Compute-and-Forward:

An Explicit Link between Finite Field and Gaussian Interference Networks

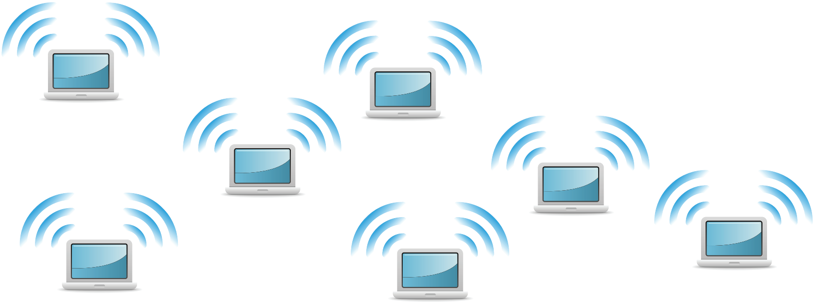
Bobak Nazer
Boston University

Netcod 2013
June 8, 2013

Multi-User Wireless Networks

- Must cope with interference, fading, and noise.

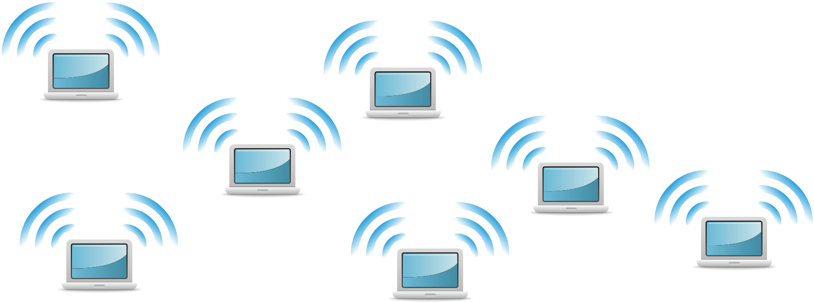
Multi-User Wireless Networks



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

$$\mathbf{y} = \sum_{\ell} h_{\ell} \mathbf{x}_{\ell} + \mathbf{z}$$

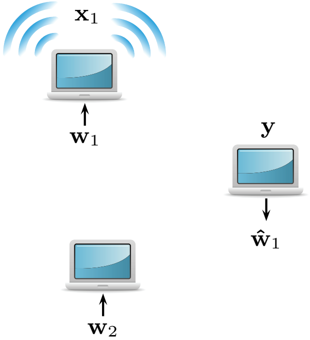
Multi-User Wireless Networks



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

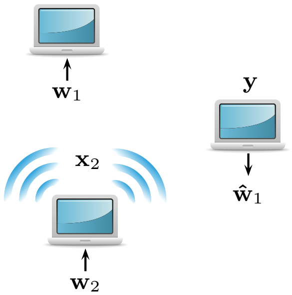
$$\mathbf{y} = \sum_{\ell} h_{\ell} \mathbf{x}_{\ell} + \mathbf{z}$$

- How should we deal with interference?



Possible Coding Strategies:

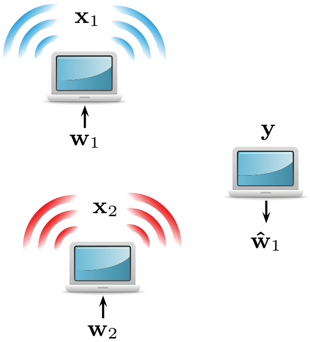
- Avoid interference / orthogonalize.



Possible Coding Strategies:

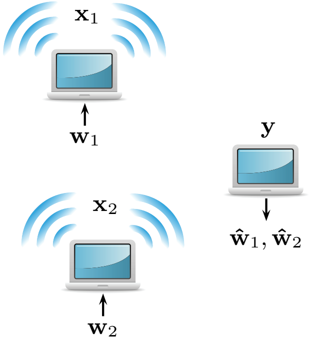
- Avoid interference / orthogonalize.

Multi-User Wireless Networks



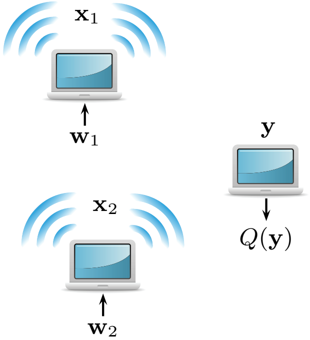
Possible Coding Strategies:

- Avoid interference / orthogonalize.
- Treat interference as noise.



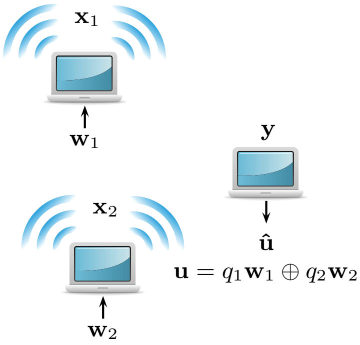
Possible Coding Strategies:

- Avoid interference / orthogonalize.
- Treat interference as noise.
- Decode interfering codewords.



Possible Coding Strategies:

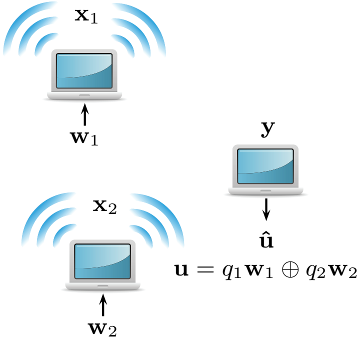
- Avoid interference / orthogonalize.
- Treat interference as noise.
- Decode interfering codewords.
- Work with the analog channel output (or a quantized version).



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.



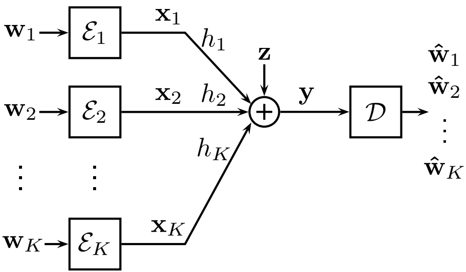
Possible Coding Strategies:

- 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.

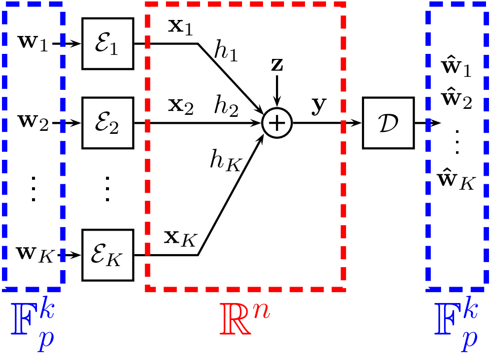
- **Conventional Approach:** First, eliminate interference and then remove noise.
- **This Talk:** First, remove noise and then eliminate interference.

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

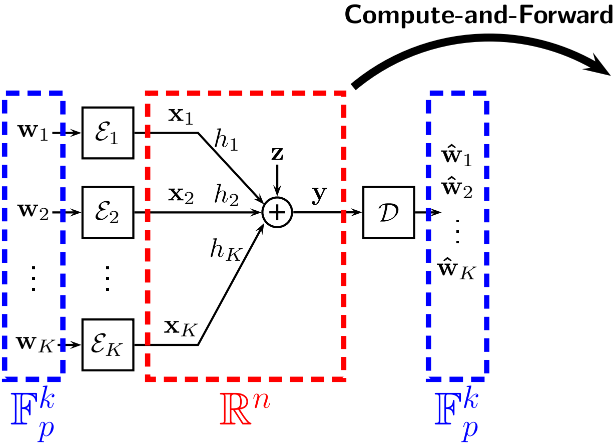
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



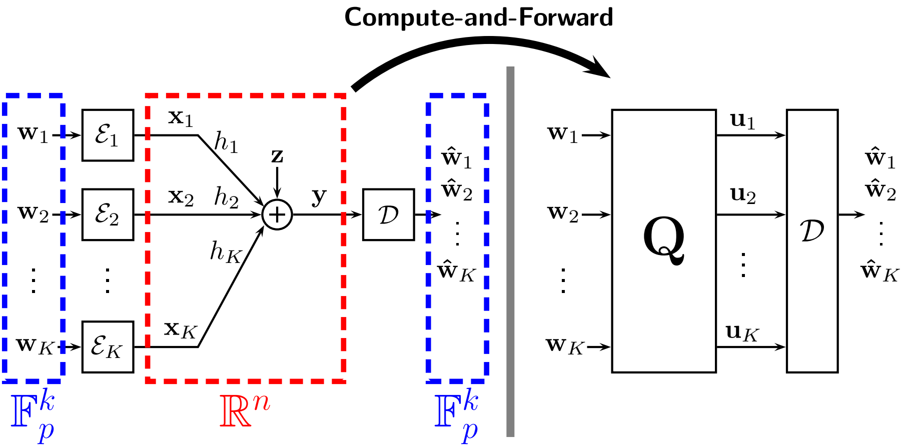
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



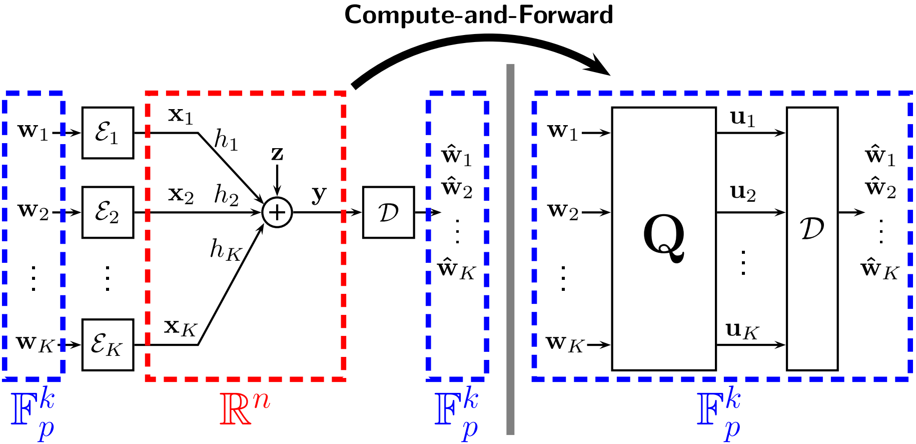
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



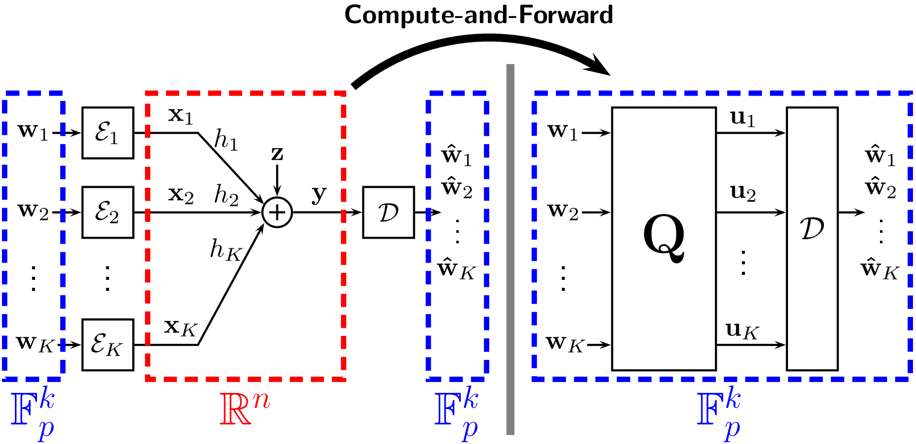
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



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

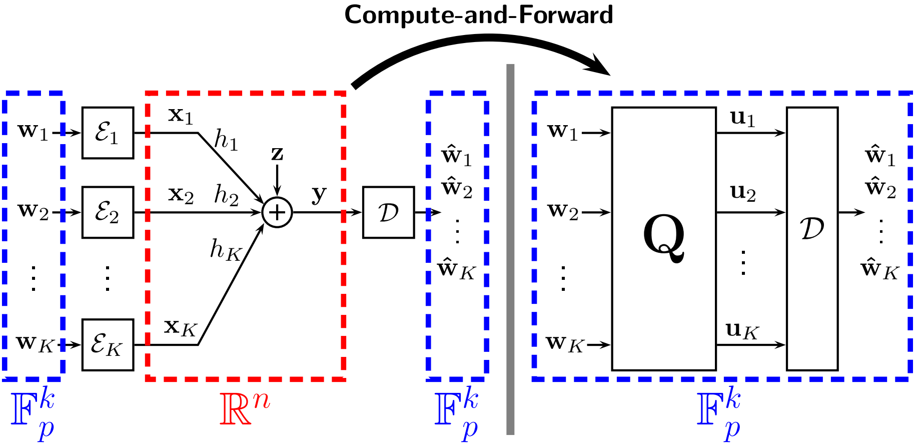


Goal: Convert noisy Gaussian networks into noiseless finite field ones.



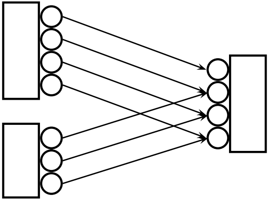
- Which linear combinations can be sent over a given channel?

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



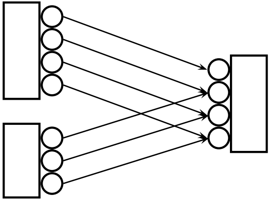
- Which linear combinations can be sent over a given channel?
- Where can this help us?

Deterministic Model from **Avestimehr-Diggavi-Tse '11:**

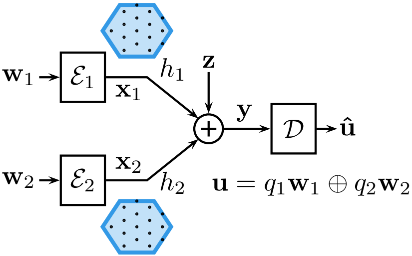


Top-Down vs. Bottom-Up

Deterministic Model from **Avestimehr-Diggavi-Tse '11:**



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.

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.
- 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

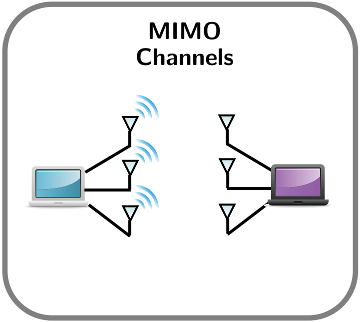
- Compute-and-Forward:
  - Achievability results for Gaussian networks.
  - Proof ideas.

Road Map

- Compute-and-Forward:
  - Achievability results for Gaussian networks.
  - Proof ideas.
- Applications to communication across single-hop Gaussian networks.

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

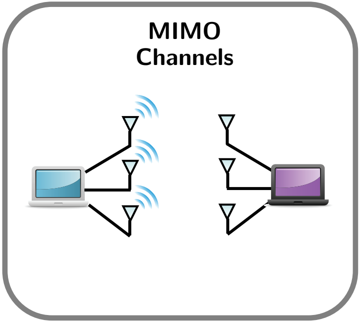
- Applications to communication across single-hop Gaussian networks.

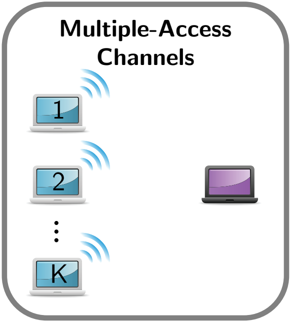


Road Map

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

- Applications to communication across single-hop Gaussian networks.

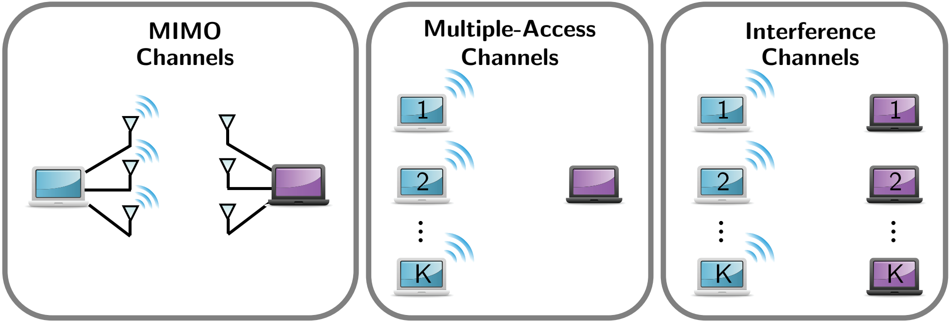




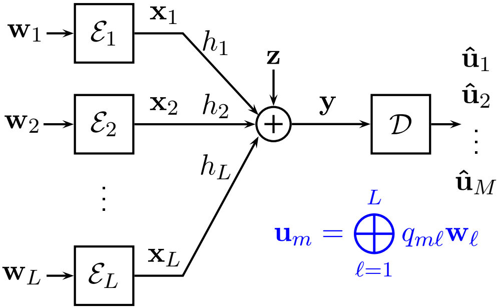
Road Map

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

- Applications to communication across single-hop Gaussian networks.



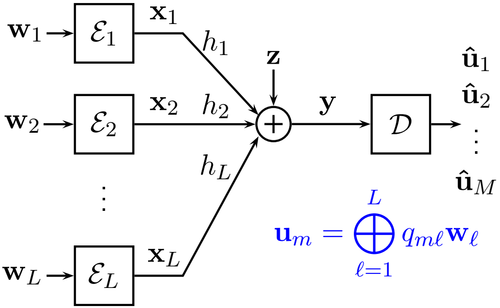
Compute-and-Forward: Problem Statement



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

- 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: $R = \dfrac{k}{n} \log_2 p$

Compute-and-Forward: Problem Statement

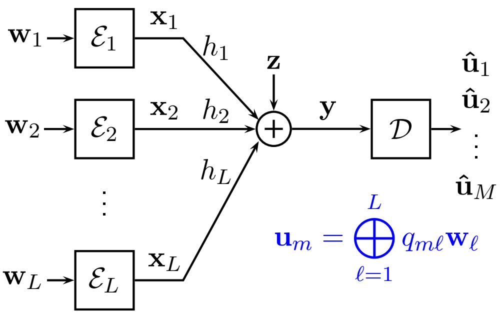


$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell$$

- 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: $R = \dfrac{k}{n}\log_2 p$

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

Compute-and-Forward: Problem Statement

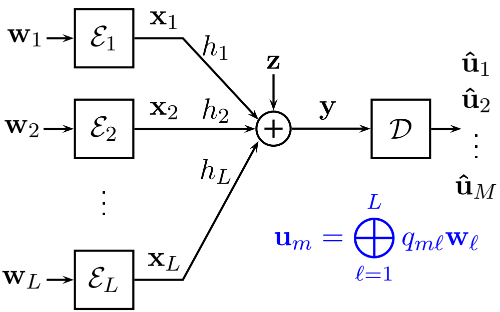


$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell$$

- 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 $\lim_{n\to\infty} \mathbb{P}\Big(\{\hat{\mathbf{u}}_1 \neq \mathbf{u}_1\} \cup \cdots \cup \{\hat{\mathbf{u}}_M \neq \hat{\mathbf{u}}_M\}\Big) = 0$.
- 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



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

- 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: $R = \dfrac{k}{n}\log_2 p$

- Decoder wants M linear combinations of the messages with vanishing probability of error $\lim\limits_{n\to\infty} \mathbb{P}\Big(\{\hat{\mathbf{u}}_1 \neq \mathbf{u}_1\} \cup \cdots \cup \{\hat{\mathbf{u}}_M \neq \hat{\mathbf{u}}_M\}\Big) = 0$.
- 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_ℓ and $q_{m\ell}$?

Computation Rate

- Want to characterize achievable rates as a function of h_ℓ and $q_{m\ell}$.

Computation Rate

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

$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell \qquad \text{where } q_{m\ell} = g^{-1}\Big([a_{m\ell}] \mod p\Big)$$

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

Computation Rate

- Want to characterize achievable rates as a function of h_ℓ and $q_{m\ell}$.
- Easier to think about integer rather than finite field coefficients.
- The linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1}\ a_{m2}\ \cdots\ a_{mL}]^{\mathsf{T}} \in \mathbb{Z}^L$ corresponds to
$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell \qquad \text{where } q_{m\ell} = g^{-1}\Big([a_{m\ell}] \mod p\Big)$$
where $g(\cdot)$ is the natural mapping between $\mathbb{F}_p$ and $\mathbb{Z}$.
- **Key Definition:** The computation rate region described by $R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a})$ 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
$$R < \min_m R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a}_m)$$

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

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

is achievable.

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

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

is achievable.

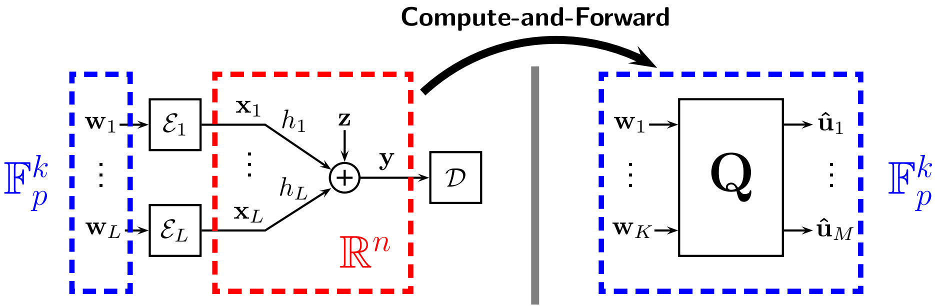
Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

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

is achievable.



if $R < \min_m R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a}_m)$ for some $\mathbf{a}_m \in \mathbb{Z}^L$ satisfying $g^{-1}\big([\mathbf{a}_m] \bmod p\big) = \mathbf{q}_m$.

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

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

is achievable.

Special Cases:

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

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

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

is achievable.

Special Cases:

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

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

Compute-and-Forward: Effective Noise

$$\begin{aligned}\mathbf{y} &= \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z} \\ &= \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell + \sum_{\ell=1}^{L} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}\end{aligned}$$

Desired Codebook:

- Closed under integer linear combinations $\implies$ lattice codebook.

$$\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$$

$$= \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell + \underbrace{\sum_{\ell=1}^{L} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}}_{\text{Effective Noise}}$$

Desired Codebook:

- Closed under integer linear combinations $\implies$ lattice codebook.
- Independent effective noise $\implies$ dithering.

Compute-and-Forward: Effective Noise

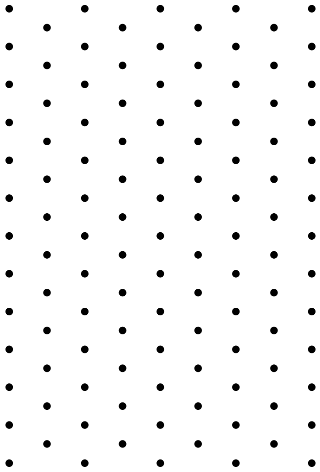
$$\begin{aligned}\mathbf{y} &= \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z} \\ &= \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell + \underbrace{\sum_{\ell=1}^{L} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}}_{\text{Effective Noise}} \xrightarrow{\text{Decode}} \bigoplus_{\ell=1}^{L} q_\ell \mathbf{w}_\ell\end{aligned}$$

Desired Codebook:

- Closed under integer linear combinations $\Longrightarrow$ lattice codebook.
- Independent effective noise $\Longrightarrow$ dithering.
- Isomorphic to $\mathbb{F}_p^k$ $\Longrightarrow$ nested lattice codebook.

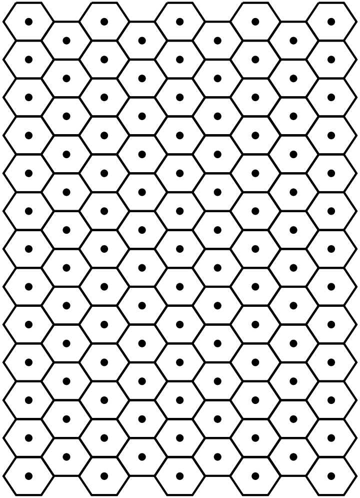
Nested Lattices

- 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:

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

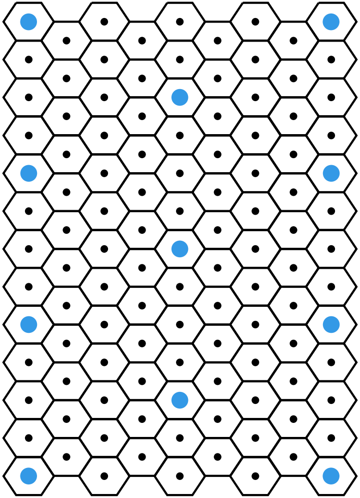
Nested Lattices

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

- Nearest neighbor quantizer:

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

- Two lattices Λ and Λ_{FINE} are nested if $\Lambda \subset \Lambda_{\mathsf{FINE}}$



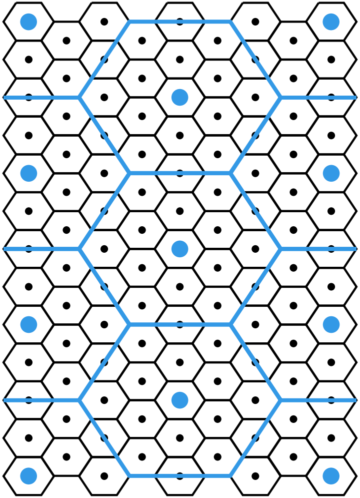
Nested Lattices

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

- Nearest neighbor quantizer:

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

- Two lattices Λ and Λ_{FINE} are nested if $\Lambda \subset \Lambda_{\mathsf{FINE}}$



Nested Lattices

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

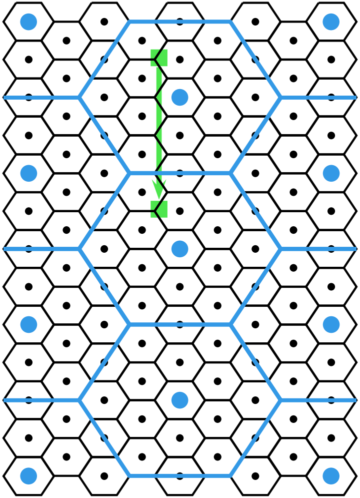
- Nearest neighbor quantizer:

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

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

- Quantization error serves as modulo operation:

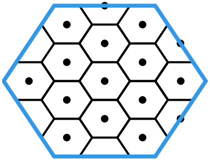
$$[\mathbf{x}] \bmod \Lambda = \mathbf{x} - Q_\Lambda(\mathbf{x}) \ .$$



Distributive Law:

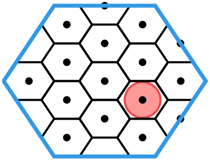
$$\big[\mathbf{x}_1 + a[\mathbf{x}_2] \bmod \Lambda\big] \bmod \Lambda = [\mathbf{x}_1 + a\mathbf{x}_2] \bmod \Lambda \quad \text{for all } a \in \mathbb{Z}.$$

- Nested Lattice Code: Formed by taking all elements of Λ_{FINE} that lie in the fundamental Voronoi region of Λ.



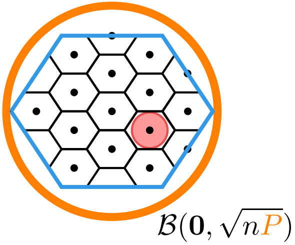
- Nested Lattice Code: Formed by taking all elements of Λ_{FINE} that lie in the fundamental Voronoi region of Λ.

- Fine lattice Λ_{FINE} protects against noise.



Nested Lattice Codes

- Nested Lattice Code: Formed by taking all elements of Λ_{FINE} that lie in the fundamental Voronoi region of Λ.
- Fine lattice Λ_{FINE} protects against noise.
- Coarse lattice Λ enforces the power constraint.



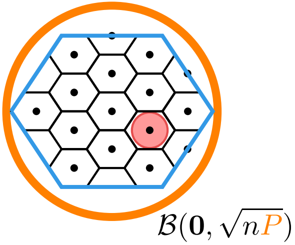
- Nested Lattice Code: Formed by taking all elements of Λ_{FINE} that lie in the fundamental Voronoi region of Λ.

- Fine lattice Λ_{FINE} protects against noise.

- Coarse lattice Λ enforces the power constraint.

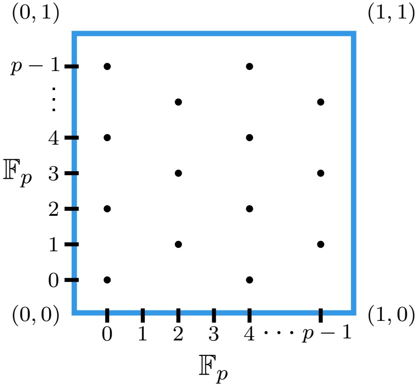
- Existence of good nested lattice codes: **Loeliger '97, Forney-Trott-Chung '00, Erez-Litsyn-Zamir '05, Ordentlich-Erez '13.**

- **Erez-Zamir '04:** Nested lattice codes can achieve the point-to-point Gaussian capacity.



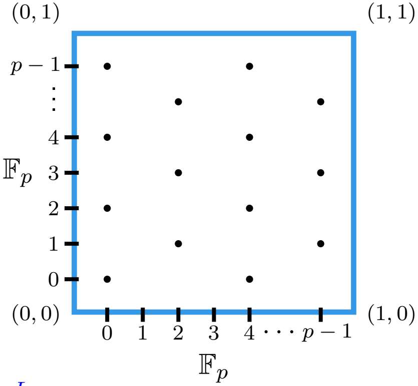
Construction A: Lattice Codes from Linear Codes

- Map elements $\{0, 1, 2, \ldots, p-1\}$ to equally spaced points on $[0, 1)$.
- Choose generator matrix $\mathbf{G} \in \mathbb{F}_p^{n \times k}$ 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 $\mathbf{t}_\ell$ and finite field messages $\mathbf{w}_\ell$:



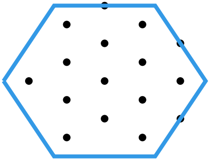
Encoding: $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$

Decoding: $\phi^{-1}\left(\left[\sum_{\ell=1}^{L} a_\ell \mathbf{t}_\ell\right] \bmod \Lambda\right) = \bigoplus_{\ell=1}^{L} q_\ell \mathbf{w}_\ell$

where $q_\ell = g^{-1}\big([a_\ell] \bmod p\big)$.

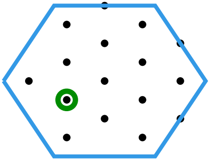
Dithering

- The effective noise will be a mixture of the channel inputs $\mathbf{x}_\ell$ and the noise $\mathbf{z}$.
- Dithering can make the effective noise look independent from the desired lattice codeword.



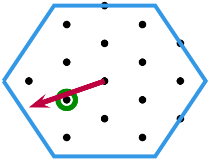
Dithering

- The effective noise will be a mixture of the channel inputs $\mathbf{x}_\ell$ and the noise $\mathbf{z}$.
- Dithering can make the effective noise look independent from the desired lattice codeword.
- Map message $\mathbf{w}_\ell$ to a lattice codeword $\mathbf{t}_\ell$.



Dithering

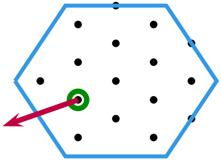
- The effective noise will be a mixture of the channel inputs $\mathbf{x}_\ell$ and the noise $\mathbf{z}$.
- Dithering can make the effective noise look independent from the desired lattice codeword.
- Map message $\mathbf{w}_\ell$ to a lattice codeword $\mathbf{t}_\ell$.
- Generate a random dither vector $\mathbf{d}_\ell$ uniformly over Voronoi region of Λ.



Dithering

- The effective noise will be a mixture of the channel inputs $\mathbf{x}_\ell$ and the noise $\mathbf{z}$.
- Dithering can make the effective noise look independent from the desired lattice codeword.
- Map message $\mathbf{w}_\ell$ to a lattice codeword $\mathbf{t}_\ell$.
- Generate a random dither vector $\mathbf{d}_\ell$ uniformly over Voronoi region of Λ.
- Transmitter sends a dithered codeword:

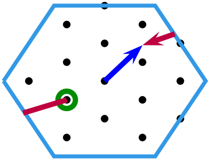
$$\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda$$



Dithering

- The effective noise will be a mixture of the channel inputs $\mathbf{x}_\ell$ and the noise $\mathbf{z}$.
- Dithering can make the effective noise look independent from the desired lattice codeword.
- Map message $\mathbf{w}_\ell$ to a lattice codeword $\mathbf{t}_\ell$.
- Generate a random dither vector $\mathbf{d}_\ell$ uniformly over Voronoi region of Λ.
- Transmitter sends a dithered codeword:

$$\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda$$

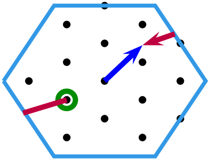


Dithering

- The effective noise will be a mixture of the channel inputs $\mathbf{x}_\ell$ and the noise $\mathbf{z}$.
- Dithering can make the effective noise look independent from the desired lattice codeword.
- Map message $\mathbf{w}_\ell$ to a lattice codeword $\mathbf{t}_\ell$.
- Generate a random dither vector $\mathbf{d}_\ell$ uniformly over Voronoi region of Λ.
- Transmitter sends a dithered codeword:

$$\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda$$

- $\mathbf{x}_\ell$ is now independent of the codeword $\mathbf{t}_\ell$.



Compute-and-Forward: Encoding and Decoding

- Map messages to lattice points, $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$.

Compute-and-Forward: Encoding and Decoding

- Map messages to lattice points, $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$.
- Transmit dithered codewords, $\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda$

Compute-and-Forward: Encoding and Decoding

- Map messages to lattice points, $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$.
- Transmit dithered codewords, $\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda$
- Receiver scales its observation by α, removes dithers, and decodes an integer-linear combination $\left[\sum a_\ell \mathbf{t}_\ell\right] \bmod \Lambda$.

$$
\begin{aligned}
\tilde{\mathbf{y}} &= \left[\alpha\Big(\sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}\Big) - \sum_{\ell=1}^{L} a_\ell \mathbf{d}_\ell\right] \bmod \Lambda \\
&= \left[\sum_{\ell=1}^{L} a_\ell(\mathbf{x}_\ell - \mathbf{d}_\ell) + \sum_{\ell=1}^{L} (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha\mathbf{z}\right] \bmod \Lambda \\
&= \left[\left[\sum_{\ell=1}^{L} a_\ell \mathbf{t}_\ell\right] \bmod \Lambda + \mathbf{z}_{\text{effec}}(\mathbf{h}, \mathbf{a})\right] \bmod \Lambda
\end{aligned}
$$

Compute-and-Forward: Encoding and Decoding

- Map messages to lattice points, $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$.
- Transmit dithered codewords, $\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda$
- Receiver scales its observation by α, removes dithers, and decodes an integer-linear combination $\left[\sum a_\ell \mathbf{t}_\ell\right] \bmod \Lambda$.

$$\begin{aligned}
\tilde{\mathbf{y}} &= \left[\alpha\Big(\sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}\Big) - \sum_{\ell=1}^{L} a_\ell \mathbf{d}_\ell\right] \bmod \Lambda \\
&= \left[\sum_{\ell=1}^{L} a_\ell(\mathbf{x}_\ell - \mathbf{d}_\ell) + \sum_{\ell=1}^{L} (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha \mathbf{z}\right] \bmod \Lambda \\
&= \left[\left[\sum_{\ell=1}^{L} a_\ell \mathbf{t}_\ell\right] \bmod \Lambda + \mathbf{z}_{\text{effec}}(\mathbf{h}, \mathbf{a})\right] \bmod \Lambda
\end{aligned}$$

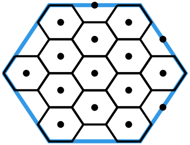
Compute-and-Forward: Encoding and Decoding

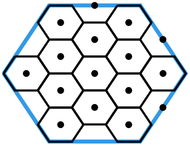
- Map messages to lattice points, $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$.
- Transmit dithered codewords, $\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda$
- Receiver scales its observation by α, removes dithers, and decodes an integer-linear combination $\left[\sum a_\ell \mathbf{t}_\ell\right] \bmod \Lambda$.

$$\begin{aligned}
\tilde{\mathbf{y}} &= \left[\alpha\Big(\sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}\Big) - \sum_{\ell=1}^{L} a_\ell \mathbf{d}_\ell\right] \bmod \Lambda \\
&= \left[\sum_{\ell=1}^{L} a_\ell(\mathbf{x}_\ell - \mathbf{d}_\ell) + \sum_{\ell=1}^{L}(\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha\mathbf{z}\right] \bmod \Lambda \\
&= \left[\Big[\sum_{\ell=1}^{L} a_\ell \mathbf{t}_\ell\Big] \bmod \Lambda + \mathbf{z}_{\text{effec}}(\mathbf{h},\mathbf{a})\right] \bmod \Lambda
\end{aligned}$$

- Apply inverse map ϕ^{-1} to integer-linear combination to obtain linear combination $\bigoplus_\ell q_\ell \mathbf{w}_\ell$ with coefficients $q_\ell = g^{-1}\big([a_\ell] \bmod p\big)$.

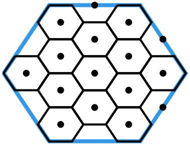
All users pick the same nested lattice code:



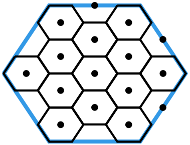


Choose message vectors over finite field $\mathbf{w}_\ell \in \mathbb{F}_p^k$:

$\mathbf{w}_1 \longrightarrow$

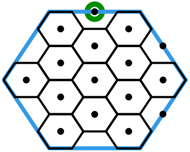


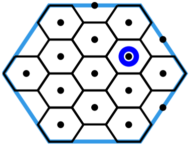
$\mathbf{w}_2 \longrightarrow$



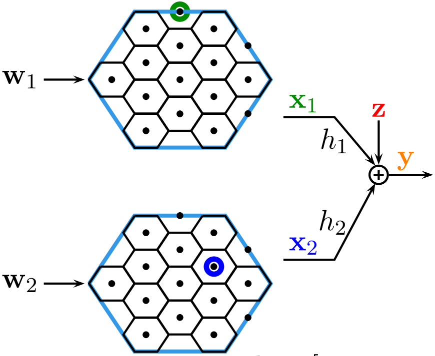
Compute-and-Forward: Illustration

Map $\mathbf{w}_\ell$ to lattice point $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$:

$\mathbf{w}_1 \longrightarrow$ 

$\mathbf{w}_2 \longrightarrow$ 

Transmit lattice points over the channel:

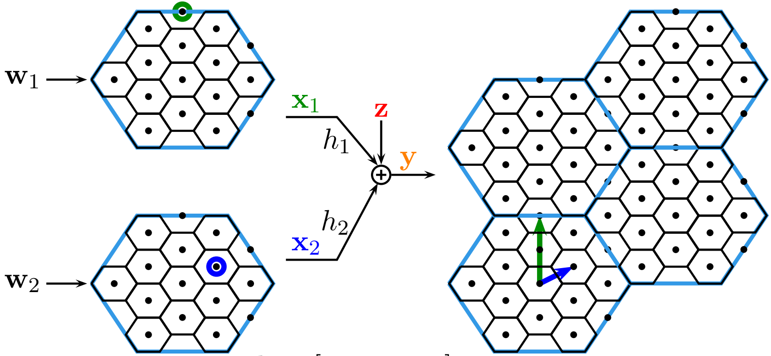


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

Transmit lattice points over the channel:

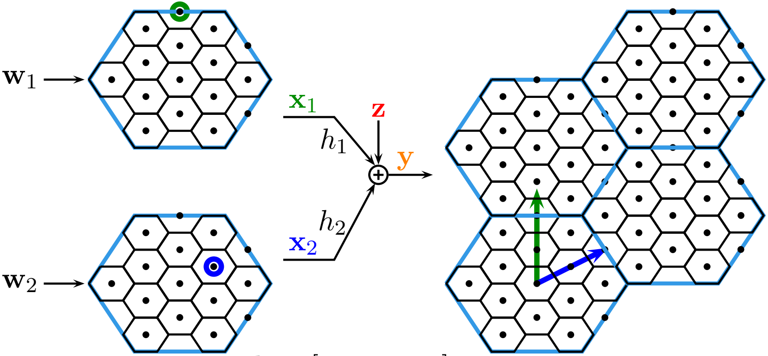


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

Lattice codewords are scaled by channel coefficients:

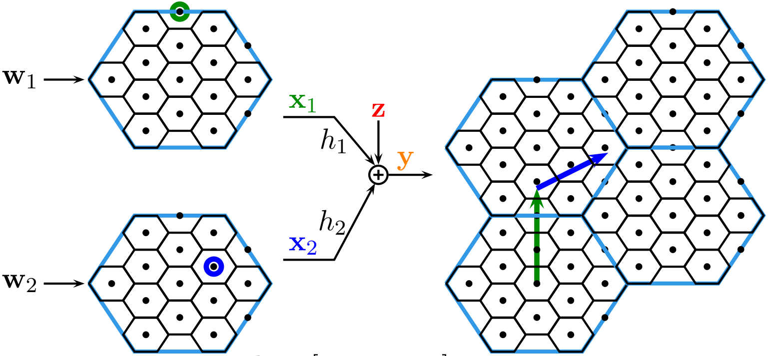


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

Scaled codewords added together plus noise:

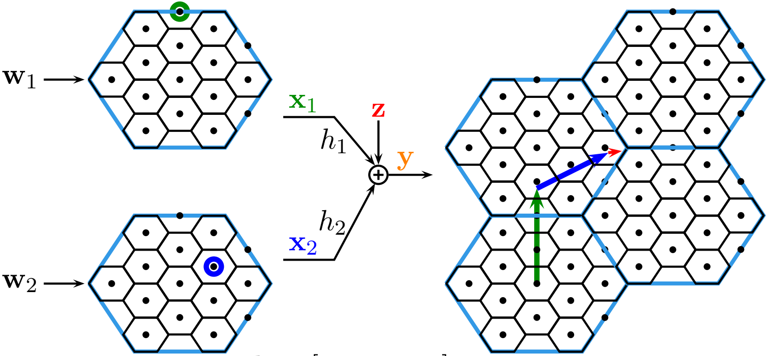


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

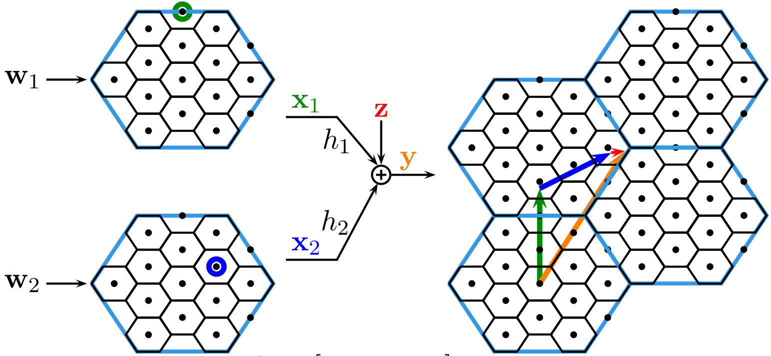
Scaled codewords added together plus noise:



$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Extra noise penalty for non-integer channel coefficients:



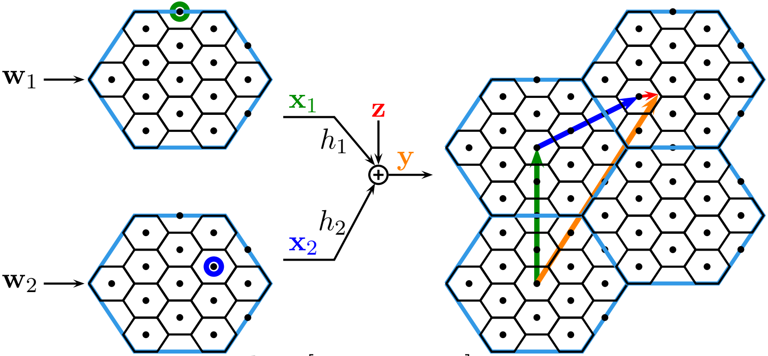
$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Effective noise: $1 + P\|\mathbf{h} - \mathbf{a}\|^2$

Compute-and-Forward: Illustration

Scale output by α to reduce non-integer noise penalty:



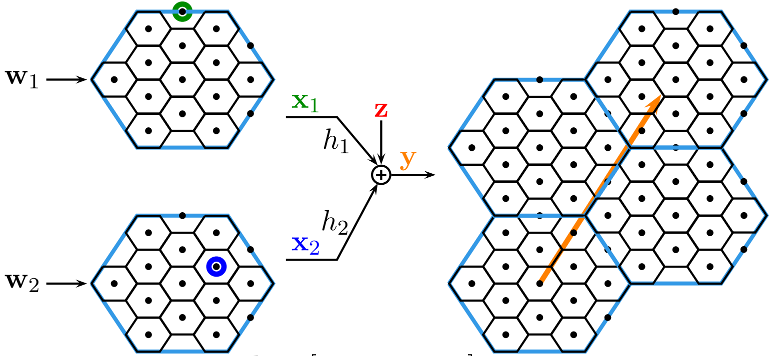
$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}\|^2$

Compute-and-Forward: Illustration

Scale output by α to reduce non-integer noise penalty:



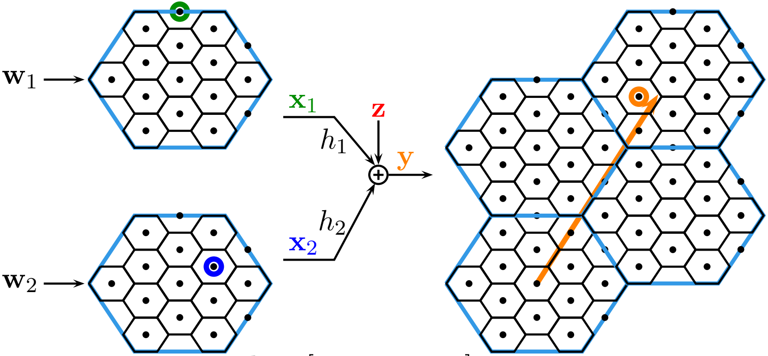
$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a} = [\ 2 \qquad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}\|^2$

Compute-and-Forward: Illustration

Decode to the closest lattice point:



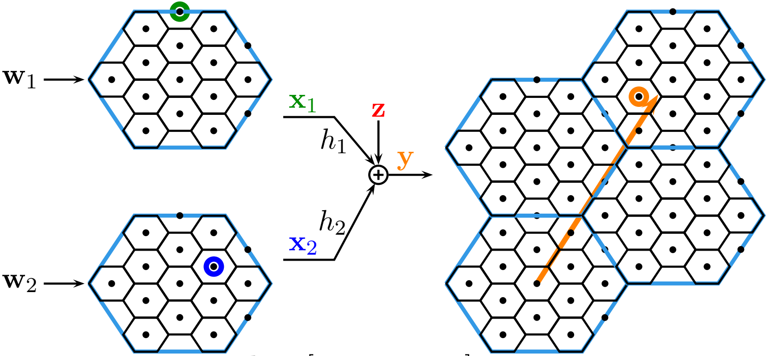
$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}\|^2$

Compute-and-Forward: Illustration

Recover integer linear combination modulo Λ:

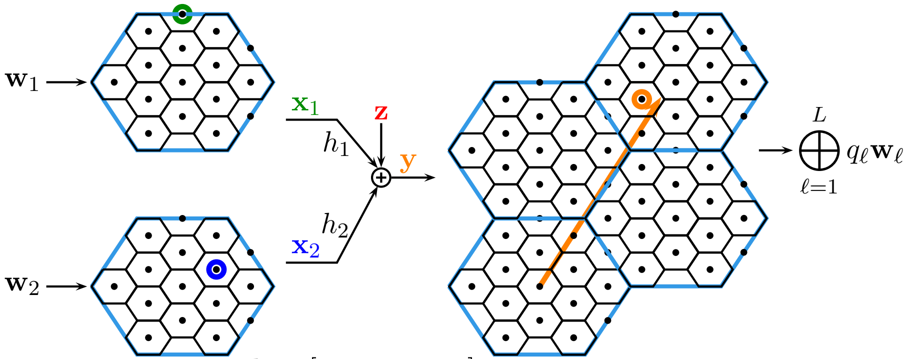


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}\|^2$

Map back to linear combination of the messages:

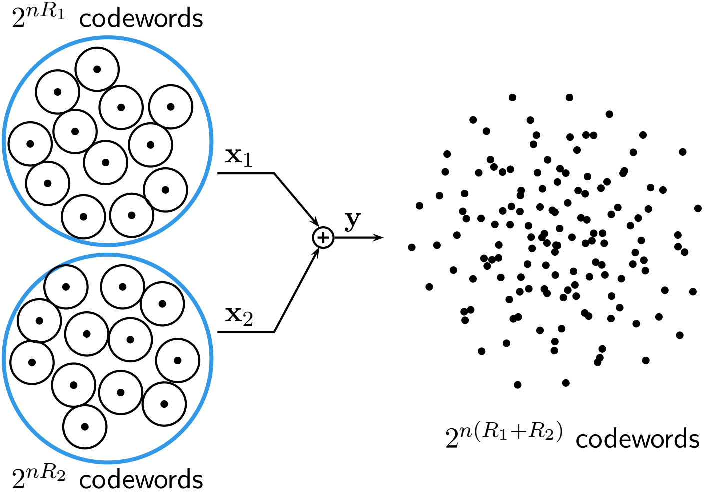


$$\alpha \mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

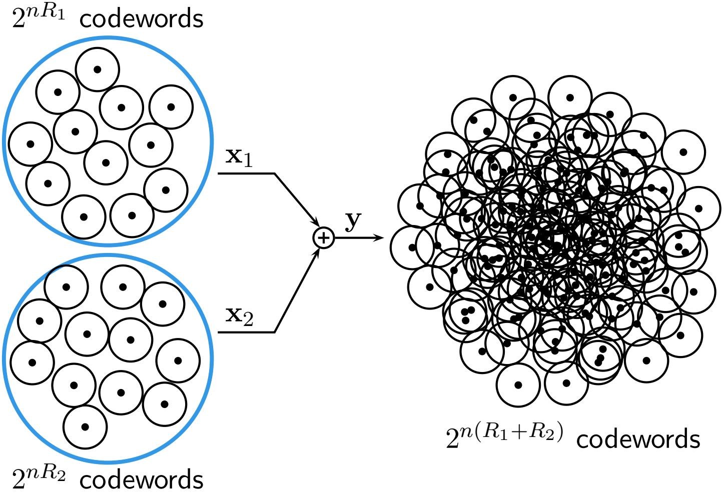
$$\mathbf{a} = [\ \ 2 \qquad 3\ \ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}\|^2$

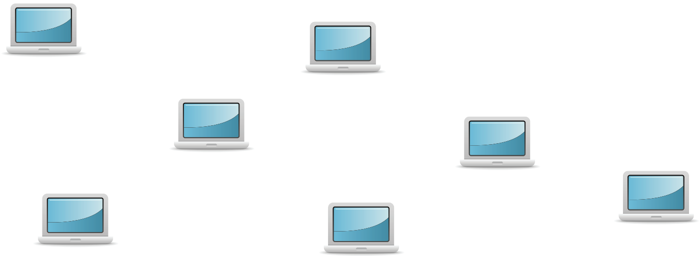
Random i.i.d. codes are not good for computation



Random i.i.d. codes are not good for computation

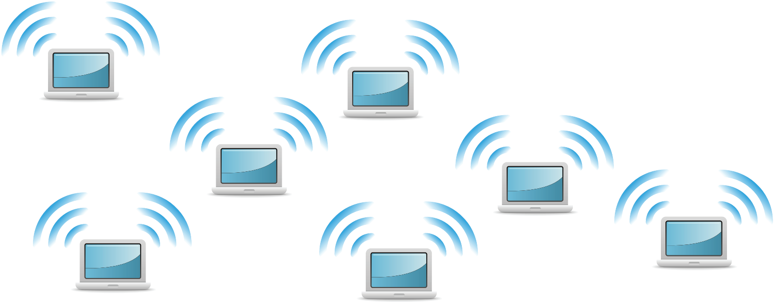


Physical-Layer Network Coding

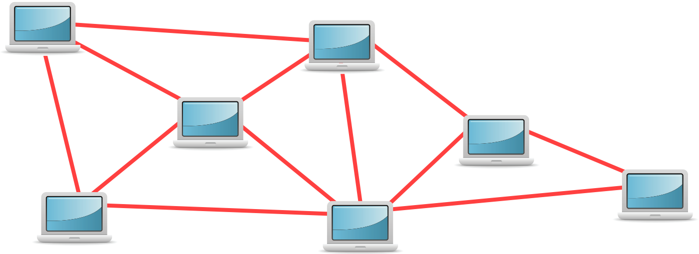


- Usually fight interference and convert to network of bit pipes.

Physical-Layer Network Coding

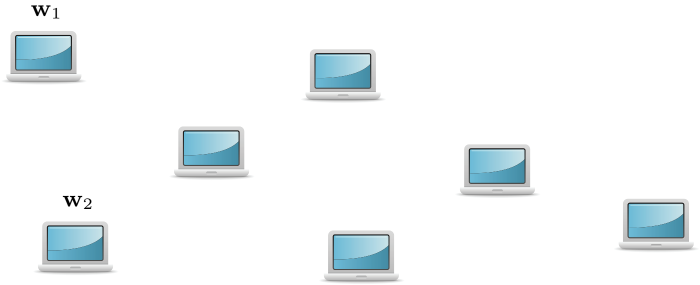


- Usually fight interference and convert to network of bit pipes.



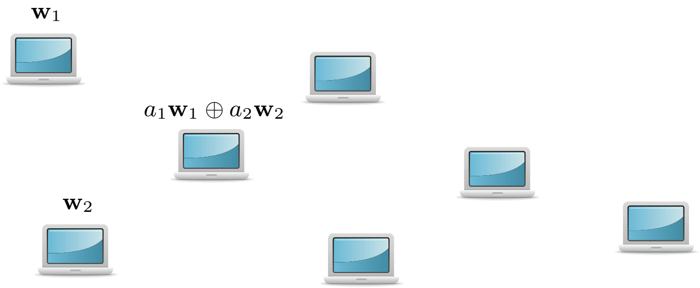
- Usually fight interference and convert to network of bit pipes.

Physical-Layer Network Coding



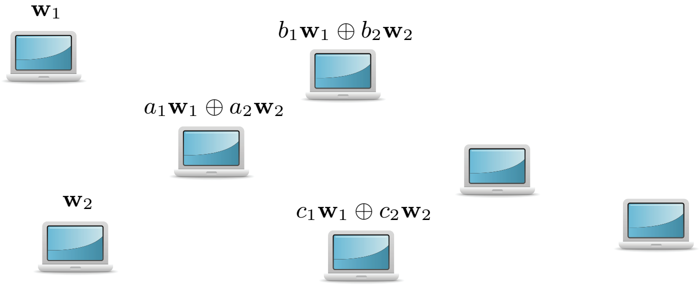
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

Physical-Layer Network Coding



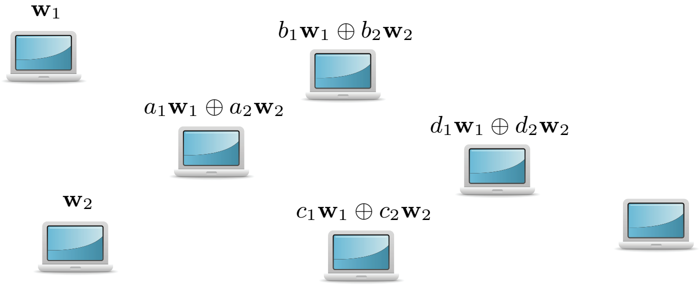
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

Physical-Layer Network Coding



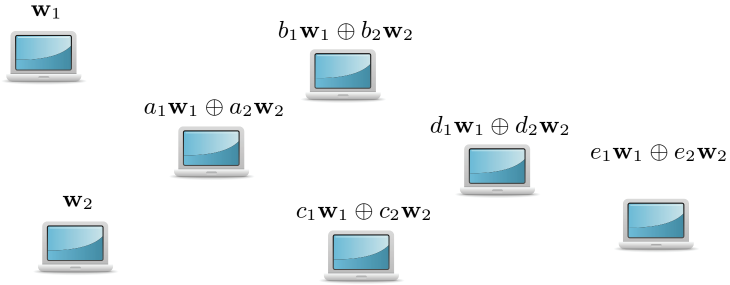
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

Physical-Layer Network Coding



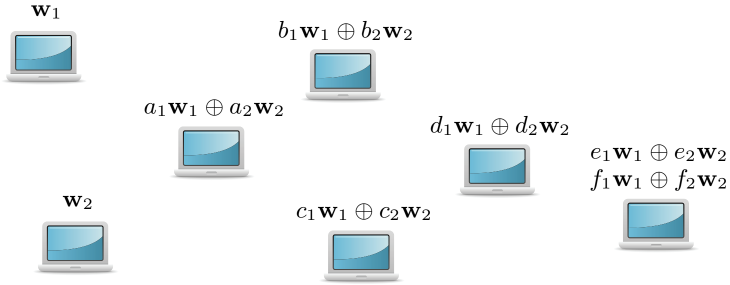
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

Physical-Layer Network Coding



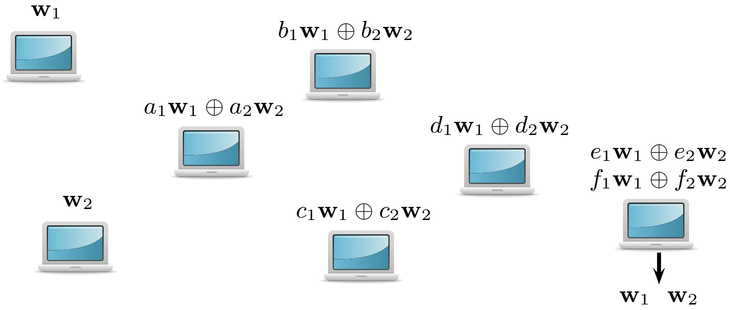
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- 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.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

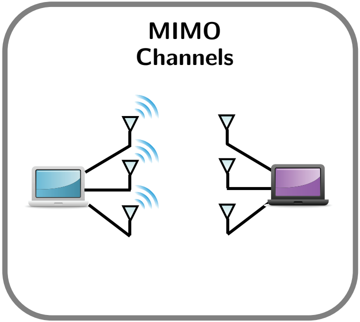
Physical-Layer Network Coding



- 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.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

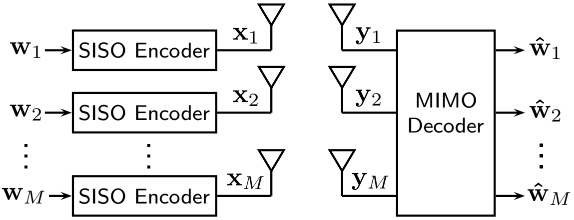
- Compute-and-Forward:
  - Achievability results for Gaussian networks.
  - Proof ideas.

- Applications to communication across single-hop Gaussian networks.



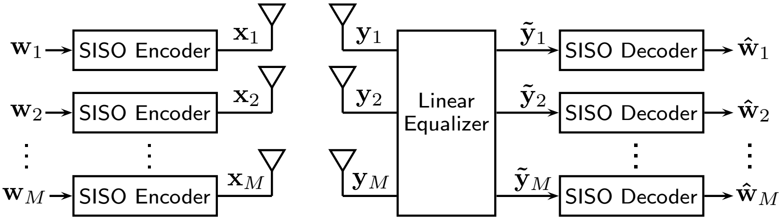
Joint work with Jiening Zhan, Uri Erez, and Michael Gastpar.

MIMO Problem Statement



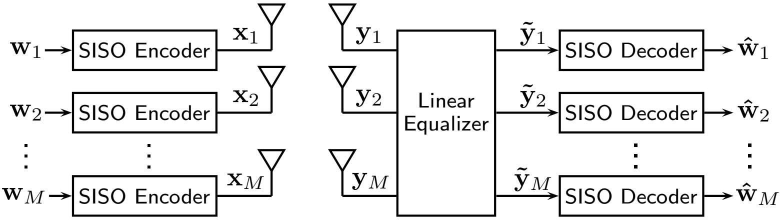
- Each antenna encodes an independent data stream of rate R (e.g., V-BLAST setting, cellular uplink).
- Channel model: $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}$ where $\mathbf{Z}$ is elementwise i.i.d. $\mathcal{N}(0, 1)$.
- Probability of error: $\mathbb{P}\big(\{\hat{\mathbf{w}}_1 \neq \mathbf{w}_1\} \cup \cdots \cup \{\hat{\mathbf{w}}_M \neq \mathbf{w}_M\}\big) < \epsilon$
- Joint maximum likelihood decoding is optimal but has high implementation complexity.

Zero-Forcing Linear Receivers

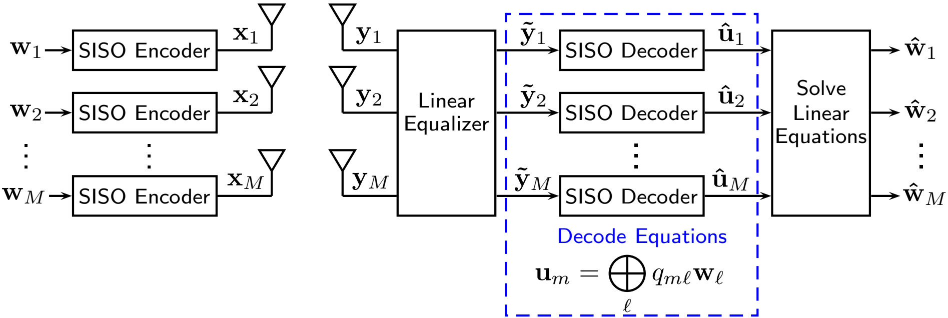


- Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to eliminate interference between data streams.
- Zero-Forcing: Set $\mathbf{B} = \mathbf{H}^{-1}$ to obtain $\tilde{\mathbf{Y}} = \mathbf{X} + \mathbf{H}^{-1}\mathbf{Z}$.
- Significantly reduces complexity at the expense of performance.

Zero-Forcing Linear Receivers

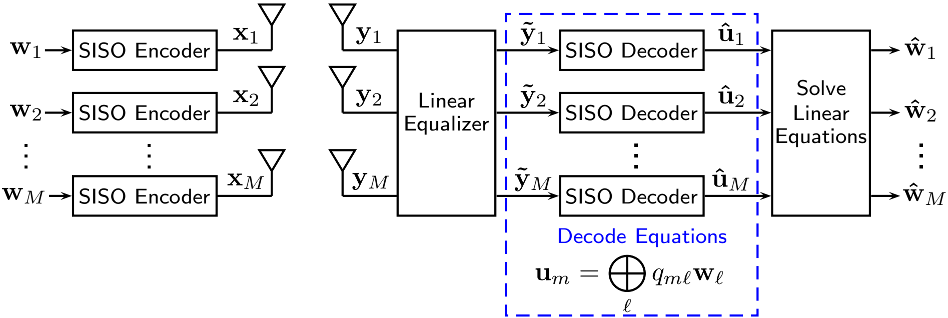


- Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{BY}$, to eliminate interference between data streams.
- Zero-Forcing: Set $\mathbf{B} = \mathbf{H}^{-1}$ to obtain $\tilde{\mathbf{Y}} = \mathbf{X} + \mathbf{H}^{-1}\mathbf{Z}$.
- Significantly reduces complexity at the expense of performance.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



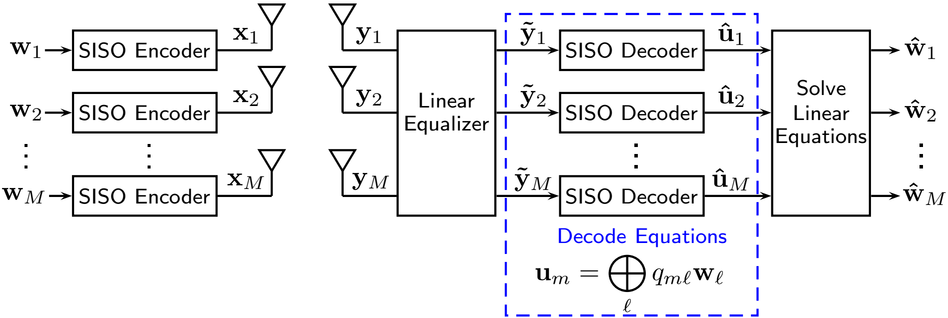
- Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to create an integer-valued effective channel matrix.
- Integer-Forcing: Set $\mathbf{B} = \mathbf{A}\mathbf{H}^{-1}$ to obtain $\tilde{\mathbf{Y}} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z}$. Solve resulting equations, **Zhan-Nazer-Erez-Gastpar '12.**
- Optimize over $\mathbf{A} \in \mathbb{Z}^{M \times M}$ to minimize effective noise.

Integer-Forcing Linear Receivers



- Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to create an integer-valued effective channel matrix.
- Integer-Forcing: Set $\mathbf{B} = \mathbf{A}\mathbf{H}^{-1}$ to obtain $\tilde{\mathbf{Y}} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z}$. Solve resulting equations, **Zhan-Nazer-Erez-Gastpar '12.**
- Optimize over $\mathbf{A} \in \mathbb{Z}^{M \times M}$ to minimize effective noise.
- Includes zero-forcing by setting $\mathbf{A} = \mathbf{I}$.

Integer-Forcing Linear Receivers



- Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to create an integer-valued effective channel matrix.
- Integer-Forcing: Set $\mathbf{B} = \mathbf{A}\mathbf{H}^{-1}$ to obtain $\tilde{\mathbf{Y}} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z}$. Solve resulting equations, **Zhan-Nazer-Erez-Gastpar '12.**
- Optimize over $\mathbf{A} \in \mathbb{Z}^{M \times M}$ to minimize effective noise.
- Includes zero-forcing by setting $\mathbf{A} = \mathbf{I}$.
- Slight improvement possible by using an MMSE projection instead.

- Receiver has an M-dimensional observation of each transmitted symbol:

$$\begin{aligned}\mathbf{Y} &= \sum_{m=1}^{M} \mathbf{h}_m \mathbf{x}_m^{\mathsf{T}} + \mathbf{Z} \\ &= \mathbf{H}\mathbf{X} + \mathbf{Z}\end{aligned}$$

MIMO Compute-and-Forward

- Receiver has an M-dimensional observation of each transmitted symbol:

$$\begin{aligned}\mathbf{Y} &= \sum_{m=1}^{M} \mathbf{h}_m \mathbf{x}_m^{\mathsf{T}} + \mathbf{Z} \\ &= \mathbf{H}\mathbf{X} + \mathbf{Z}\end{aligned}$$

- To recover the linear combination with integer coefficient vector $\mathbf{a} \in \mathbb{Z}^L$, the receiver projects its observation:

$$\mathbf{b}^{\mathsf{T}}\mathbf{Y} = \mathbf{a}^{\mathsf{T}}\mathbf{X} + (\mathbf{b}^{\mathsf{T}}\mathbf{H} - \mathbf{a}^{\mathsf{T}})\mathbf{X} + \mathbf{b}^{\mathsf{T}}\mathbf{Z}$$

MIMO Compute-and-Forward

- Receiver has an M-dimensional observation of each transmitted symbol:

$$\begin{aligned}\mathbf{Y} &= \sum_{m=1}^{M} \mathbf{h}_m \mathbf{x}_m^{\mathsf{T}} + \mathbf{Z} \\ &= \mathbf{H}\mathbf{X} + \mathbf{Z}\end{aligned}$$

- To recover the linear combination with integer coefficient vector $\mathbf{a} \in \mathbb{Z}^L$, the receiver projects its observation:

$$\mathbf{b}^{\mathsf{T}}\mathbf{Y} = \mathbf{a}^{\mathsf{T}}\mathbf{X} + (\mathbf{b}^{\mathsf{T}}\mathbf{H} - \mathbf{a}^{\mathsf{T}})\mathbf{X} + \mathbf{b}^{\mathsf{T}}\mathbf{Z}$$

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

The computation rate region described by

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

is achievable.

MIMO Compute-and-Forward

- Receiver has an M-dimensional observation of each transmitted symbol:

$$\begin{aligned}\mathbf{Y} &= \sum_{m=1}^{M} \mathbf{h}_m \mathbf{x}_m^{\mathsf{T}} + \mathbf{Z} \\ &= \mathbf{H}\mathbf{X} + \mathbf{Z}\end{aligned}$$

- To recover the linear combination with integer coefficient vector $\mathbf{a} \in \mathbb{Z}^L$, the receiver projects its observation:

$$\mathbf{b}^{\mathsf{T}}\mathbf{Y} = \mathbf{a}^{\mathsf{T}}\mathbf{X} + (\mathbf{b}^{\mathsf{T}}\mathbf{H} - \mathbf{a}^{\mathsf{T}})\mathbf{X} + \mathbf{b}^{\mathsf{T}}\mathbf{Z}$$

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

The computation rate region described by

$$R_{comp}(\mathbf{H}, \mathbf{a}) = \frac{1}{2} \log^{+} \left(\frac{P}{\mathbf{a}^{\mathsf{T}} \left(P^{-1}\mathbf{I} + \mathbf{H}\mathbf{H}^{\mathsf{T}} \right)^{-1} \mathbf{a}} \right)$$

is achievable.

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \quad \xrightarrow{\text{Project}} \quad \mathbf{A}\mathbf{H}^{-1}\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z} \quad \xrightarrow{\text{Decode}} \quad \mathbf{A}\mathbf{X}$$

Comparison

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \quad \xrightarrow{\text{Project}} \quad \mathbf{A}\mathbf{H}^{-1}\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z} \quad \xrightarrow{\text{Decode}} \quad \mathbf{A}\mathbf{X}$$

- Achievable rate: $R_{\text{IF}}(\mathbf{H}) = M \max_{\substack{\mathbf{A} \in \mathbb{Z}^{M \times M} \\ \text{rank}(\mathbf{A}) = \text{M}}} \min_{m} R_{\mathsf{comp}}(\mathbf{H}, \mathbf{a}_m)$

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{A}\mathbf{H}^{-1}\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z} \xrightarrow{\text{Decode}} \mathbf{A}\mathbf{X}$$

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

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{A}\mathbf{H}^{-1}\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z} \xrightarrow{\text{Decode}} \mathbf{A}\mathbf{X}$$

- Achievable rate: $R_{\mathrm{IF}}(\mathbf{H}) = M \max_{\substack{\mathbf{A} \in \mathbb{Z}^{M \times M} \\ \mathrm{rank}(\mathbf{A}) = \mathrm{M}}} \min_{m} R_{\mathsf{comp}}(\mathbf{H}, \mathbf{a}_m)$
- Only need to search over vectors satisfying $\|\mathbf{a}_m\|^2 \leq 1 + P\lambda^2_{\max}(\mathbf{H})$.
- Faster search: Apply LLL algorithm to $\mathbf{F} = \left(P^{-1}\mathbf{I} + \ \mathbf{H}\mathbf{H}^{\mathsf{T}}\right)^{-1/2}$.

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{A}\mathbf{H}^{-1}\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z} \xrightarrow{\text{Decode}} \mathbf{A}\mathbf{X}$$

- Achievable rate: $R_{\text{IF}}(\mathbf{H}) = M \max_{\substack{\mathbf{A} \in \mathbb{Z}^{M \times M} \\ \text{rank}(\mathbf{A}) = \text{M}}} \min_{m} R_{\text{comp}}(\mathbf{H}, \mathbf{a}_m)$
- Only need to search over vectors satisfying $\|\mathbf{a}_m\|^2 \leq 1 + P\lambda_{\max}^2(\mathbf{H})$.
- Faster search: Apply LLL algorithm to $\mathbf{F} = \left(P^{-1}\mathbf{I} + \mathbf{H}\mathbf{H}^{\mathsf{T}}\right)^{-1/2}$.

Zero-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{H}^{-1}\mathbf{Y} = \mathbf{X} + \mathbf{H}^{-1}\mathbf{Z} \xrightarrow{\text{Decode}} \mathbf{X}$$

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{A}\mathbf{H}^{-1}\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z} \xrightarrow{\text{Decode}} \mathbf{A}\mathbf{X}$$

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

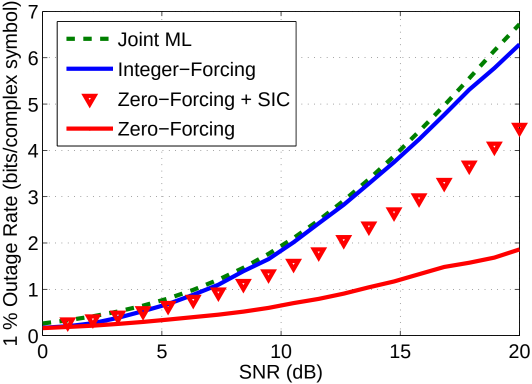
Zero-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{H}^{-1}\mathbf{Y} = \mathbf{X} + \mathbf{H}^{-1}\mathbf{Z} \xrightarrow{\text{Decode}} \mathbf{X}$$

- Achievable rate: $R_{\text{ZF}}(\mathbf{H}) = M \min_m R_{\textsf{comp}}\big(\mathbf{H}, [\underbrace{0 \ \cdots \ 0}_{m-1 \text{ zeros}} \ 1 \ 0 \ \cdots \ 0]^{\mathsf{T}}\big)$

- Channel matrix $\mathbf{H}$ is i.i.d. Rayleigh, only known at the receiver.
- For a fixed probability ρ, the *outage rate* is defined to be

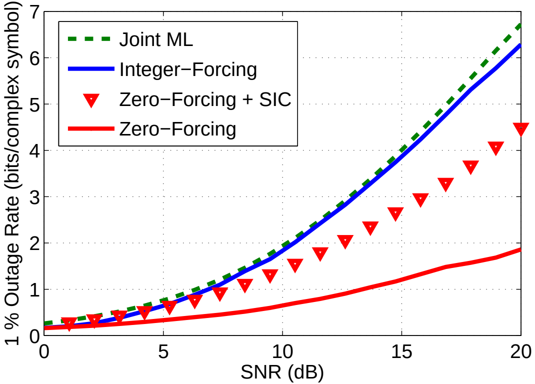
$$R_{\mathsf{OUT}}(\rho) = \sup \left\{ \mathsf{R} : \Pr\left(R(\mathbf{H}) < \mathsf{R} \right) \leq \rho \right\}.$$



- Channel matrix $\mathbf{H}$ is i.i.d. Rayleigh, only known at the receiver.
- For a fixed probability ρ, the *outage rate* is defined to be

$$R_{\mathsf{OUT}}(\rho) = \sup\left\{\mathsf{R} : \Pr\left(R(\mathbf{H}) < \mathsf{R}\right) \leq \rho\right\}.$$

Outage Rates

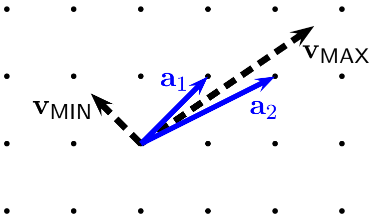


- Channel matrix $\mathbf{H}$ is i.i.d. Rayleigh, only known at the receiver.
- For a fixed probability ρ, the *outage rate* is defined to be

$$R_{\mathsf{OUT}}(\rho) = \sup\left\{\mathsf{R} : \Pr\left(R(\mathbf{H}) < \mathsf{R}\right) \leq \rho\right\}.$$

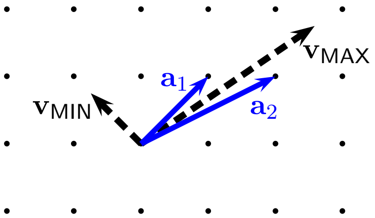
- Integer-forcing beats zero-forcing (even if it is augmented with successive interference cancellation (SIC)).

- Integer-forcing can adapt to the channel by choosing a basis (of integer vectors) close to the maximum singular vector.

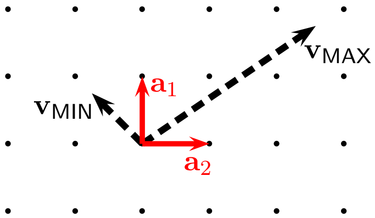


Integer-Forcing

- 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.



Integer-Forcing



Zero-Forcing

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

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

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

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

$$\lim_{\mathsf{SNR}\to\infty} \frac{R(\mathsf{SNR})}{\log \mathsf{SNR}} \geq r$$
$$\lim_{\mathsf{SNR}\to\infty} \frac{\log p_{\mathsf{error}}(\mathsf{SNR})}{\log \mathsf{SNR}} \leq -d.$$

- **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

$$\lim_{\mathsf{SNR}\to\infty} \frac{R(\mathsf{SNR})}{\log \mathsf{SNR}} \geq r$$
$$\lim_{\mathsf{SNR}\to\infty} \frac{\log p_{\mathsf{error}}(\mathsf{SNR})}{\log \mathsf{SNR}} \leq -d.$$

- **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

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

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

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

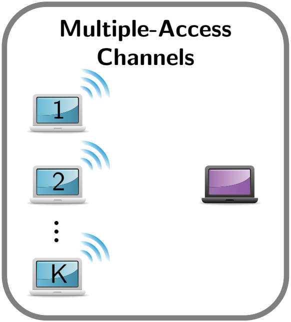
- What about space-time coding at the transmitter?

- **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

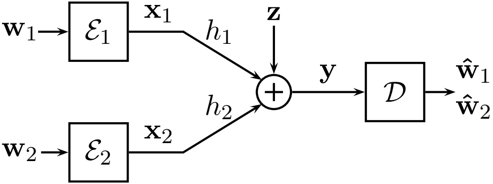
- Compute-and-Forward:
  - Achievability results for Gaussian networks.
  - Proof ideas.

- Applications to communication across single-hop Gaussian networks.



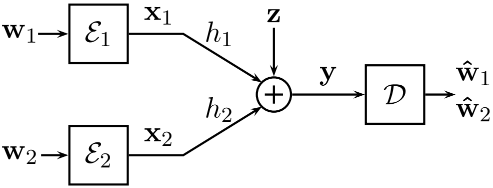
Joint work with Or Ordentlich and Uri Erez.

Gaussian Multiple-Access Channel

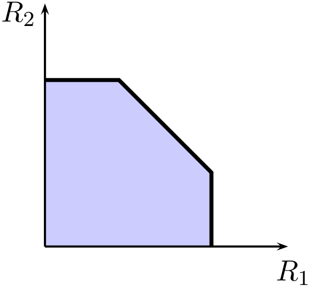


$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Gaussian Multiple-Access Channel



$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$



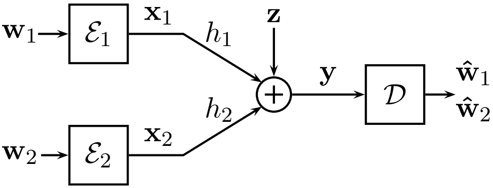
Theorem (Ahlswede '71, Liao '72, Wyner '74, Cover '75)

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

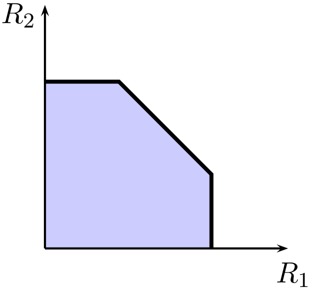
$$R_1 < \frac{1}{2}\log(1 + h_1^2 P) \qquad R_2 < \frac{1}{2}\log(1 + h_2^2 P)$$

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

Gaussian Multiple-Access Channel



$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$



Theorem (Ahlswede '71, Liao '72, Wyner '74, Cover '75)

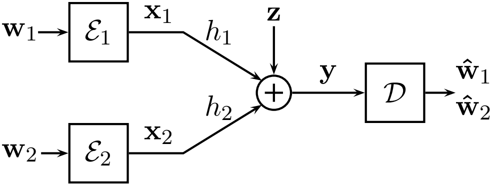
The capacity region is the set of all rate pairs (R_1, R_2) *satisfying:*

$$R_1 < \frac{1}{2}\log(1 + h_1^2 P) \qquad R_2 < \frac{1}{2}\log(1 + h_2^2 P)$$

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

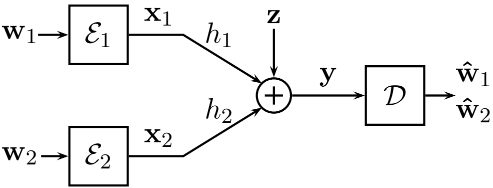
Achievable via joint decoding.

Successive Cancellation

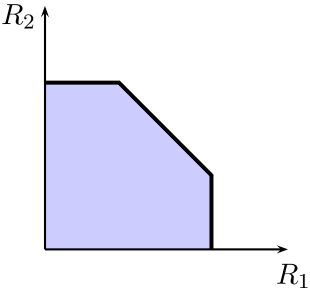


$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Successive Cancellation

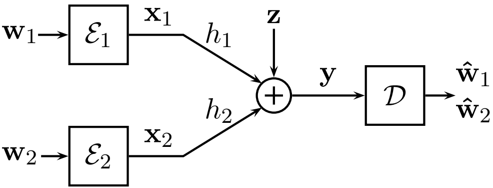


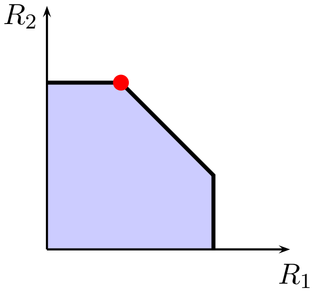
$$\|\mathbf{x}_\ell\|^2 \le nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$



- Treat $\mathbf{x}_2$ as noise and decode $\mathbf{x}_1$, $R_1 < \dfrac{1}{2}\log\left(1 + \dfrac{h_1^2 P}{1 + h_2^2 P}\right)$.

Successive Cancellation

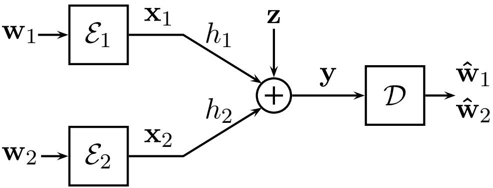




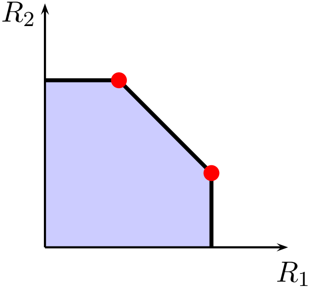
$$\|\mathbf{x}_\ell\|^2 \le nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

- Treat $\mathbf{x}_2$ as noise and decode $\mathbf{x}_1$, $R_1 < \frac{1}{2} \log \left(1 + \frac{h_1^2 P}{1 + h_2^2 P}\right)$.
- Cancel $\mathbf{x}_1$ and decode $\mathbf{x}_2$, $R_2 < \frac{1}{2} \log \left(1 + h_2^2 P\right)$.

Successive Cancellation

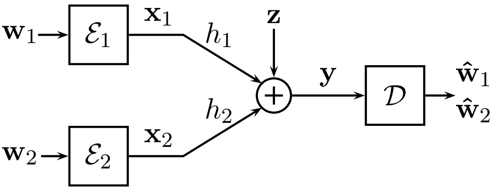


$$\|\mathbf{x}_\ell\|^2 \le nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

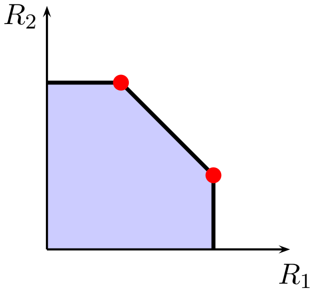


- Treat $\mathbf{x}_2$ as noise and decode $\mathbf{x}_1$, $R_1 < \dfrac{1}{2}\log\left(1 + \dfrac{h_1^2 P}{1 + h_2^2 P}\right)$.
- Cancel $\mathbf{x}_1$ and decode $\mathbf{x}_2$, $R_2 < \frac{1}{2}\log\left(1 + h_2^2 P\right)$.
- Switch decoding order for the other corner point.

Successive Cancellation

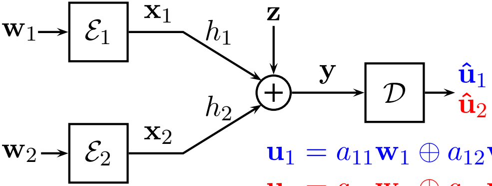


$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$



- Treat $\mathbf{x}_2$ as noise and decode $\mathbf{x}_1$, $R_1 < \frac{1}{2}\log\left(1 + \frac{h_1^2 P}{1 + h_2^2 P}\right)$.
- Cancel $\mathbf{x}_1$ and decode $\mathbf{x}_2$, $R_2 < \frac{1}{2}\log\left(1 + h_2^2 P\right)$.
- Switch decoding order for the other corner point.
- Achieves capacity when combined with time-sharing or rate-splitting (**Rimoldi-Urbanke '96**).

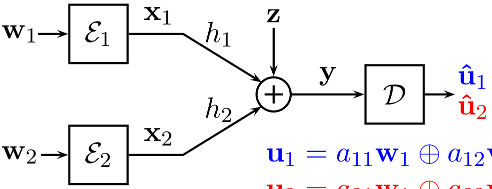
Two Linear Combinations



$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

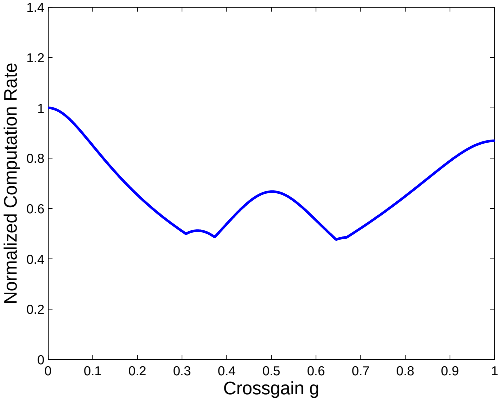
- Decode two linearly independent equations.

Two Linear Combinations

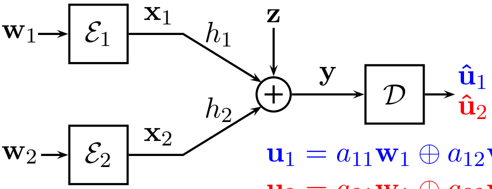


$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Decode two linearly independent equations.

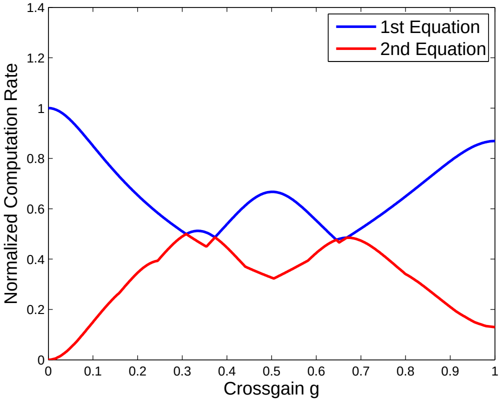


Two Linear Combinations

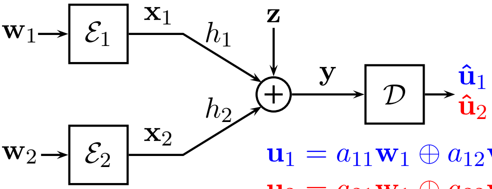


$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Decode two linearly independent equations.

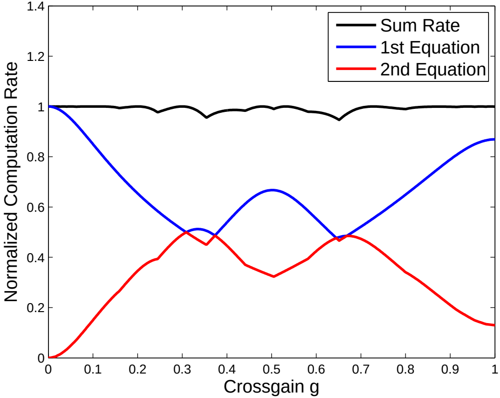


Two Linear Combinations

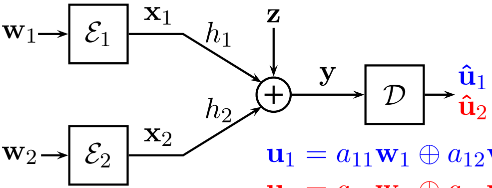


$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Decode two linearly independent equations.

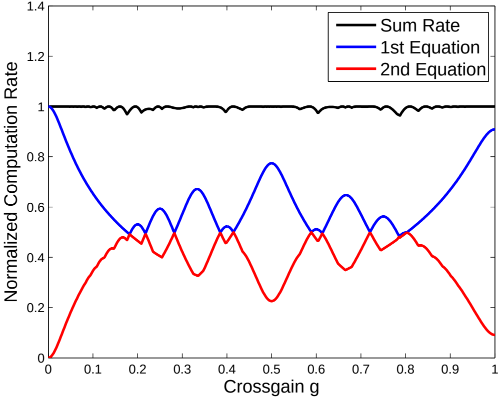


Two Linear Combinations

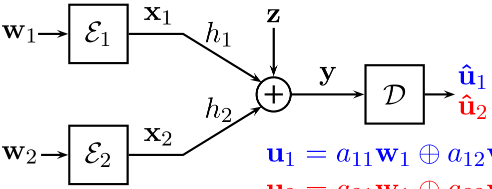


$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Decode two linearly independent equations.



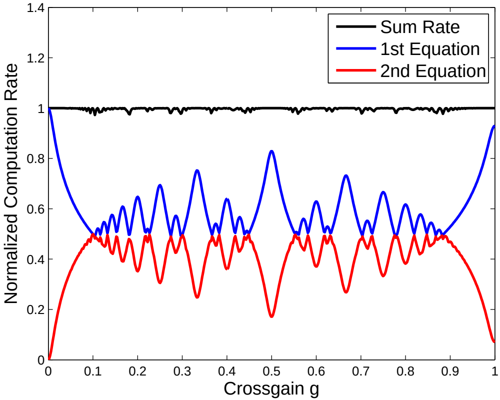
Two Linear Combinations



$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$

$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Decode two linearly independent equations.



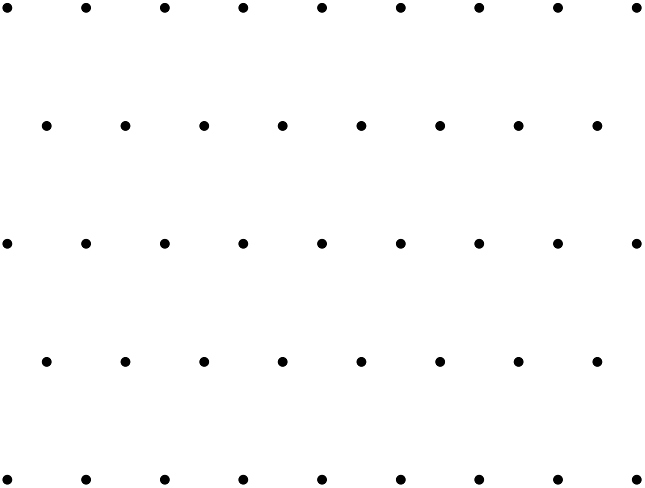
Sum of Computation Rates

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

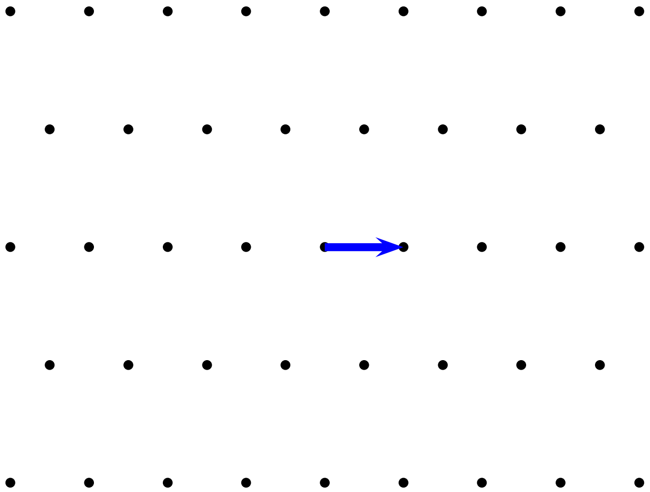
- Let $\mathbf{F} = \left(P^{-1/2}\mathbf{I} + \ \mathbf{h}\mathbf{h}^\mathsf{T}\right)^{-1/2}$. Then, each computation rate can be written as
$$R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a}_k) = \frac{1}{2}\log^+\left(\frac{P}{\|\mathbf{F}\ \mathbf{a}_k\|^2}\right) .$$

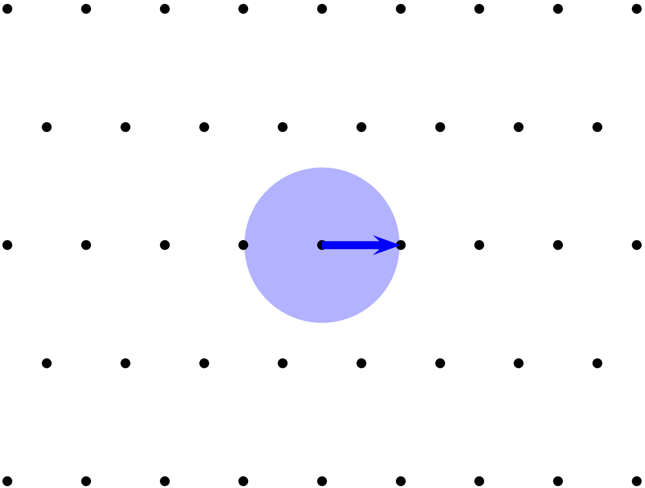
- Thus, decoding the best linear combinations is the same as finding the successive minima $\lambda_k(\mathbf{F})$ for the lattice $\Lambda(\mathbf{F}) = \mathbf{F}\mathbb{Z}^K$:

$$\lambda_k(\mathbf{F}) \triangleq \inf\left\{r : \dim\left(\operatorname{span}\left(\Lambda(\mathbf{F})\ \cap\ \mathcal{B}(\mathbf{0}, r)\right)\right) \geq k\right\} .$$

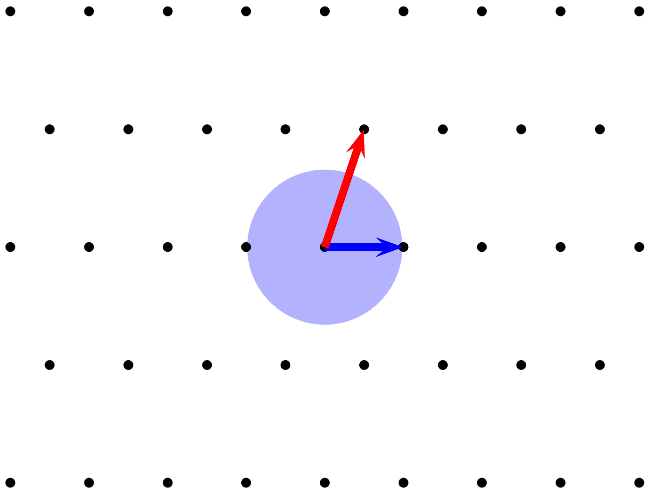


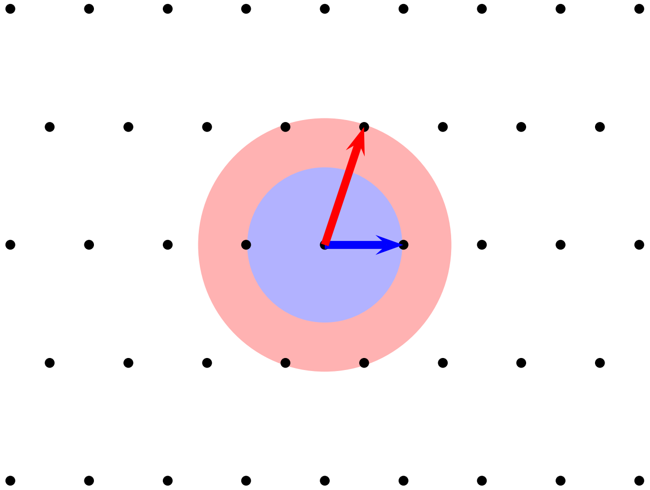
Successive Minima





Successive Minima





Minkowski's Theorem on 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

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

Minkowski's Theorem on 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

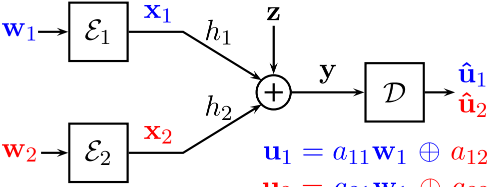
$$\prod_{k=1}^{K} \lambda_k^2(\mathbf{F}) \leq K^K \big| \det(\mathbf{F}) \big|^2 \ .$$

Theorem (Ordentlich-Erez-Nazer '12)

The sum of the K best linearly independent computation rates satisfies

$$\sum_{k=1}^{K} R_{comp}(\mathbf{h}, \mathbf{a}_k) \geq \frac{1}{2} \log(1 + \|\mathbf{h}\|^2 \mathsf{SNR}) - \frac{K}{2} \log K \ .$$

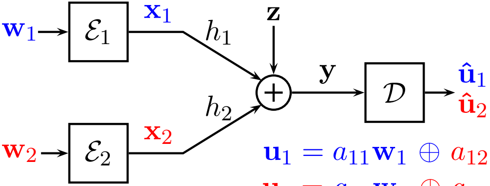
Operational Interpretation: Multiple-Access



$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Associate each computation rate to a message.

Operational Interpretation: Multiple-Access



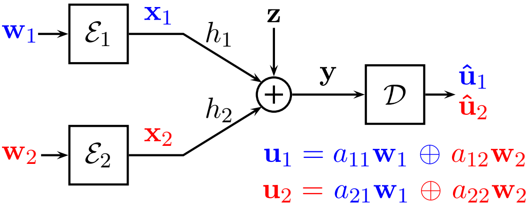
$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Associate each computation rate to a message.

- Decoding the first equation succeeds since

$$\max(R_1, R_2) < R_{\text{comp}}(\mathbf{h}, \mathbf{a}_1) \ .$$

Operational Interpretation: Multiple-Access



- Associate each computation rate to a message.

- Decoding the first equation succeeds since

$$\max(R_1, R_2) < R_{\text{comp}}(\mathbf{h}, \mathbf{a}_1) \ .$$

- Decoding the second equation runs into an issue:

$$\max(R_1, R_2) > R_{\text{comp}}(\mathbf{h}, \mathbf{a}_2) \ .$$

(Algebraic) Successive Cancellation

- After decoding the first equation, the receiver knows

$$\mathbf{v}_1 = \left[a_{11}\mathbf{t}_1 + a_{12}\mathbf{t}_2\right] \mod \Lambda \ .$$

(Algebraic) Successive Cancellation

- After decoding the first equation, the receiver knows

$$\mathbf{v}_1 = \big[a_{11}\mathbf{t}_1 + a_{12}\mathbf{t}_2\big] \mod \Lambda \ .$$

- The effective channel for the second equation is

$$\tilde{\mathbf{y}}_2 = \big[a_{21}\mathbf{t}_1 + a_{22}\mathbf{t}_2 + \mathbf{z}_{\mathsf{effec}}(\mathbf{h}, \mathbf{a}_2)\big] \mod \Lambda \ .$$

(Algebraic) Successive Cancellation

- After decoding the first equation, the receiver knows

$$\mathbf{v}_1 = \left[a_{11}\mathbf{t}_1 + a_{12}\mathbf{t}_2\right] \mod \Lambda \ .$$

- The effective channel for the second equation is

$$\tilde{\mathbf{y}}_2 = \left[a_{21}\mathbf{t}_1 + a_{22}\mathbf{t}_2 + \mathbf{z}_{\mathsf{effec}}(\mathbf{h}, \mathbf{a}_2)\right] \mod \Lambda \ .$$

- Using $\mathbf{v}_1$ we can cancel out $\mathbf{t}_1$ from $\tilde{\mathbf{y}}_2$ without changing the effective noise.

$$\begin{aligned} \tilde{\mathbf{y}}_2^{\mathsf{SI}} &= \Big[\mathbf{s}_2 - b_1\mathbf{v}_1\Big] \bmod \Lambda \\ &= \Big[\big(a_{22} - b_1 a_{12}\big)\mathbf{t}_2 + \mathbf{z}_{\mathsf{effec}}(\mathbf{h}, \mathbf{a}_2)\Big] \bmod \Lambda \ . \end{aligned}$$

(Algebraic) Successive Cancellation

- After decoding the first equation, the receiver knows

$$\mathbf{v}_1 = \left[a_{11}\mathbf{t}_1 + a_{12}\mathbf{t}_2\right] \mod \Lambda \ .$$

- The effective channel for the second equation is

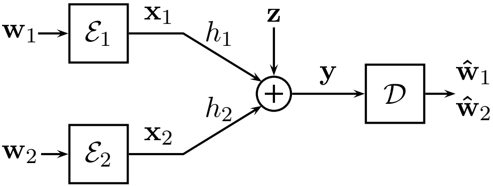
$$\tilde{\mathbf{y}}_2 = \left[a_{21}\mathbf{t}_1 + a_{22}\mathbf{t}_2 + \mathbf{z}_{\mathsf{effec}}(\mathbf{h}, \mathbf{a}_2)\right] \mod \Lambda \ .$$

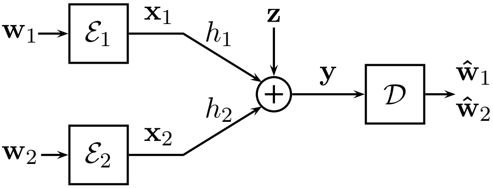
- Using $\mathbf{v}_1$ we can cancel out $\mathbf{t}_1$ from $\tilde{\mathbf{y}}_2$ without changing the effective noise.

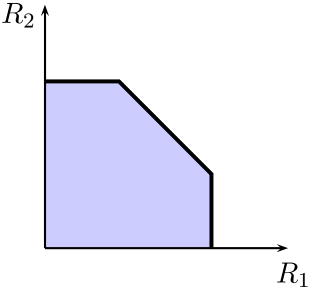
$$\begin{aligned} \tilde{\mathbf{y}}_2^{\mathsf{SI}} &= \Big[\mathbf{s}_2 - b_1\mathbf{v}_1\Big] \mod \Lambda \\ &= \Big[\big(a_{22} - b_1 a_{12}\big)\mathbf{t}_2 + \mathbf{z}_{\mathsf{effec}}(\mathbf{h}, \mathbf{a}_2)\Big] \mod \Lambda \ . \end{aligned}$$

- Now, the receiver can decode since $R_2 < R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a}_2)$.

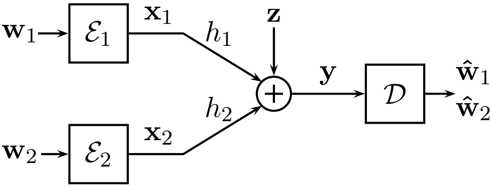
Multiple-Access via Computation

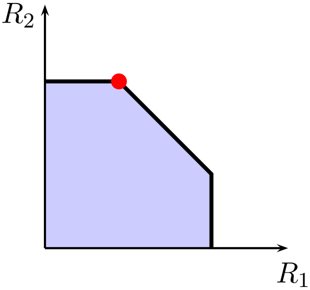




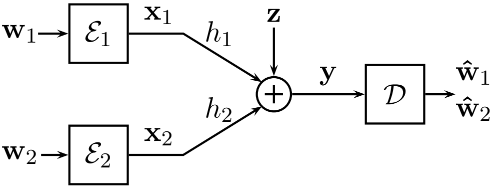


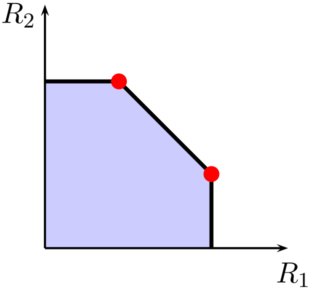
- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.





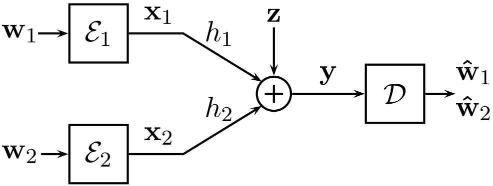
- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.

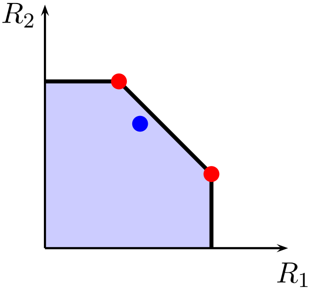




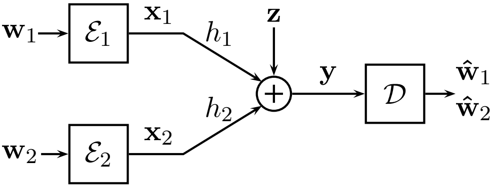
- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.

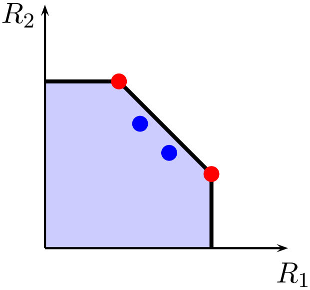
Multiple-Access via Computation



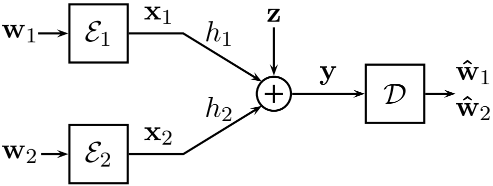


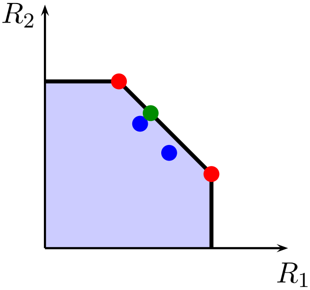
- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.
- Compute-and-forward achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.



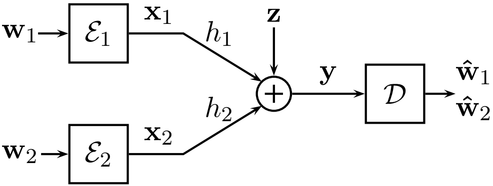


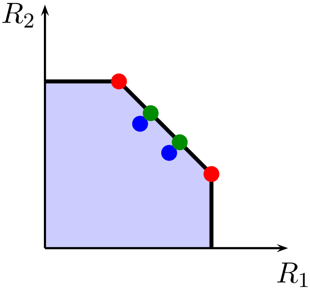
- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.
- Compute-and-forward achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.





- Successive 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.**



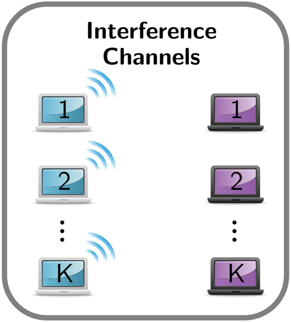


- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.
- Compute-and-forward achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.
- Successive compute-and-forward can attain the exact sum rate. Key Idea: Perform an MMSE 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.

Stream-by-stream case: Vasilis Ntranos, Viveck Cadambe, and Giuseppe Caire.

Interference-Free Capacity

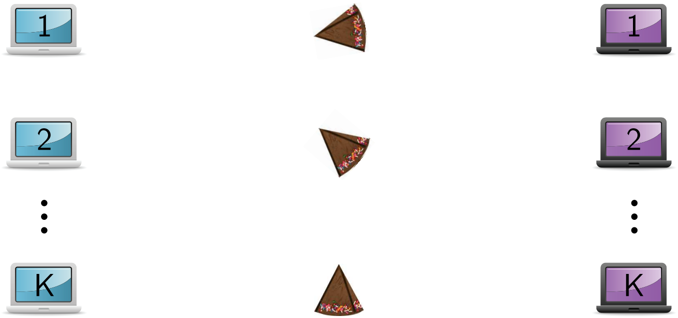
Interference-Free Capacity

Time Division

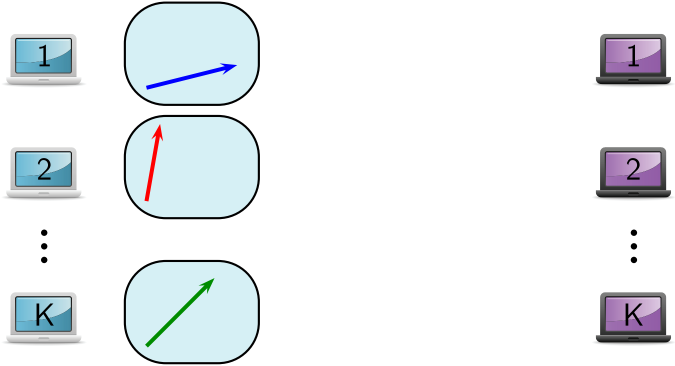
Time Division

Time Division

Time Division

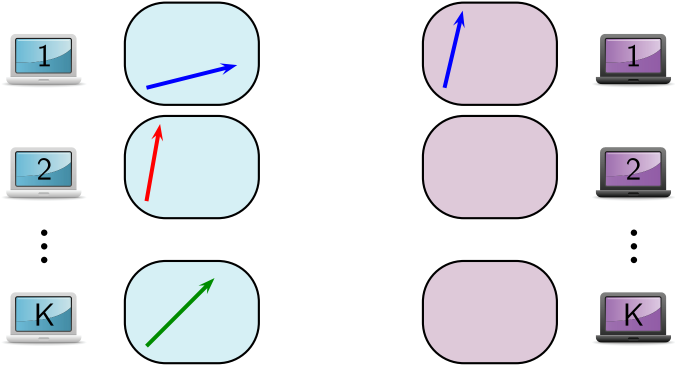


- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

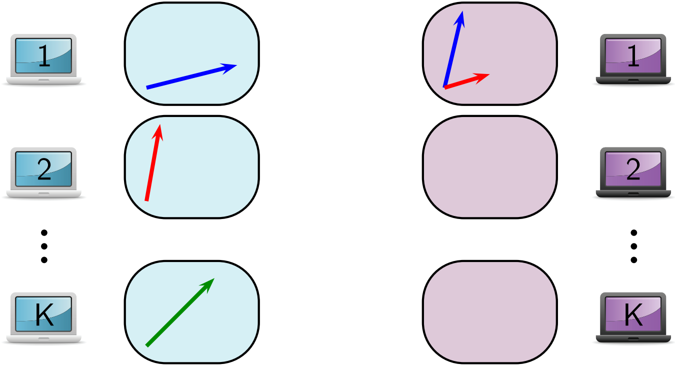


- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Interference Alignment

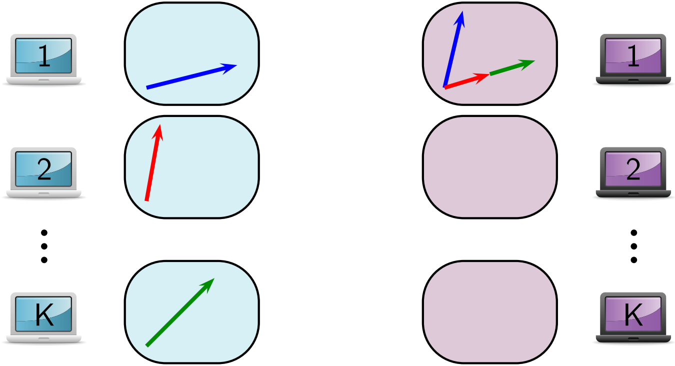


- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.



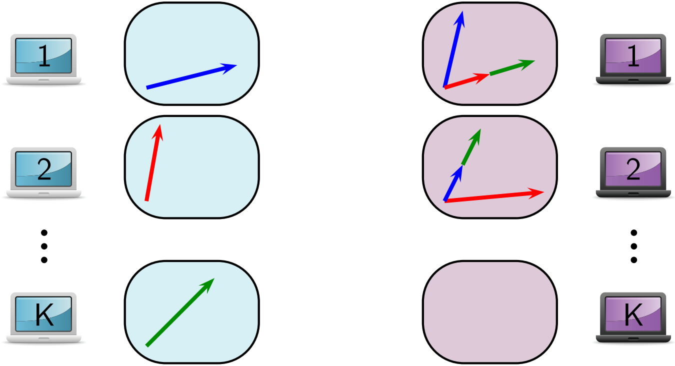
- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Interference Alignment



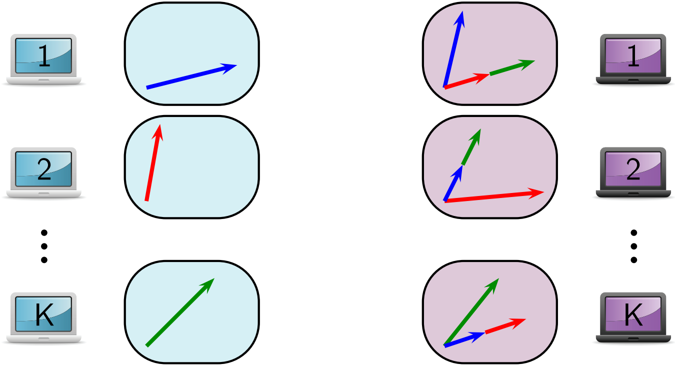
- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Interference Alignment



- **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.

Interference Alignment



- **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.

Interference Alignment

- **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 $\frac{K}{2}\log(1+2\mathsf{SNR})$ across roughly $(K\mathsf{SNR})^{K^2/2}$ 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 $\frac{K}{2}\log(1+2\mathsf{SNR})$ across roughly $(K\mathsf{SNR})^{K^2/2}$ 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.

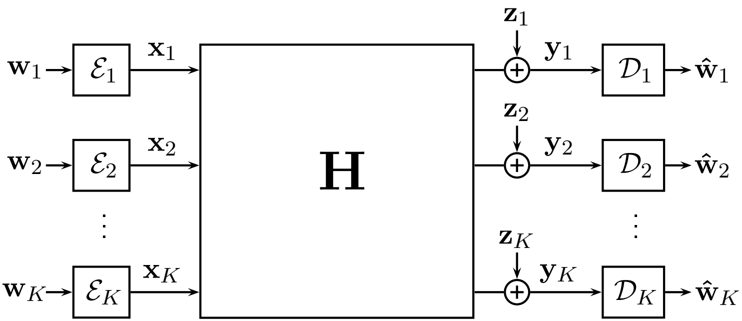
Ntranos-Cadambe-Nazer-Caire '13:

- 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.

Ntranos-Cadambe-Nazer-Caire '13:

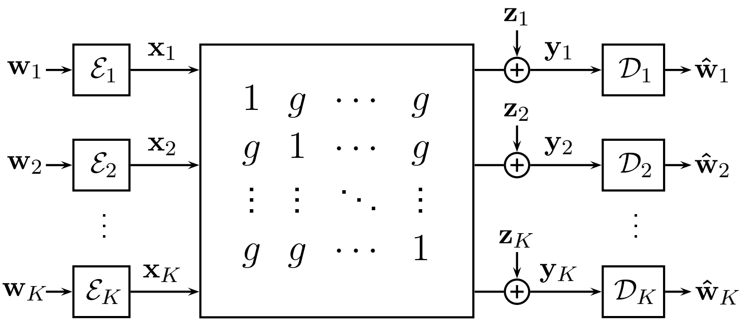
- Proposed a new framework, **integer-forcing interference alignment**, that can simultaneously exploit signal space and signal scale alignment.
- Aimed at scenarios with finite channel diversity (e.g., a few independent fading realizations) and finite SNR.
- Yields an achievable rate region for any scenario which employs "stream-by-stream" alignment.
- 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



- 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**.

Symmetric K-User Gaussian Interference Channel



- 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,

$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

Effective Multiple-Access Channel

- Lattice codes can enable signal scale alignment.

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

$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

- **Successive cancellation:** Decode and subtract interference $\sum_{\ell \neq k} \mathbf{x}_\ell$ before going after desired message.

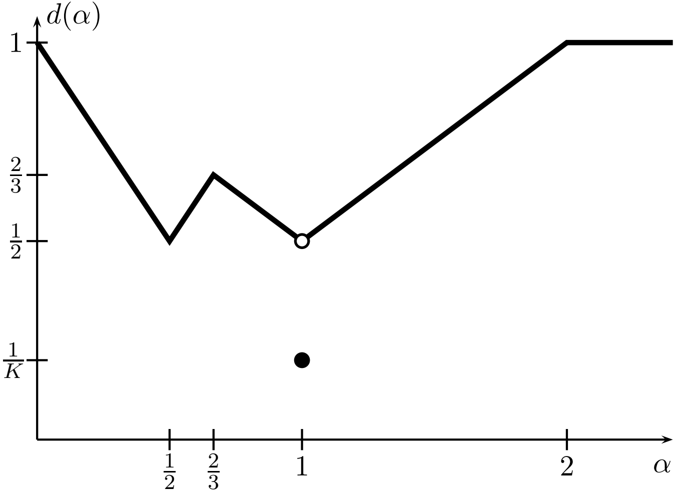
- Only optimal when the interference is very strong, **Sridharan et al. '08**.

Effective Multiple-Access Channel

- Lattice codes can enable signal scale alignment.
- Each receiver sees an effective two-user multiple-access channel,

$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

- **Successive cancellation:** Decode and subtract interference $\sum_{\ell \neq k} \mathbf{x}_\ell$ 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.**

$$\alpha = \frac{\log g^2 \mathsf{SNR}}{\log \mathsf{SNR}} \qquad\qquad d(\alpha) = \lim_{\mathsf{SNR}\to\infty} \frac{R(\mathsf{SNR})}{\frac{1}{2}\log \mathsf{SNR}}$$

Alignment via Two Equations

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

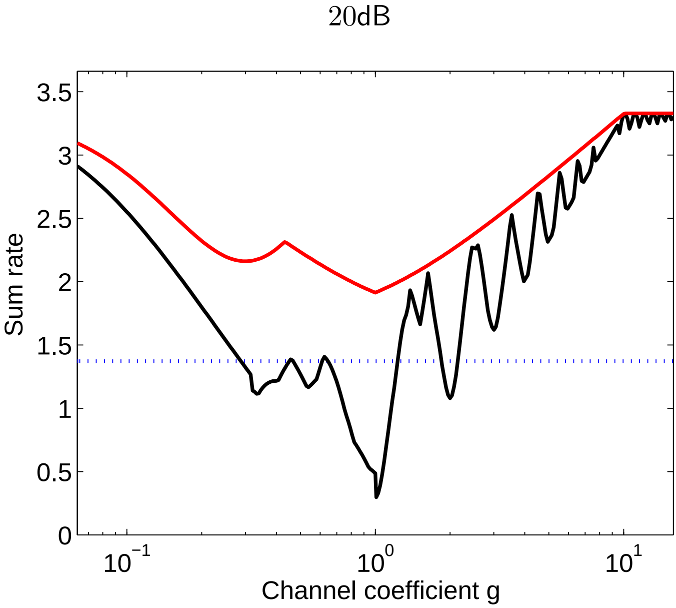
$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

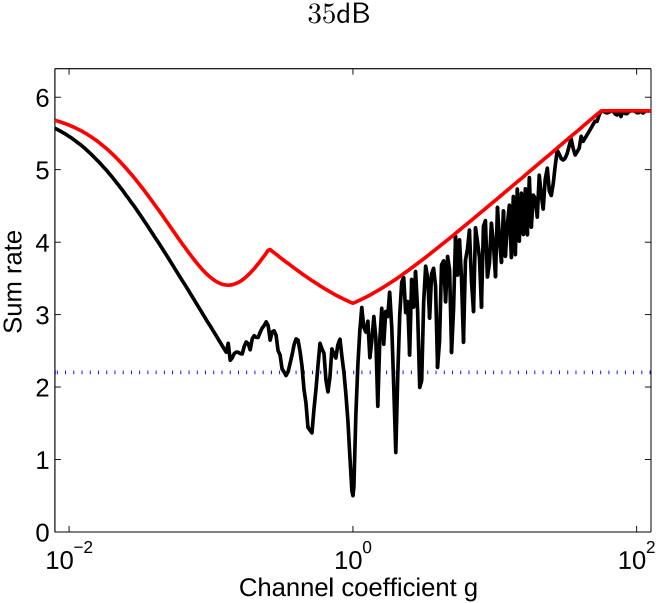
- **Ordentlich-Erez-Nazer '12:** Decode two linear combinations:

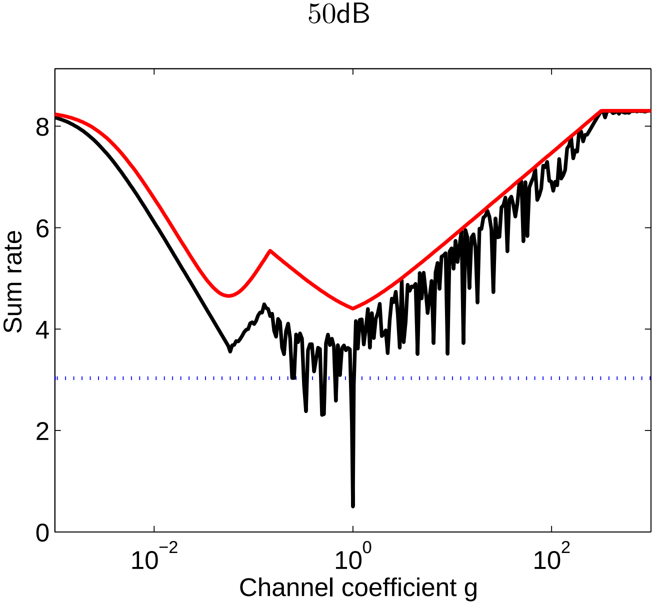
$$a_1 \mathbf{x}_k + a_2 \sum_{\ell \neq k} \mathbf{x}_\ell \qquad\qquad b_1 \mathbf{x}_k + b_2 \sum_{\ell \neq k} \mathbf{x}_\ell \ .$$

- 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.

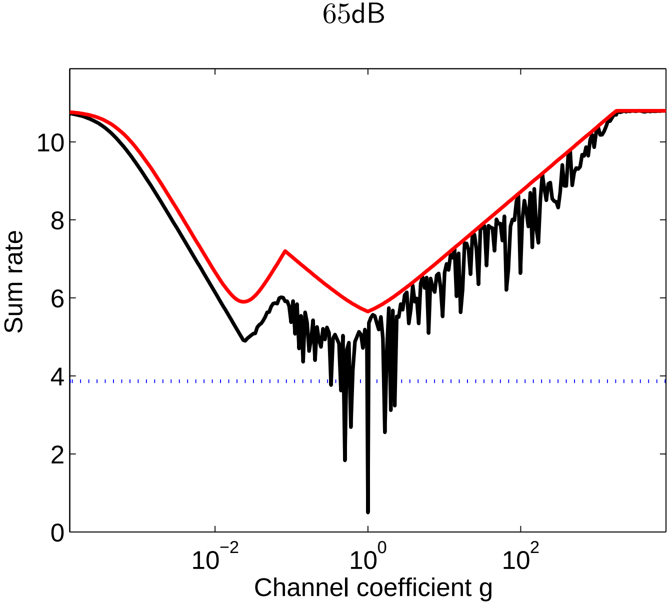
Symmetric K-User Gaussian Interference Channel




35dB
Sum rate
Channel coefficient g
0
1
2
3
4
5
6
10^{-2}
10^{0}
10^{2}



50dB
Sum rate
0
2
4
6
8
10^{-2}
10^{0}
10^{2}
Channel coefficient g



65dB
Sum rate
Channel coefficient g
0
2
4
6
8
10
10^{-2}
10^{0}
10^{2}

Beyond Symmetric Channels

- 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) $\mathbf{s}^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $\mathbf{v}^{[\ell]} \in \mathbb{C}^M$.

Problem Setting:

- Multiple data streams (i.e., codewords) $\mathbf{s}^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $\mathbf{v}^{[\ell]} \in \mathbb{C}^M$.
- Each receiver sees a noisy linear combination of its desired and interfering streams:

$$\mathbf{Y} = \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{D}}^{[\ell]} \mathbf{v}_{\mathrm{D}}^{[\ell]} \big(\mathbf{s}_{\mathrm{D}}^{[\ell]}\big)^{\mathsf{T}} + \sum_{j=1}^{J} \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} \big(\mathbf{s}_{\mathrm{I}}^{[j,\ell]}\big)^{\mathsf{T}} + \mathbf{Z} \ .$$

Problem Setting:

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

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

$$\mathbf{Y} = \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{D}}^{[\ell]} \mathbf{v}_{\mathrm{D}}^{[\ell]} \big(\mathbf{s}_{\mathrm{D}}^{[\ell]}\big)^{\mathsf{T}} + \sum_{j=1}^{J} \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} \big(\mathbf{s}_{\mathrm{I}}^{[j,\ell]}\big)^{\mathsf{T}} + \mathbf{Z} \ .$$

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

$$\mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} = \mathbf{H}_{\mathrm{I}}^{[j,\tilde{\ell}]} \mathbf{v}_{\mathrm{I}}^{[j,\tilde{\ell}]} \qquad \forall \tilde{\ell} \neq \ell \ .$$

Problem Setting:

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

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

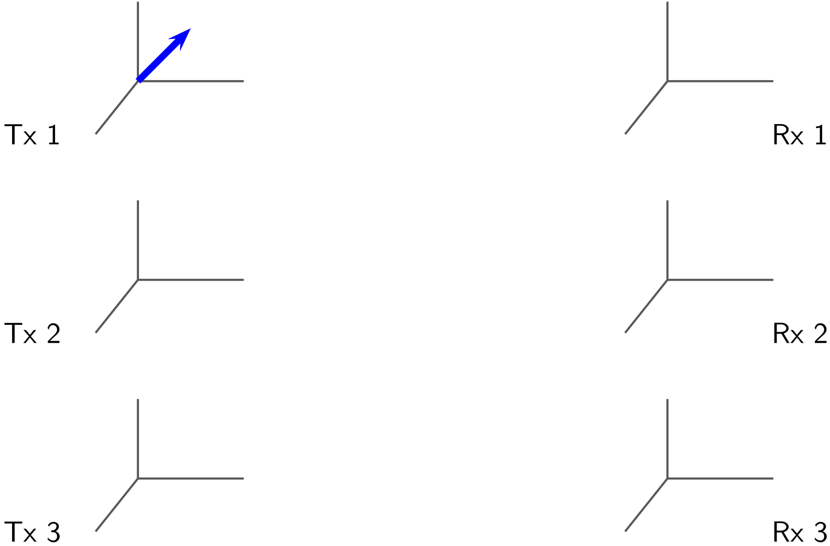
$$\mathbf{Y} = \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{D}}^{[\ell]} \mathbf{v}_{\mathrm{D}}^{[\ell]} \big(\mathbf{s}_{\mathrm{D}}^{[\ell]}\big)^{\mathsf{T}} + \sum_{j=1}^{J} \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} \big(\mathbf{s}_{\mathrm{I}}^{[j,\ell]}\big)^{\mathsf{T}} + \mathbf{Z} \ .$$

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

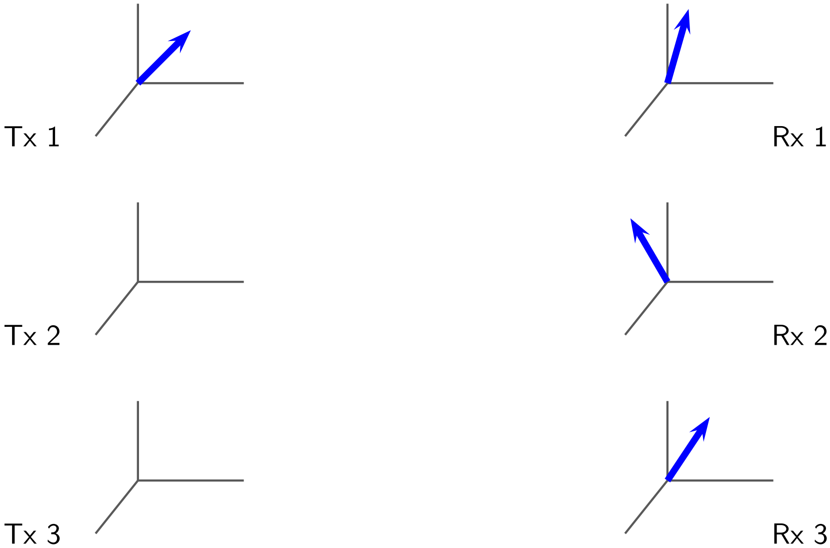
$$\mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} = \mathbf{H}_{\mathrm{I}}^{[j,\tilde{\ell}]} \mathbf{v}_{\mathrm{I}}^{[j,\tilde{\ell}]} \qquad \forall \tilde{\ell} \neq \ell \ .$$

- Different powers and rates allowed across data streams.

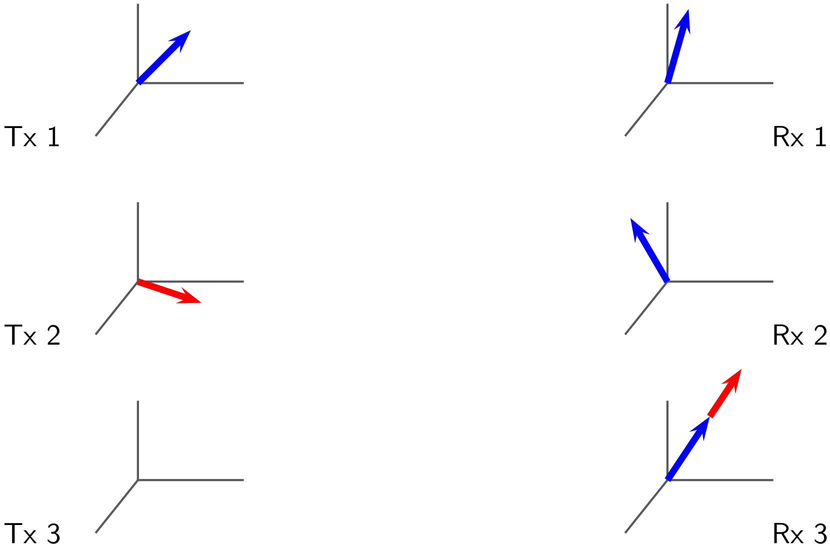
Example: Cadambe-Jafar '08 over 3 Channel Realizations



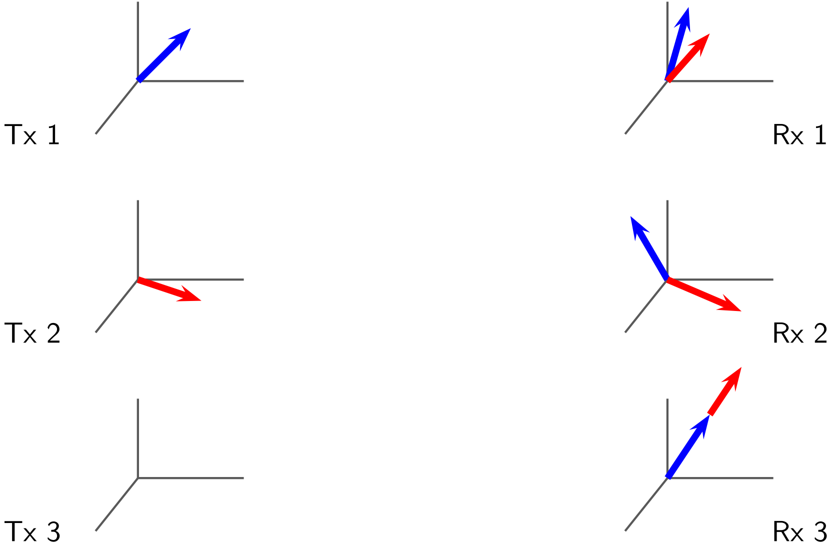
Example: Cadambe-Jafar '08 over 3 Channel Realizations



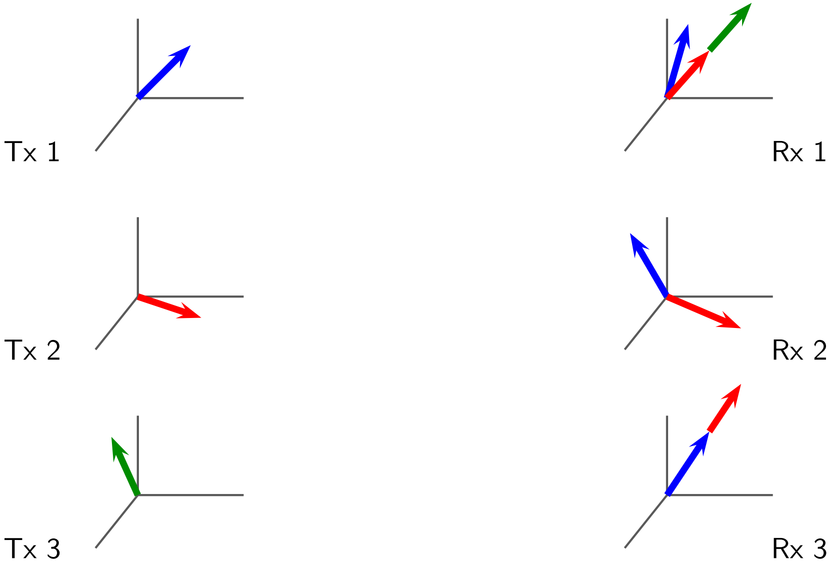
Example: Cadambe-Jafar '08 over 3 Channel Realizations



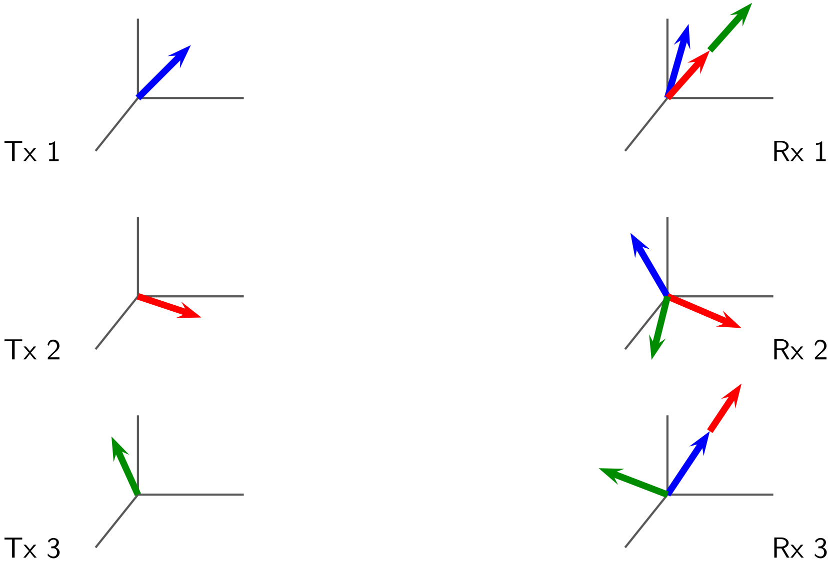
Example: Cadambe-Jafar '08 over 3 Channel Realizations



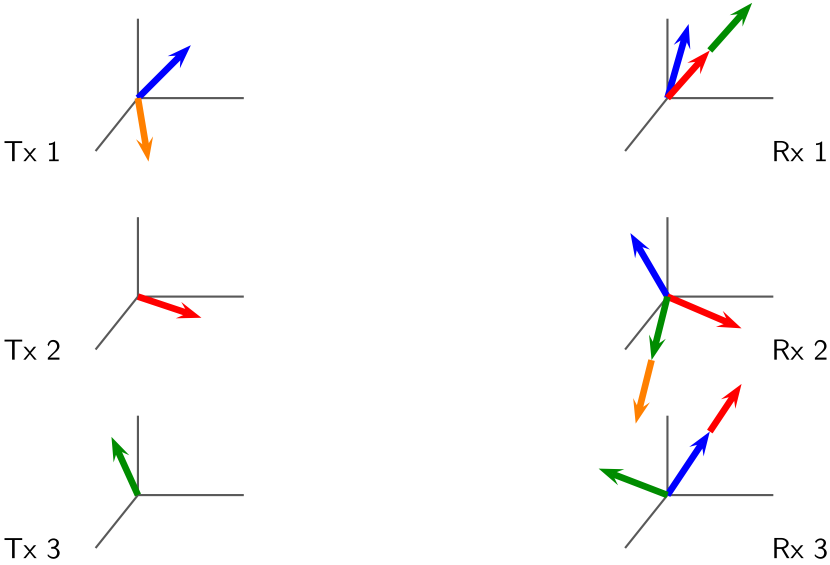
Example: Cadambe-Jafar '08 over 3 Channel Realizations



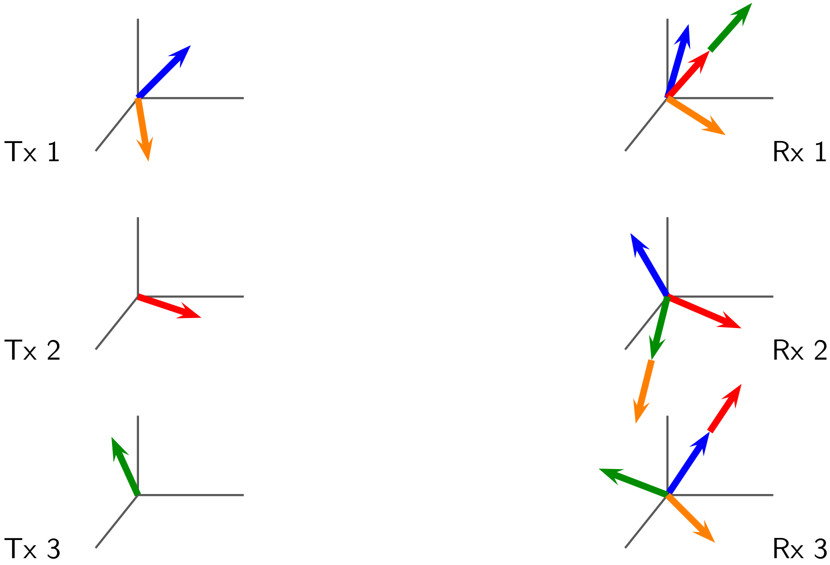
Example: Cadambe-Jafar '08 over 3 Channel Realizations



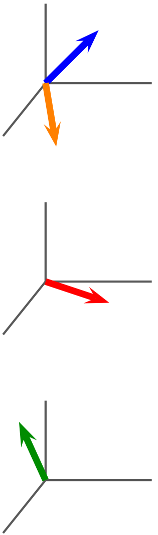
Example: Cadambe-Jafar '08 over 3 Channel Realizations



Example: Cadambe-Jafar '08 over 3 Channel Realizations



Example: Cadambe-Jafar '08 over 3 Channel Realizations



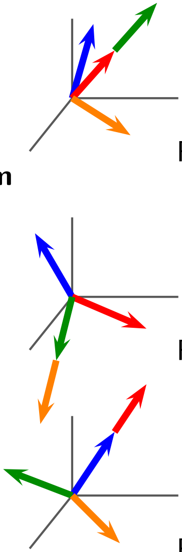
Tx 1

Tx 2

Tx 3

Total Degrees of Freedom

$$\text{DoF} = \frac{4 \text{ vectors}}{3 \text{ channel uses}}$$
$$= \frac{4}{3}$$

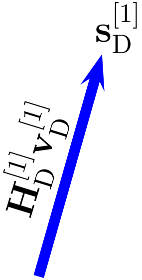


Rx 1

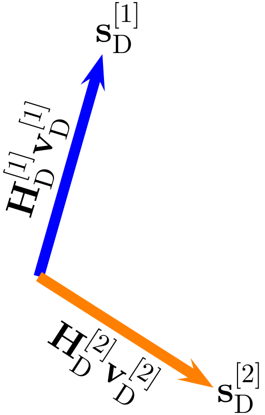
Rx 2

Rx 3

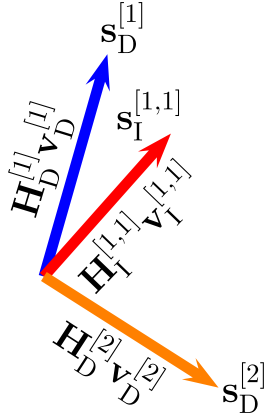
Received Signal



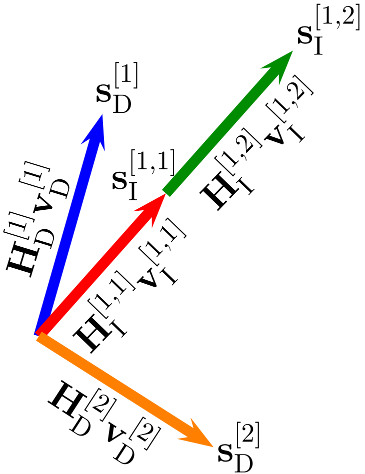
Received Signal



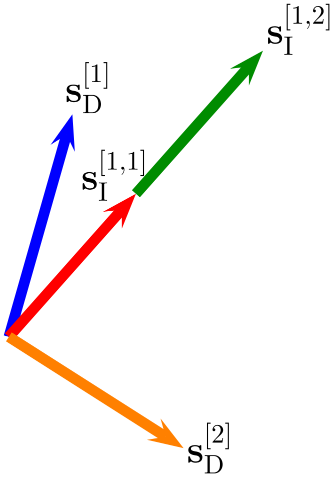
Received Signal



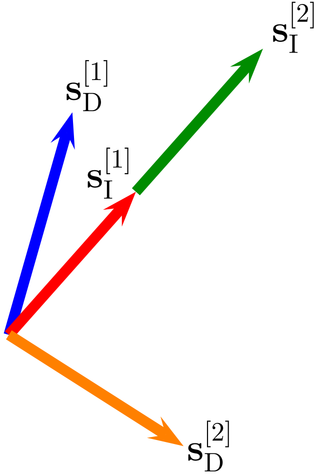
Received Signal

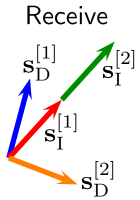


Received Signal



Received Signal



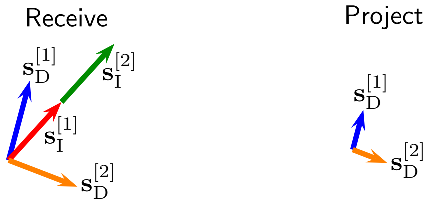


How should each receiver decoder its desired data streams?

Zero-Forcing Interference Alignment:

- Generate the data streams using i.i.d. random coding.

Zero-Forcing Decoding

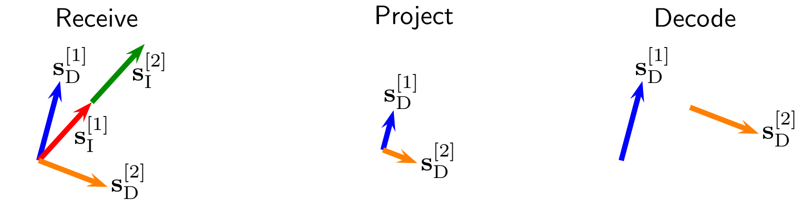


How should each receiver decoder its desired data streams?

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.

Zero-Forcing Decoding

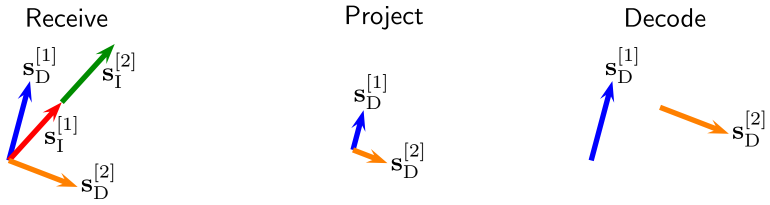


How should each receiver decoder its desired data streams?

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.

Zero-Forcing Decoding

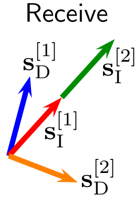


How should each receiver decoder its desired data streams?

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.
- Suffices from a degrees-of-freedom perspective.

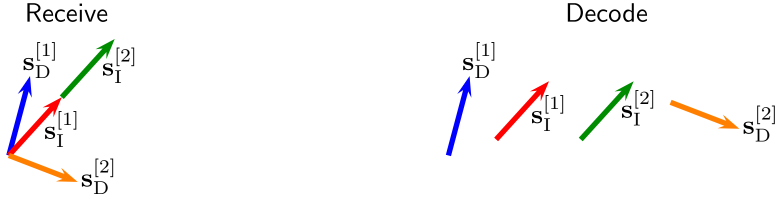
Joint Decoding (with i.i.d Random Codes)



How should each receiver decoder its desired data streams?

Joint Typicality Decoding:

- Generate the data streams using i.i.d. random coding.

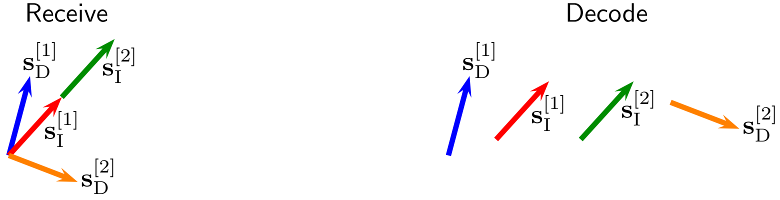


How should each receiver decoder its desired data streams?

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.

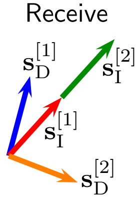
Joint Decoding (with i.i.d Random Codes)



How should each receiver decoder its desired data streams?

Joint Typicality Decoding:

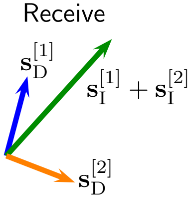
- Generate the data streams using i.i.d. random coding.
- If we attempt to decode the aligned interference, we will end up decoding each interferer separately.
- This significantly reduces the achievable rate per data stream.
- Analyzing lattice-coded data streams is beyond the reach of current techniques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



How should each receiver decoder its desired data streams?

Integer-Forcing Interference Alignment:

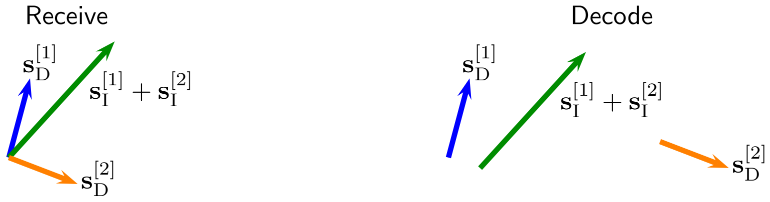
- Beamforming directions chosen to induce signal space alignment.



How should each receiver decoder its desired data streams?

Integer-Forcing Interference Alignment:

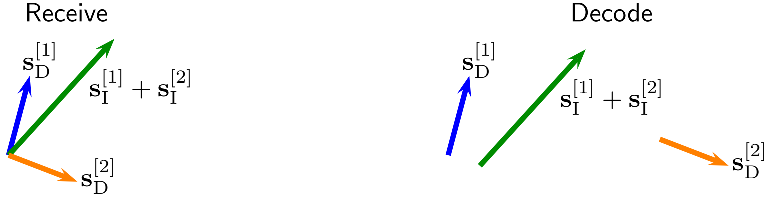
- Beamforming directions chosen to induce signal space alignment.
- Data streams are encoded using nested lattice codes according to some power allocation. This induces signal scale alignment.



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.



How should each receiver decoder its desired data streams?

Integer-Forcing Interference Alignment:

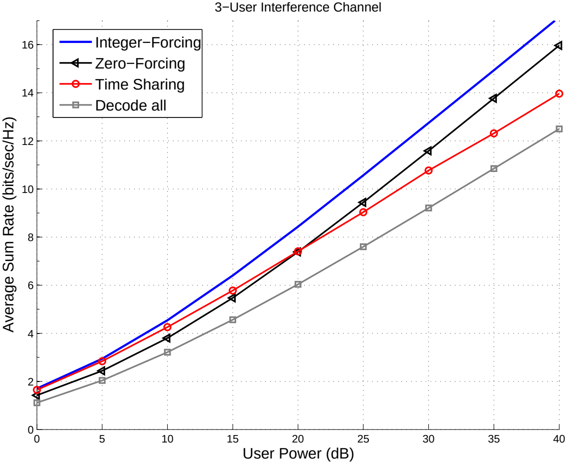
- Beamforming directions chosen to induce signal space alignment.
- Data streams are encoded using nested lattice codes according to some power allocation. This induces signal scale alignment.
- Receiver decodes linear combinations and solves for its desired data streams.
- Requires extension of compute-and-forward to asymmetric powers.

Performance Comparison

- 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.
- Broader story: Algebraic Structure in Network Information Theory. ISIT '11 Tutorial. Survey on physical-layer network coding in Proceedings of the IEEE, March 2011.