_{\ell \neq k} x_\ell + z_k. \]

Successive cancellation: Decode and subtract interference \( \sum_{\ell \neq k} x_\ell \) before going after desired message.

Only optimal when the interference is very strong, Sridharan et al. ’08.

Unfortunately, direct analysis of joint decoding is hindered by dependencies between codeword pairs. Existing work only applies at very high SNR, Ordentlich-Erez ’13.
Capacity understood in the high SNR regime. *Jafar-Vishwanath ’10.*

\[
\alpha = \frac{\log g^2 \text{SNR}}{\log \text{SNR}} \quad \quad \quad \quad \quad d(\alpha) = \lim_{\text{SNR} \to \infty} \frac{R(\text{SNR})}{\frac{1}{2} \log \text{SNR}}
\]
Alignment via Two Equations

- Each receiver sees an effective two-user multiple-access channel,
  \[ y_k = x_k + g \sum_{\ell \neq k} x_{\ell} + z_k. \]

- **Ordentlich-Erez-Nazer ’12**: Decode two linear combinations:
  \[ a_1 x_k + a_2 \sum_{\ell \neq k} x_{\ell} \quad \quad b_1 x_k + b_2 \sum_{\ell \neq k} x_{\ell}. \]

- Using the fact that the sum of the computation rates is nearly equal to the multiple-access sum capacity, we can approximate the sum capacity of the symmetric $K$-user Gaussian interference channel in all regimes.
Symmetric $K$-User Gaussian Interference Channel

![Graph showing sum rate vs. channel coefficient $g$ at 20 dB]
Symmetric $K$-User Gaussian Interference Channel

35dB

Channel coefficient $g$

Sum rate
Symmetric $K$-User Gaussian Interference Channel

50dB

![Graph showing sum rate vs. channel coefficient $g$. The graph has two curves, one black and one red, indicating different scenarios or parameters. The x-axis represents the channel coefficient $g$, ranging from $10^{-2}$ to $10^2$, and the y-axis represents the sum rate, ranging from 0 to 8.]
Symmetric $K$-User Gaussian Interference Channel

![Graph showing sum rate vs. channel coefficient $g$ with a peak at $65$ dB.]
• Would like to combine signal scale alignment (e.g., lattice codes) with signal space alignment (e.g., beamforming vectors).

• Ntranos-Cadambe-Nazer-Caire ’13: New framework, integer-forcing interference alignment, that can be applied to any scenario with “stream-by-stream” alignment.
Problem Setting:

- Multiple data streams (i.e., codewords) \( s^{[\ell]} \in \mathbb{C}^T \), each assigned to its own beamforming vector \( v^{[\ell]} \in \mathbb{C}^M \).
Problem Setting:

- Multiple data streams (i.e., codewords) $s^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $v^{[\ell]} \in \mathbb{C}^M$.

- Each receiver sees a noisy linear combination of its desired and interfering streams:

\[
Y = \sum_{\ell=1}^{L} H^{[\ell]}_D v^{[\ell]}_D (s^{[\ell]}_D)^T + \sum_{j=1}^{J} \sum_{\ell=1}^{L} H^{[j,\ell]}_I v^{[j,\ell]}_I (s^{[j,\ell]}_I)^T + Z.
\]
Problem Setting:

- Multiple data streams (i.e., codewords) $s^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $v^{[\ell]} \in \mathbb{C}^M$.

- Each receiver sees a noisy linear combination of its desired and interfering streams:

$$Y = \sum_{\ell=1}^{L} H_D^{[\ell]} v_D^{[\ell]} (s_D^{[\ell]})^T + \sum_{j=1}^{J} \sum_{\ell=1}^{L} H_I^{[j,\ell]} v_I^{[j,\ell]} (s_I^{[j,\ell]})^T + Z.$$ 

- **Signal space** alignment occurs if, within each group $j$, all interferers have the same effective channel:

$$H_I^{[j,\ell]} v_I^{[j,\ell]} = H_I^{[j,\tilde{\ell}]} v_I^{[j,\tilde{\ell}]} \quad \forall \tilde{\ell} \neq \ell.$$
Stream-by-Stream Alignment

Problem Setting:

- Multiple data streams (i.e., codewords) $s^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $v^{[\ell]} \in \mathbb{C}^M$.

- Each receiver sees a noisy linear combination of its desired and interfering streams:

$$Y = \sum_{\ell=1}^{L} H_D^{[\ell]} v_D^{[\ell]} (s_D^{[\ell]})^T + \sum_{j=1}^{J} \sum_{\ell=1}^{L} H_I^{[j,\ell]} v_I^{[j,\ell]} (s_I^{[j,\ell]})^T + Z .$$

- Signal space alignment occurs if, within each group $j$, all interferers have the same effective channel:

$$H_I^{[j,\ell]} v_I^{[j,\ell]} = H_I^{[j,\tilde{\ell}]} v_I^{[j,\tilde{\ell}]} \quad \forall \tilde{\ell} \neq \ell .$$

- Different powers and rates allowed across data streams.
Example: Cadambe-Jafar ’08 over 3 Channel Realizations
Example: Cadambe-Jafar ’08 over 3 Channel Realizations
Example: Cadambe-Jafar ’08 over 3 Channel Realizations
Example: Cadambe-Jafar ’08 over 3 Channel Realizations
Example: Cadambe-Jafar ’08 over 3 Channel Realizations
Example: Cadambe-Jafar ’08 over 3 Channel Realizations
Example: Cadambe-Jafar '08 over 3 Channel Realizations
Example: Cadambe-Jafar ’08 over 3 Channel Realizations
Example: Cadambe-Jafar ’08 over 3 Channel Realizations

Total Degrees of Freedom

\[ \text{DoF} = \frac{4 \text{ vectors}}{3 \text{ channel uses}} = \frac{4}{3} \]
Stream-by-Stream Alignment: Receiver Perspective

Received Signal

$H_D^{[1]} v_D^{[1]} \rightarrow s_D^{[1]}$
Stream-by-Stream Alignment: Receiver Perspective

Received Signal

\[ H_{D}^{[1]} v_{D}^{[1]} \]

\[ H_{D}^{[2]} v_{D}^{[2]} \]

\[ s_{D}^{[1]} \]

\[ s_{D}^{[2]} \]
Received Signal

\[ \mathbf{s}_D \]

\[ \mathbf{H}_D^{[1]} \mathbf{v}_D^{[1]} \]

\[ \mathbf{H}_I^{[1,1]} \mathbf{v}_I^{[1,1]} \]

\[ \mathbf{H}_D^{[2]} \mathbf{v}_D^{[2]} \]

\[ \mathbf{s}_D^{[2]} \]
Received Signal

- $s_D^{[1]}$ and $s_I^{[1]}$ for stream 1
- $s_D^{[2]}$ and $s_I^{[2]}$ for stream 2
- $H_D^{[1]}\cdot v_D^{[1]}$ and $H_I^{[1]}\cdot v_I^{[1]}$ for stream 1
- $H_D^{[2]}\cdot v_D^{[2]}$ and $H_I^{[2]}\cdot v_I^{[2]}$ for stream 2
Received Signal

- $s_D^{[1]}$
- $s_D^{[2]}$
- $s_I^{[1,1]}$
- $s_I^{[1,2]}$
Stream-by-Stream Alignment: Receiver Perspective

Received Signal

\[ s_D^{[1]} \rightarrow s_I^{[1]} \rightarrow s_I^{[2]} \rightarrow s_D^{[2]} \]
Zero-Forcing Decoding

How should each receiver decode its desired data streams?

Zero-Forcing Interference Alignment:
- Generate the data streams using i.i.d. random coding.
Zero-Forcing Decoding

How should each receiver decode its desired data streams?

Zero-Forcing Interference Alignment:

- Generate the data streams using i.i.d. random coding.
- First, project the received signal into the nullspace of the interference.
Zero-Forcing Decoding

How should each receiver decoder its desired data streams?

Zero-Forcing Interference Alignment:

- Generate the data streams using i.i.d. random coding.
- First, project the received signal into the nullspace of the interference.
- Then, jointly decode desired data streams.
How should each receiver decoder its desired data streams?

Zero-Forcing Interference Alignment:

- Generate the data streams using i.i.d. random coding.
- First, project the received signal into the nullspace of the interference.
- Then, jointly decode desired data streams.
- Suffices from a degrees-of-freedom perspective.
Joint Decoding (with i.i.d Random Codes)

How should each receiver decoder its desired data streams?

Joint Typicality Decoding:
• Generate the data streams using i.i.d. random coding.
How should each receiver decoder its desired data streams?

Joint Typicality Decoding:

- Generate the data streams using i.i.d. random coding.
- If we attempt to decode the aligned interference, we will end up decoding each interferer separately.
- This significantly reduces the achievable rate per data stream.
How should each receiver decoder its desired data streams?

Joint Typicality Decoding:

- Generate the data streams using i.i.d. random coding.
- If we attempt to decode the aligned interference, we will end up decoding each interferer separately.
- This significantly reduces the achievable rate per data stream.
- Analyzing lattice-coded data streams is beyond the reach of current techniques owing to dependencies.
How should each receiver decoder its desired data streams?

**Integer-Forcing Interference Alignment:**

- Beamforming directions chosen to induce *signal space* alignment.
How should each receiver decoder its desired data streams?

**Integer-Forcing Interference Alignment:**

- Beamforming directions chosen to induce *signal space* alignment.
- Data streams are encoded using *nested lattice codes* according to some power allocation. This induces *signal scale* alignment.
**Integer-Forcing Interference Alignment**

How should each receiver decoder its desired data streams?

**Integer-Forcing Interference Alignment:**

- Beamforming directions chosen to induce *signal space* alignment.
- Data streams are encoded using *nested lattice codes* according to some power allocation. This induces *signal scale* alignment.
- Receiver decodes linear combinations and solves for its desired data streams.
Integer-Forcing Interference Alignment:

 receive decode

\[ s_1^{[1]} + s_1^{[2]} \]

\[ s_2^{[1]} \]

\[ s_2^{[2]} \]

\[ s_D^{[1]} \]

\[ s_D^{[2]} \]

How should each receiver decoder its desired data streams?

Integer-Forcing Interference Alignment:

- Beamforming directions chosen to induce signal space alignment.
- Data streams are encoded using nested lattice codes according to some power allocation. This induces signal scale alignment.
- Receiver decodes linear combinations and solves for its desired data streams.
- Requires extension of compute-and-forward to asymmetric powers.
• 3-user Gaussian interference channel.
• Can code over 3 independent fading realizations from an i.i.d. Rayleigh distribution.

Strategies:
• CJ ’08 Beamforming + Zero-Forcing Decoding.
• CJ ’08 Beamforming + Integer-Forcing Decoding.
Recent coding perspectives on **compute-and-forward**:

- **Feng-Silva-Kschischang ’13**: General algebraic framework in terms of lattice partitions and R-modules.

- **Hern-Narayanan ’11**: Multilevel binary codes.

- **Ordentlich-Erez ’10, Yang et al. ’12**: Binary convolutional codes.

- **Hong and Caire ’11, Ordentlich et al. ’11**: Binary and $p$-ary LDPC codes.

- **Feng-Silva-Kschischang ’11, Belfiore-Ling ’12**: Code design criteria.
Many other scenarios where lattice codes can help:

- **Two-Way Relaying**: Wilson-Narayanan-Pfister-Sprintson ’10, Nam-Chung-Lee ’10

- **Distributed Source Coding**: Krithivasan-Pradhan ’08,’09, Wagner ’11, Tse-Maddah-Ali ’10

- **Decentralized Processing**: Sanderovich-Peleg-Shamai ’11, Nazer-Sanderovich-Gastpar-Shamai ’09, Hong-Caire ’12

- **Distributed Dirty-Paper Coding**: Philosof-Zamir ’09, Philosof-Zamir-Erez-Khisti ’11, Wang ’12

- **Joint Source-Channel Coding**: Kochman-Zamir ’09, Nazer-Gastpar ’07, ’08, Soundararajan-Viswanath ’09

- **Physical-Layer Secrecy**: He-Yener ’09, Agrawal-Vishwanath ’09

- **MIMO Broadcast/Downlink**: Hong-Caire ’12
Concluding Remarks

• Even if you **only want to recover messages**, it can help to first **decode linear combinations**.

• Compute-and-forward creates a direct link between **Gaussian interference networks** and **finite field** ones.

• Enables more efficient encoding/decoding for networks where the capacity is already known.

• Yields new achievable rates for interference channels.