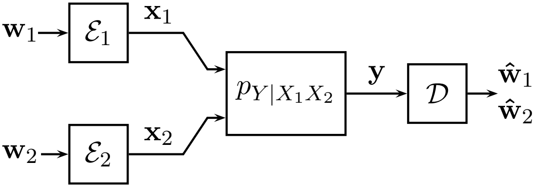
Interference Management: The Compute-and-Forward Perspective

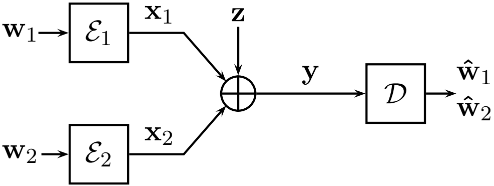
Michael Gastpar, EPFL and UC Berkeley

Acknowledgement: Bobak Nazer

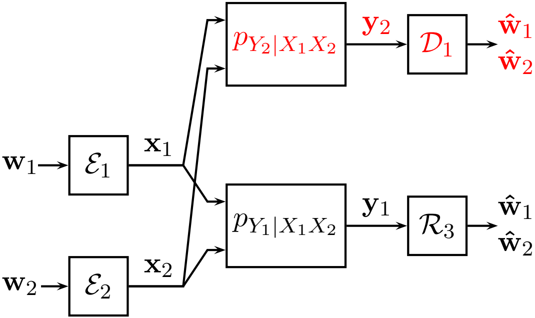
NEWCOM School on Interference Management, EURECOM
May 30, 2013


$\mathbf{w}_1$
$\mathcal{E}_1$
$\mathbf{x}_1$
$\mathbf{w}_2$
$\mathcal{E}_2$
$\mathbf{x}_2$
$p_{Y|X_1X_2}$
$\mathbf{y}$
$\mathcal{D}$
$\hat{\mathbf{w}}_1$
$\hat{\mathbf{w}}_2$

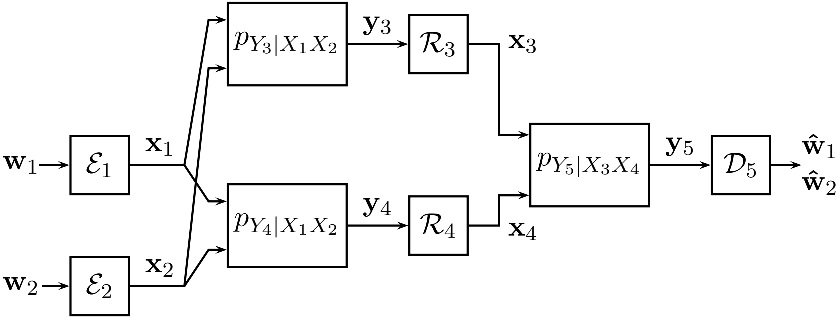
Interference



Interference



Interference



Interference

What should an *intermediate node* do with interfering signals?

- It could decode all of the transmitted signals.
- It could compress its observation and forward this description.
- ...

To discuss these questions, we need a more formal framework, which we will introduce next.

Point-to-Point Channels

$$\mathbf{w} \longrightarrow \boxed{\mathcal{E}} \xrightarrow{\mathbf{x}} \boxed{p_{Y|X}} \xrightarrow{\mathbf{y}} \boxed{\mathcal{D}} \longrightarrow \hat{\mathbf{w}}$$

The Usual Suspects:

- Message $\mathbf{w} \in \{0,1\}^k$
- Encoder $\mathcal{E} : \{0,1\}^k \to \mathcal{X}^n$
- Input $\mathbf{x} \in \mathcal{X}^n$
- Estimate $\hat{\mathbf{w}} \in \{0,1\}^k$
- Decoder $\mathcal{D} : \mathcal{Y}^n \to \{0,1\}^k$
- Output $\mathbf{y} \in \mathcal{Y}^n$

- Memoryless Channel $p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(y_i|x_i)$
- Rate $R = \dfrac{k}{n}$.
- (Average) Probability of Error: $\mathbb{P}\{\hat{\mathbf{w}} \neq \mathbf{w}\} \to 0$ as $n \to \infty$. Assume $\mathbf{w}$ is uniform over $\{0,1\}^k$.

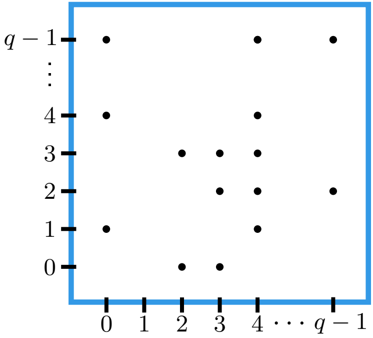
i.i.d. Random Codes

- Generate 2^{nR} codewords $\mathbf{x} = [X_1 \ X_2 \ \cdots \ X_n]$ independently and elementwise i.i.d. according to some distribution p_X

$$p(\mathbf{x}) = \prod_{i=1}^{n} p_X(x_i)$$

- Bound the average error probability for a random codebook.

- If the average performance over codebooks is good, there must exist at least one good fixed codebook.



(Weak) Joint Typicality

- Two sequences $\mathbf{x}$ and $\mathbf{y}$ are (weakly) jointly typical if

$$\left| -\frac{1}{n}\log p(\mathbf{x}) - H(X)\right| < \epsilon$$
$$\left| -\frac{1}{n}\log p(\mathbf{y}) - H(Y)\right| < \epsilon$$
$$\left| -\frac{1}{n}\log p(\mathbf{x},\mathbf{y}) - H(X,Y)\right| < \epsilon$$

- For our considerations, weak typicality is convenient as it can also be stated in terms of differential entropies.

- If $\mathbf{x}$ and $\mathbf{y}$ are i.i.d. sequences, the probability that they are jointly typical goes to 1 as n goes to infinity.

Decoder looks for a codeword that is jointly typical with the received sequence $\mathbf{y}$

Error Events

1. Transmitted codeword $\mathbf{x}$ is not jointly typical with $\mathbf{y}$.
   $\Longrightarrow$ Low probability by the Weak Law of Large Numbers.
2. Another codeword $\tilde{\mathbf{x}}$ is jointly typical with $\mathbf{y}$.

Cuckoo's Egg Lemma

Let $\tilde{\mathbf{x}}$ be an i.i.d. sequence that is independent from the received sequence $\mathbf{y}$.

$$\mathbb{P}\Big\{(\tilde{\mathbf{x}}, \mathbf{y}) \text{ is jointly typical}\Big\} \leq 2^{-n(I(X;Y)-3\epsilon)}$$

See **Cover and Thomas**.

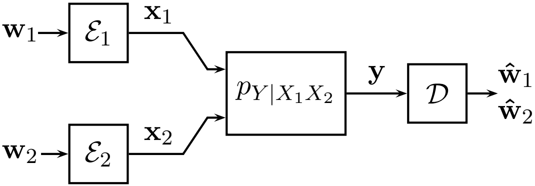
- We can upper bound the probability of error via the union bound:

$$\begin{aligned}\mathbb{P}\{\hat{\mathbf{w}} \neq \mathbf{w}\} &\leq \sum_{\tilde{\mathbf{w}} \neq \mathbf{w}} \mathbb{P}\Big\{(\mathbf{x}(\tilde{\mathbf{w}}), \mathbf{y}) \text{ is jointly typical.}\Big\} \\ &\leq 2^{-n(I(X;Y) - R - 3\epsilon)} \qquad \leftarrow \text{Cuckoo's Egg Lemma}\end{aligned}$$

- If $R < I(X;Y)$, then the probability of error can be driven to zero as the blocklength increases.

Theorem (Shannon '48)

The capacity of a point-to-point channel is $C = \max_{p_X} I(X;Y)$.



- **Rate Region:** Set of rates (R_1, R_2) such that the encoders can send $\mathbf{w}_1$ and $\mathbf{w}_2$ to the decoder with vanishing probability of error

$$\mathbb{P}\{(\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2) \neq (\mathbf{w}_1, \mathbf{w}_2)\} \to 0 \text{ as } m \to \infty$$

Rate Region (Ahlswede, Liao)

Convex closure of all (R_1, R_2) satisfying

$$\begin{aligned} R_1 &< I(X_1; Y | X_2) \\ R_2 &< I(X_2; Y | X_1) \\ R_1 + R_2 &< I(X_1, X_2; Y) \end{aligned}$$

for some $p(x_1)p(x_2)$.

I. Interference

II. Compute-and-Forward

(a) Basic Ideas
(b) AWGN Case: Introduction to Lattice Codes
(c) AWGN Case: Lattice Codes for Compute-and-Forward
(d) Beyond the AWGN Case: A few thoughts

III. Interference: The Compute-and-Forward Perspective

IV. Single-Hop Networks

V. Multi-hop Networks

Point-to-Point Channels

$$\mathbf{w} \to \boxed{\mathcal{E}} \xrightarrow{\mathbf{x}} \boxed{p_{Y|X}} \xrightarrow{\mathbf{y}} \boxed{\mathcal{D}} \to \hat{\mathbf{w}}$$

The Usual Suspects:

- Message $\mathbf{w} \in \{0,1\}^k$
- Encoder $\mathcal{E} : \{0,1\}^k \to \mathcal{X}^n$
- Input $\mathbf{x} \in \mathcal{X}^n$
- Estimate $\hat{\mathbf{w}} \in \{0,1\}^k$
- Decoder $\mathcal{D} : \mathcal{Y}^n \to \{0,1\}^k$
- Output $\mathbf{y} \in \mathcal{Y}^n$

- Memoryless Channel $p(\mathbf{y}|\mathbf{x}) = \displaystyle\prod_{i=1}^{n} p(y_i|x_i)$
- Rate $R = \dfrac{k}{n}$.
- (Average) Probability of Error: $\mathbb{P}\{\hat{\mathbf{w}} \neq \mathbf{w}\} \to 0$ as $n \to \infty$. Assume $\mathbf{w}$ is uniform over $\{0,1\}^k$.

- Linear Codebook: A linear map between messages and codewords (instead of a lookup table).

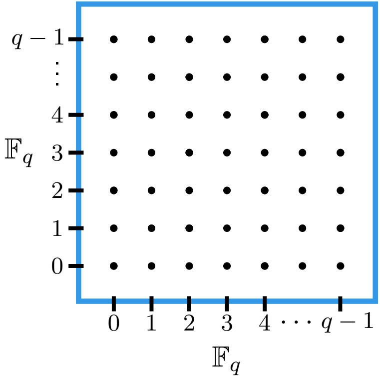
q-ary Linear Codes

- Represent message $\mathbf{w}$ as a length-k vector over $\mathbb{F}_q$.
- Codewords $\mathbf{x}$ are length-n vectors over $\mathbb{F}_q$.
- Encoding process is just a matrix multiplication, $\mathbf{x} = \mathbf{G}\mathbf{w}$.

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1k} \\ g_{21} & g_{22} & \cdots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nk} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}$$

- Recall that, for prime q, operations over $\mathbb{F}_q$ are just $\mod q$ operations over the reals.
- Rate $R = \frac{k}{n} \log q$

- Linear code looks like a regular subsampling of the elements of $\mathbb{F}_q^n$.
- **Random linear code:** Generate each element g_{ij} of the generator matrix $\mathbf{G}$ elementwise i.i.d. according to a uniform distribution over $\{0, 1, 2, \ldots, q-1\}$.
- How are the codewords distributed?



It is convenient to instead analyze the shifted ensemble $\bar{\mathbf{x}} = \mathbf{G}\mathbf{w} \oplus \mathbf{v}$ where $\mathbf{v}$ is an i.i.d. uniform sequence. (See **Gallager**.)

Shifted Codeword Properties

1. Marginally uniform over $\mathbb{F}_q^n$. For a given message $\mathbf{w}$, the codeword $\bar{\mathbf{x}}$ looks like an i.i.d. uniform sequence.

$$\mathbb{P}\{\bar{\mathbf{x}} = \mathsf{x}\} = \frac{1}{q^n} \quad \text{for all } \mathsf{x} \in \mathbb{F}_q^n$$

2. Pairwise independent. For $\mathbf{w}_1 \neq \mathbf{w}_2$, codewords $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2$ are independent.

$$\mathbb{P}\{\bar{\mathbf{x}}_\mathbf{1} = \mathsf{x}_1, \bar{\mathbf{x}}_\mathbf{2} = \mathsf{x}_2\} = \frac{1}{q^{2n}} = \mathbb{P}\{\bar{\mathbf{x}}_1 = \mathsf{x}_1\}\mathbb{P}\{\bar{\mathbf{x}}_2 = \mathsf{x}_2\}$$

Achievable Rates

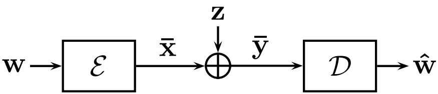
- Cuckoo's Egg Lemma only requires independence between the true codeword $\mathbf{x}(\mathbf{w})$ and the other codeword $\mathbf{x}(\tilde{\mathbf{w}})$. From the union bound:

$$\begin{aligned}\mathbb{P}\{\hat{\mathbf{w}} \neq \mathbf{w}\} &\leq \sum_{\tilde{\mathbf{w}} \neq \mathbf{w}} \mathbb{P}\Big\{(\mathbf{x}(\tilde{\mathbf{w}}), \mathbf{y}) \;\text{ is jointly typical.}\Big\} \\ &\leq 2^{-n(I(X;Y) - R - 3\epsilon)}\end{aligned}$$

- This is exactly what we get from pairwise independence.

- Thus, there exists a good fixed generator matrix $\mathbf{G}$ and shift $\mathbf{v}$ for any rate $R < I(X;Y)$ where X is uniform.

Removing the Shift



- For a binary symmetric channel (BSC), the output can be written as the modulo sum of the input plus i.i.d. Bernoulli(p) noise,

$$\bar{\mathbf{y}} = \bar{\mathbf{x}} \oplus \mathbf{z}$$
$$\bar{\mathbf{y}} = \mathbf{G}\mathbf{w} \oplus \mathbf{v} \oplus \mathbf{z}$$

- Due to this symmetry, the probability of error depends *only* on the realization of the noise vector $\mathbf{z}$.
  $\implies$ For a BSC, $\mathbf{x} = \mathbf{G}\mathbf{w}$ is a good code as well.

- We can now assume the existence of good generator matrices for channel coding.

Random I.I.D. vs. Random Linear

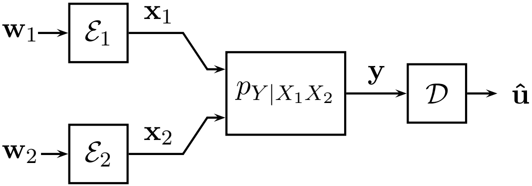
- What have we gotten for linearity (so far)?
  Simplified encoding. (Decoder is still quite complex.)

- What have we lost?
  Can only achieve $R = I(X;Y)$ for uniform X instead of $\max_{p_X} I(X;Y)$.

- In fact, this is a fundamental limitation of group codes, **Ahlswede '71**.

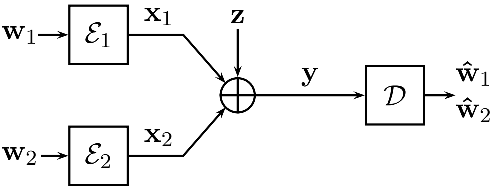
- Workarounds: symbol remapping **Gallager '68**, nested linear codes

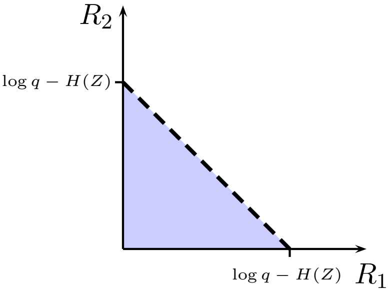
- Are random linear codes strictly worse than random i.i.d. codes?



- **Rate Region:** Set of rates (R_1, R_2) such that the decoder can recover $f(\mathbf{w}_1, \mathbf{w}_2)$ with vanishing probability of error

$$\mathbb{P}\{\hat{\mathbf{u}} \neq f(\mathbf{w}_1, \mathbf{w}_2)\} \to 0 \text{ as } m \to \infty$$





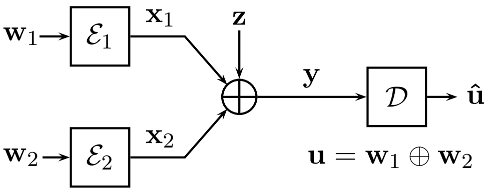
- Receiver observes noisy modulo sum of codewords $\mathbf{y} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \mathbf{z}$

Finite Field MAC Rate Region

All rates (R_1, R_2) satisfying

$$R_1 + R_2 \leq \log q - H(Z)$$

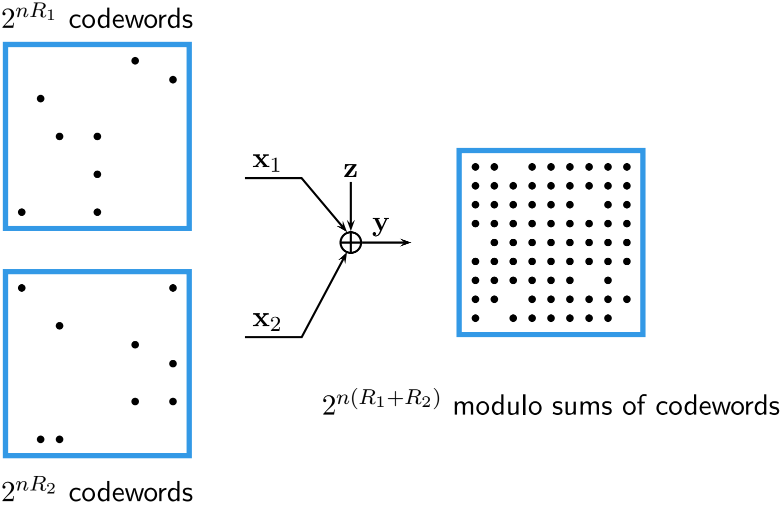
- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_q^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}\{\hat{\mathbf{u}} \neq \mathbf{u}\} \to 0$



I.I.D. Random Coding

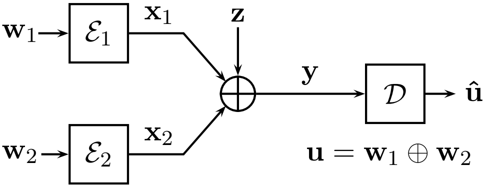
- Generate 2^{nR_1} i.i.d. uniform codewords for user 1.
- Generate 2^{nR_2} i.i.d. uniform codewords for user 2.
- With high probability, (nearly) all sums of codewords are distinct.
- This is ideal for multiple-access but not for computation.
- Need $R_1 + R_2 \leq \log q - H(Z)$

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



Independent msgs $\mathbf{w}_1, \mathbf{w}_2$.

Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}\{\hat{\mathbf{u}} \neq \mathbf{u}\} \to 0$



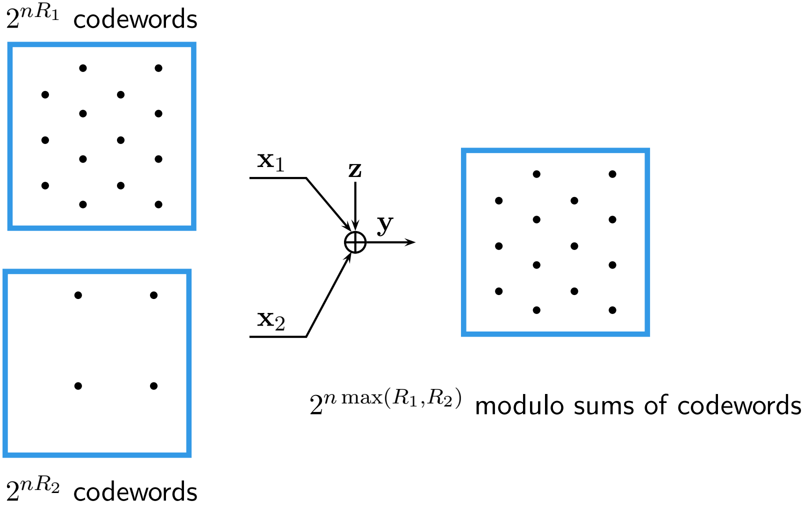
Random Linear Coding

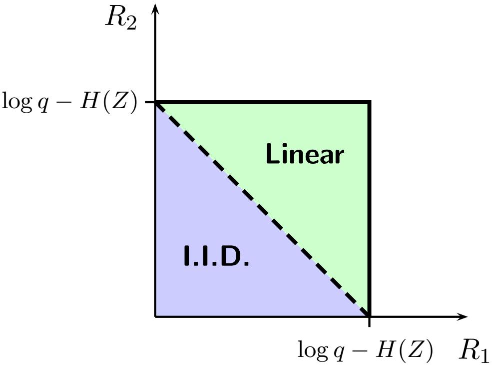
- Same linear code at both transmitters $\mathbf{x}_1 = \mathbf{G}\mathbf{w}_1$, $\mathbf{x}_2 = \mathbf{G}\mathbf{w}_2$.
- Sums of codewords are themselves codewords:

$$\begin{aligned}\mathbf{y} &= \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \mathbf{z} \\ &= \mathbf{G}\mathbf{w}_1 \oplus \mathbf{G}\mathbf{w}_2 \oplus \mathbf{z} \\ &= \mathbf{G}(\mathbf{w}_1 \oplus \mathbf{w}_2) \oplus \mathbf{z} \\ &= \mathbf{G}\mathbf{u} \oplus \mathbf{z}\end{aligned}$$

- Need $\max(R_1, R_2) \leq \log q - H(Z)$

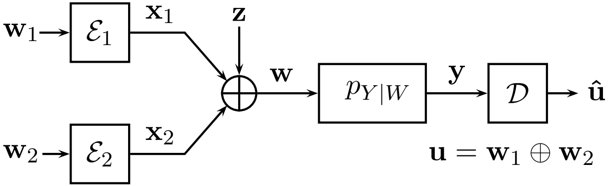
Random linear codes are good for computation



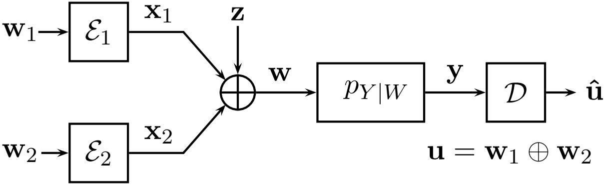


- **I.I.D. Random Coding**: $R_1 + R_2 \leq \log q - H(Z)$
- **Random Linear Coding**: $\max(R_1, R_2) \leq \log q - H(Z)$
- Linear codes double the sum rate *without any dependency*.
- Is this useful for *sending messages* (no computation)?

- Consider the following model:



- Find an achievable computation rate.

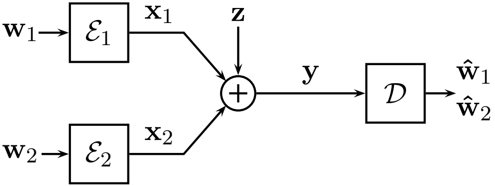


- One possibly interesting rate is attained by using the same binary linear code at both transmitters.
- Then, the resulting computation rate is simply $R = I(W; Y)$, where W is Bernoulli(1/2).

Gaussian Multiple-Access Channel: Capacity

Rate Region

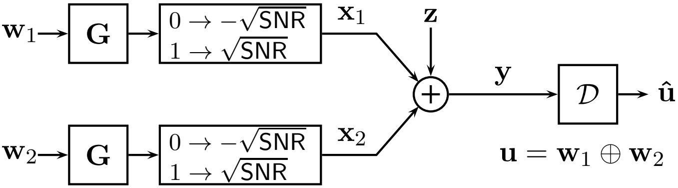
$$R_1 < \frac{1}{2}\log\left(1 + \frac{P_1}{N}\right)$$
$$R_2 < \frac{1}{2}\log\left(1 + \frac{P_2}{N}\right)$$
$$R_1 + R_2 < \frac{1}{2}\log\left(1 + \frac{P_1 + P_2}{N}\right)$$



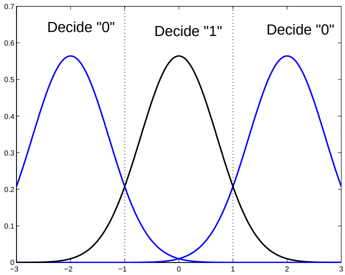
Power constraints P_1, P_2. Noise variance N.

How can we extend this to the Gaussian Multiple-Access Channel?

- Let us assume $P = P_1 = P_2$, and introduce, for simplicity, $\mathsf{SNR} = P/N$.
- Then, consider the following simple approach for the computation problem:



Let us use a simple sub-optimal decoding step:



- Step 1: for each symbol, decide between $-2, 0, 2$.
- Step 2: Map: 0 to 1, and both -2 and 2 to 0. Overall, this leads to a binary asymmetric channel.
- Step 3: ML decoding with respect to the code.

Exercise: Calculate the rate at which we can decode the modulo-2 sum.

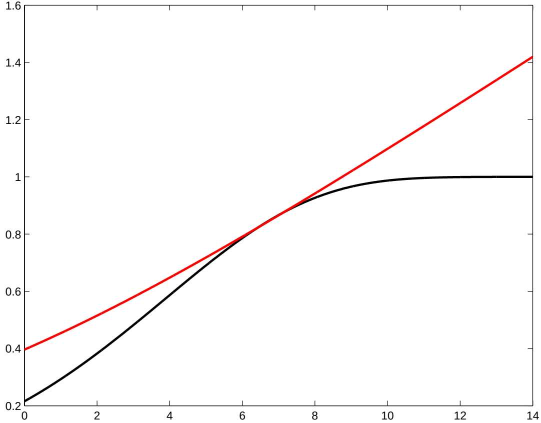
Exercise: Calculate the rate at which we can decode the modulo-2 sum.

Solution:

- Since both users use the *same code*, the overall scenario can be thought of as a *point-to-point channel* with a binary input.
- 0 is flipped to 1 with probability $Q(\sqrt{\mathsf{SNR}}) - Q(3\sqrt{\mathsf{SNR}})$.
- 1 is flipped to 0 with probability $2Q(\sqrt{\mathsf{SNR}})$.
- On this channel, we are using a *uniform input distribution.*
- Hence, the rate is equal to the mutual information across this channel, evaluated for uniform inputs.

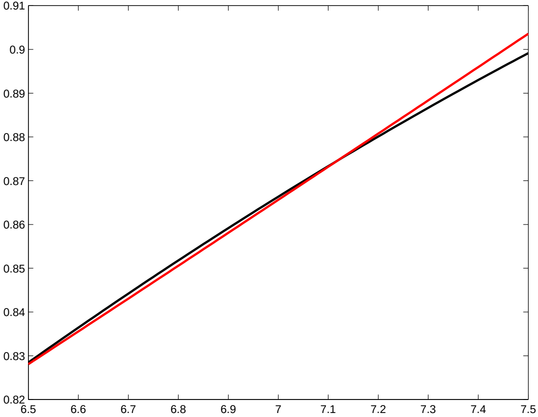
Gaussian Multiple-Access Channels: Mod-2 Sum Computation?

Two users:



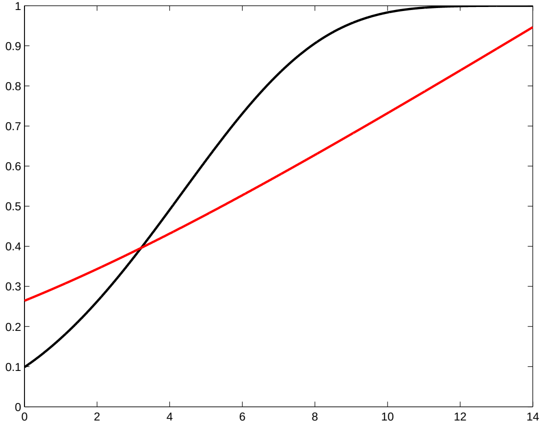
Gaussian Multiple-Access Channels: Mod-2 Sum Computation?

Two users, detail:



Gaussian Multiple-Access Channels: Mod-2 Sum Computation?

Three users:



The Role of Alphabet ("field") Size

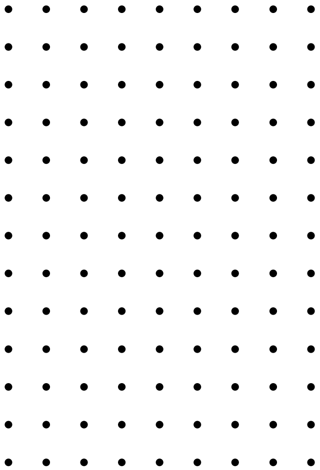
- In binary, the rate can be at most one.
- This may be acceptable in low-SNR.
- In high SNR, it appears inevitable to consider larger alphabets...

Lattices

- A lattice Λ is a discrete subgroup of $\mathbb{R}^n$.
- Can write a lattice as a linear transformation of the integer vectors,
$$\Lambda = \mathbf{B}\mathbb{Z}^n \ ,$$
for some $\mathbf{B} \in \mathbb{R}^{n \times n}$.

Lattice Properties

- Closed under addition: $\lambda_1, \lambda_2 \in \Lambda \implies \lambda_1 + \lambda_2 \in \Lambda$.
- Symmetric: $\lambda \in \Lambda \implies -\lambda \in \Lambda$



$\mathbb{Z}^n$ is a simple lattice.

Voronoi Regions

- Nearest neighbor quantizer:

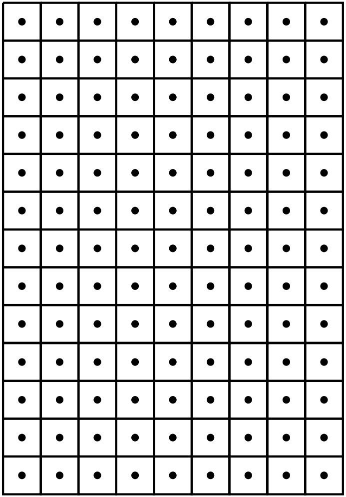
$$Q_\Lambda(\mathbf{x}) = \arg\min_{\lambda \in \Lambda} \|\mathbf{x} - \lambda\|_2$$

- The Voronoi region of a lattice point is the set of all points that quantize to that lattice point.

- Fundamental Voronoi region $\mathcal{V}$: points that quantize to the origin,

$$\mathcal{V} = \{\mathbf{x} : Q_\Lambda(\mathbf{x}) = \mathbf{0}\}$$

- Each Voronoi region is just a shift of the fundamental Voronoi region $\mathcal{V}$

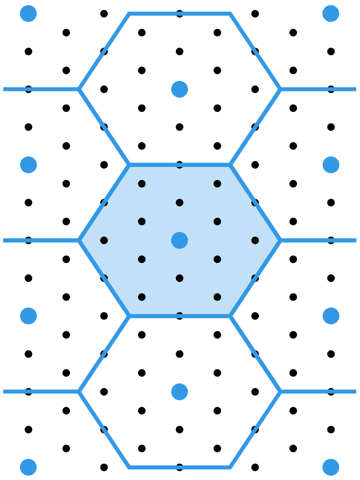


Lattice Tricks

1. Observing the power constraint: *Nested Lattices*
2. Proving achievable rates:
   - *Dithering*
   - *“MMSE Scaling”*

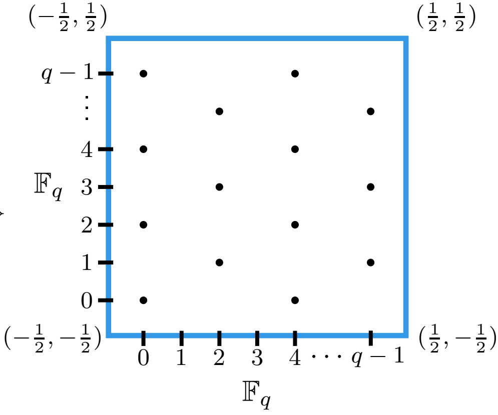
- Two lattices Λ and Λ_{FINE} are nested if $\Lambda \subset \Lambda_{\mathsf{FINE}}$
- **Nested Lattice Code:** All lattice points from Λ_{FINE} that fall in the fundamental Voronoi region $\mathcal{V}$ of Λ.
- $\mathcal{V}$ acts like a power constraint

$$\text{Rate} = \frac{1}{n} \log \left(\frac{\text{Vol}(\mathcal{V})}{\text{Vol}(\mathcal{V}_{\mathsf{FINE}})} \right)$$



Nested Lattice Codes from q-ary Linear Codes

- Choose an $n \times k$ generator matrix $\mathbf{G} \in \mathbb{F}_q^{n \times k}$ for q-ary code.
- Integers serve as coarse lattice, $\Lambda = \mathbb{Z}^n$.
- Map elements $\{0, 1, 2, \ldots, q-1\}$ to equally spaced points between $-1/2$ and $1/2$.
- Place codewords $\mathbf{x} = \mathbf{G}\mathbf{w}$ into the fundamental Voronoi region $\mathcal{V} = [-1/2, 1/2)^n$



Modulo Operation

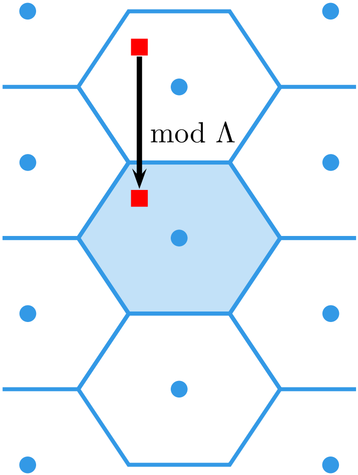
- Modulo operation with respect to lattice Λ is just the residual quantization error,

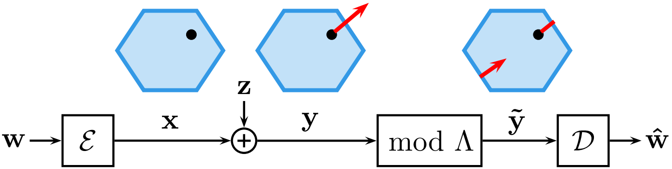
$$[\mathbf{x}] \bmod \Lambda = \mathbf{x} - Q_\Lambda(\mathbf{x}) \ .$$

- Mimics the role of $\bmod q$ in q-ary alphabet.

- Distributive Law:

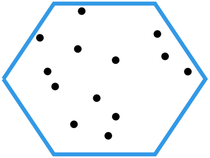
$$\Big[\mathbf{x}_1 + [\mathbf{x}_2] \bmod \Lambda\Big] \bmod \Lambda$$
$$= [\mathbf{x}_1 + \mathbf{x}_2] \bmod \Lambda$$



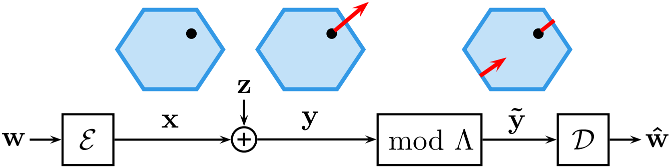


- Codebook lives on Voronoi region $\mathcal{V}$ of coarse lattice Λ.
- Take $\mod \Lambda$ of received signal prior to decoding.
- What is the capacity of the $\mod \Lambda$ channel?

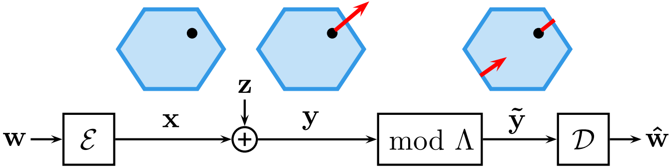
Using random i.i.d. code drawn over $\mathcal{V}$: $$C = \frac{1}{n} \max_{p(\mathbf{x})} I(\mathbf{x}; \tilde{\mathbf{y}})$$



$\bmod \Lambda$ *AWGN Channel Capacity*



$$
\begin{aligned}
nC &= \max_{p(\mathbf{x})} I(\mathbf{x};\tilde{\mathbf{y}}) \\
&= \max_{p(\mathbf{x})} \Big(h(\tilde{\mathbf{y}}) - h(\tilde{\mathbf{y}}|\mathbf{x}) \Big) \\
&= \max_{p(\mathbf{x})} \Big(h(\tilde{\mathbf{y}}) - h\Big([\mathbf{z}] \bmod \Lambda \Big) \Big) && \text{Distributive Law} \\
&\geq \max_{p(\mathbf{x})} \Big(h(\tilde{\mathbf{y}}) - h(\mathbf{z}) \Big) && \text{Point Symmetry of Voronoi Region} \\
&= \max_{p(\mathbf{x})} \Big(h(\tilde{\mathbf{y}}) - \frac{n}{2} \log(2\pi e N) \Big) && \text{Entropy of Gaussian Noise}
\end{aligned}
$$



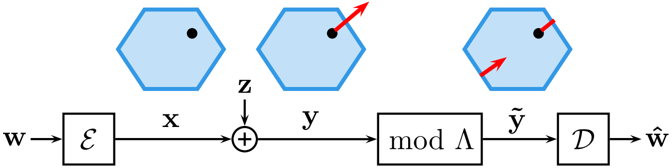
- Channel output entropy is equal to the logarithm of the Voronoi region volume if it is uniform over $\mathcal{V}$:

$$h(\tilde{\mathbf{y}}) = \log(\mathsf{Vol}(\mathcal{V})) \qquad \text{if } \tilde{\mathbf{y}} \sim \mathsf{Unif}(\mathcal{V})$$

- $\tilde{\mathbf{y}} = [\mathbf{x} + \mathbf{z}] \bmod \Lambda$ is uniform over $\mathcal{V}$ if $\mathbf{x}$ is uniform over $\mathcal{V}$.

- Random i.i.d. coding over the Voronoi region $\mathcal{V}$ can achieve:

$$R = \frac{1}{n}\log(\mathsf{Vol}(\mathcal{V})) - \frac{1}{2}\log(2\pi e N)$$

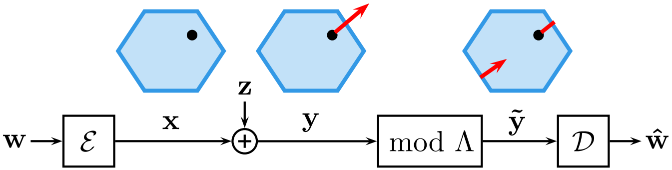


- Must scale lattice Λ so that the uniform distribution over the Voronoi region $\mathcal{V}$ meets the power constraint P.
- Set second moment $\sigma_\Lambda^2 = \frac{1}{n\mathsf{Vol}(\mathcal{V})} \int_{\mathcal{V}} \|\mathbf{x}\|^2 d\mathbf{x}$ equal to P.

Normalized Second Moment: $$G(\Lambda) = \frac{\sigma_\Lambda^2}{(\mathsf{Vol}(\mathcal{V}))^{2/n}}$$

$$\Longrightarrow \frac{1}{n}\log(\mathsf{Vol}(\mathcal{V})) = \frac{1}{2}\log\left(\frac{\sigma_\Lambda^2}{G(\Lambda)}\right) = \frac{1}{2}\log\left(\frac{P}{G(\Lambda)}\right)$$

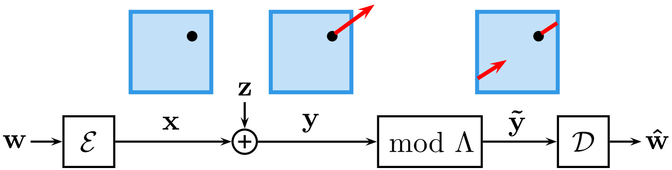
$\bmod \Lambda$ *AWGN Channel Capacity*



- Random i.i.d. coding over the Voronoi region $\mathcal{V}$ can achieve:

$$\begin{aligned} C &\geq \frac{1}{n}\log(\mathsf{Vol}(\mathcal{V})) - \frac{1}{2}\log(2\pi e N) \\ &= \frac{1}{2}\log\left(\frac{P}{G(\Lambda)}\right) - \frac{1}{2}\log(2\pi e N) \\ &= \frac{1}{2}\log\left(\frac{P}{N}\right) - \frac{1}{2}\log(2\pi e G(\Lambda)) \end{aligned}$$

What is $G(\Lambda)$?



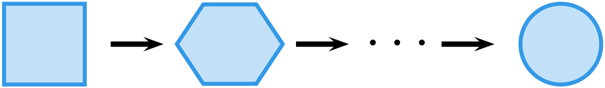
- The normalized second moment $G(\Lambda)$ is a dimensionless quantity that captures the shaping gain.
- Integer lattice is not so bad, $G(\mathbb{Z}^n) = 1/12$.
- Capacity under $\mod \mathbb{Z}^n$ is at least

$$C \geq \frac{1}{2}\log\left(\frac{P}{N}\right) - \frac{1}{2}\log\left(\frac{2\pi e}{12}\right)$$
$$\approx \frac{1}{2}\log\left(\frac{P}{N}\right) - 0.255$$

Asymptotically Good $G(\Lambda)$

Theorem (Zamir-Feder-Poltyrev '94)

There exists a sequence of lattices $\Lambda^{(n)}$ *such that* $\lim_{n\to\infty} G(\Lambda^{(n)}) = \frac{1}{2\pi e}$.



$n = 1$ $\quad$ $n = 2$ $\quad$ $n \to \infty$

- Best possible normalized second moment is that of a sphere.
- Using a sequence $\Lambda^{(n)}$ with an asymptotically good $G(\Lambda^{(N)})$ allows to approach

$$
\begin{aligned}
R &= \frac{1}{2}\log\left(\frac{P}{N}\right) - \frac{1}{2}\log\left(\frac{2\pi e}{2\pi e}\right) \\
&= \frac{1}{2}\log\left(\frac{P}{N}\right)
\end{aligned}
$$

Asymptotically Good $G(\Lambda)$

- Can actually get this with a linear code tiled over $\mathbb{Z}^n$ (see, for instance, **Erez-Litsyn-Zamir '05**.)
- Many works looking at this from different perspectives.
- We will just assume existence.

Recall the two key properties of random linear codes $\mathbf{G}$ from earlier:

Codeword Properties

1. Marginally uniform over $\mathbb{F}_q^n$. For a given message $\mathbf{w} \neq \mathbf{0}$, the codeword $\mathbf{x} = \mathbf{G}\mathbf{w}$ looks like an i.i.d. uniform sequence.

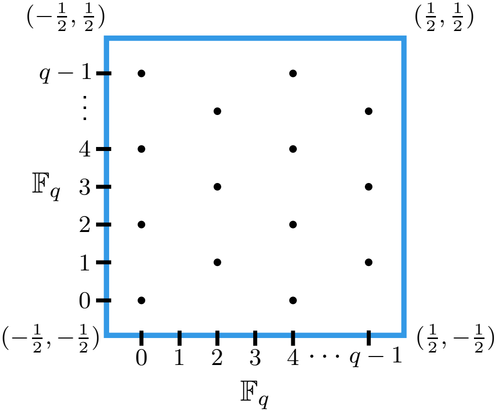
$$\mathbb{P}\{\mathbf{x} = \mathsf{x}\} = \frac{1}{q^n} \quad \text{for all } \mathsf{x} \in \mathbb{F}_q^n$$

2. Pairwise independent. For $\mathbf{w}_1, \mathbf{w}_2 \neq \mathbf{0}$, $\mathbf{w}_1 \neq \mathbf{w}_2$, codewords $\mathbf{x}_1, \mathbf{x}_2$ are independent.

$$\mathbb{P}\{\mathbf{x_1} = \mathsf{x}_1, \mathbf{x_2} = \mathsf{x}_2\} = \frac{1}{q^{2n}} = \mathbb{P}\{\mathbf{x}_1 = \mathsf{x}_1\}\mathbb{P}\{\mathbf{x}_2 = \mathsf{x}_2\}$$

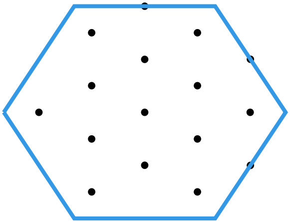
Linear Codes for $\bmod \Lambda$ *Channels*

- Instead of an "inner" random codes, we can use a q-ary linear code.
- This is exactly a nested lattice.
- Each codeword has a uniform marginal distribution over the grid.
- Rate loss due to finite constellation which goes to 0 as $q \to \infty$.
- Codewords are pairwise independent so we can apply the union bound.



$$\mathbf{x} = [\gamma \mathbf{G} \mathbf{w}] \mod \mathbb{Z}^n$$

- General coarse lattice $\Lambda = \mathbf{B}\mathbb{Z}^n$.
- First, apply generator matrix for linear code $\mathbf{Gw}$. Then scale down by γ and tile over $\mathbb{Z}^n$.
- Multiply by $\mathbf{B}$ and apply $\mod \Lambda$ to get codebook.
- As q gets large, each codeword's marginal distribution looks uniform over $\mathcal{V}$.
- Codewords are pairwise independent so we can apply the union bound.



$$\mathbf{x} = [\mathbf{B}\gamma\mathbf{G}\mathbf{w}] \mod \Lambda$$

- **Erez-Zamir '04:** Prior to taking $\bmod \Lambda$, scale by α.

$$\begin{aligned}\tilde{\mathbf{y}} &= [\alpha \mathbf{y}] \bmod \Lambda \\ &= [\alpha \mathbf{x} + \alpha \mathbf{z}] \bmod \Lambda \\ &= [\mathbf{x} + \underbrace{\alpha \mathbf{z} - (1-\alpha)\mathbf{x}}_{\text{Effective Noise}}] \bmod \Lambda\end{aligned}$$

- For now, ignore that the effective noise is not independent of the codeword. Effective noise variance $N_{\mathsf{EFFEC}} = \alpha^2 N + (1-\alpha)^2 P$.
- Optimal choice of α is the MMSE coefficient $\alpha_{\mathsf{MMSE}} = \dfrac{P}{N+P}$.

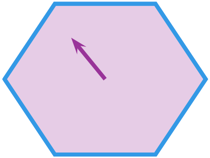
$$N_{\mathsf{EFFEC}} = \alpha_{\mathsf{MMSE}}^2 N + (1-\alpha_{\mathsf{MMSE}})^2 P = \frac{PN}{N+P}$$

$$C = \frac{1}{2}\log\left(\frac{P}{N_{\mathsf{EFFEC}}}\right) = \frac{1}{2}\log\left(1 + \frac{P}{N}\right)$$

- Now the noise is dependent on the codeword.
- Dithering can solve this problem (just as in the discrete case).
- Map message $\mathbf{w}$ to a lattice codeword $\mathbf{t}$.
- Generate a random dither vector $\mathbf{d}$ uniformly over $\mathcal{V}$.
- Transmitter sends a dithered codeword:

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

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



Decoding – Remove Dither First

- Transmitter sends dithered codeword $\mathbf{x} = [\mathbf{t} + \mathbf{d}] \bmod \Lambda$.
- After scaling the channel output $\mathbf{y}$ by α, the decoder subtracts the dither $\mathbf{d}$.

$$
\begin{aligned}
\tilde{\mathbf{y}} &= [\alpha\mathbf{y} - \mathbf{d}] \bmod \Lambda \\
&= [\alpha\mathbf{x} + \alpha\mathbf{z} - \mathbf{d}] \bmod \Lambda \\
&= [\mathbf{x} - \mathbf{d} + \alpha\mathbf{z} - (1-\alpha)\mathbf{x}] \bmod \Lambda \\
&= \Big[[\mathbf{t} + \mathbf{d}] \bmod \Lambda - \mathbf{d} + \alpha\mathbf{z} - (1-\alpha)\mathbf{x}\Big] \bmod \Lambda \\
&= [\mathbf{t} + \alpha\mathbf{z} - (1-\alpha)\mathbf{x}] \bmod \Lambda \qquad \text{Distributive Law}
\end{aligned}
$$

- Effective noise is now independent from the codeword $\mathbf{t}$.
- By the probabilistic method, (at least) one good fixed dither exists. No common randomness necessary.

Summary

- Linear code embedded in the integer lattice:

$$R = \frac{1}{2}\log\left(\frac{P}{N}\right) - \frac{1}{2}\log\left(\frac{2\pi e}{12}\right)$$

- Linear code embedded in the integer lattice, MMSE scaling:

$$R = \frac{1}{2}\log\left(1+\frac{P}{N}\right) - \frac{1}{2}\log\left(\frac{2\pi e}{12}\right)$$

- Linear code embedded in a good shaping lattice, MMSE scaling:

$$R = \frac{1}{2}\log\left(1+\frac{P}{N}\right)$$

Theorem (Erez-Zamir '04)

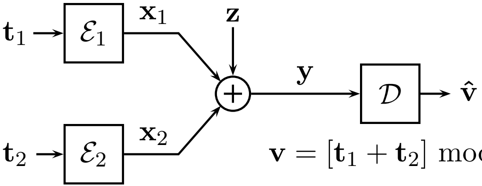
Nested lattice codes can achieve the AWGN capacity.

Decoding the Sum of Lattice Codewords

Encoders use the same nested lattice codebook.

Transmit lattice codewords:

$$\mathbf{x}_1 = \mathbf{t}_1$$
$$\mathbf{x}_2 = \mathbf{t}_2$$



$$\mathbf{v} = [\mathbf{t}_1 + \mathbf{t}_2] \bmod \Lambda$$

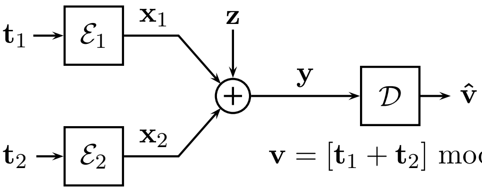
Decoder recovers modulo sum.

$$\begin{aligned}
&[\mathbf{y}] \bmod \Lambda \\
&= [\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}] \bmod \Lambda \\
&= [\mathbf{t}_1 + \mathbf{t}_2 + \mathbf{z}] \bmod \Lambda \\
&= \Big[[\mathbf{t}_1 + \mathbf{t}_2] \bmod \Lambda + \mathbf{z} \Big] \bmod \Lambda \qquad \text{Distributive Law} \\
&= [\mathbf{v} + \mathbf{z}] \bmod \Lambda
\end{aligned}$$

$$R = \frac{1}{2} \log \left(\frac{P}{N} \right)$$

Decoding the Sum of Lattice Codewords – MMSE Scaling

Encoders use the same nested lattice codebook.



$\mathbf{v} = [\mathbf{t}_1 + \mathbf{t}_2] \bmod \Lambda$

Transmit dithered codewords:

$$\mathbf{x}_1 = [\mathbf{t}_1 + \mathbf{d}_1] \bmod \Lambda$$
$$\mathbf{x}_2 = [\mathbf{t}_2 + \mathbf{d}_2] \bmod \Lambda$$

Decoder scales by α, removes dithers, recovers modulo sum.

$$\begin{aligned}
&[\alpha\mathbf{y} - \mathbf{d}_1 - \mathbf{d}_2] \bmod \Lambda \\
&= [\alpha(\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}) - \mathbf{d}_1 - \mathbf{d}_2] \bmod \Lambda \\
&= [\mathbf{x}_1 + \mathbf{x}_2 - (1-\alpha)(\mathbf{x}_1 + \mathbf{x}_2) + \alpha\mathbf{z} - \mathbf{d}_1 - \mathbf{d}_2] \bmod \Lambda \\
&= \Big[[\mathbf{t}_1 + \mathbf{t}_2] \bmod \Lambda - (1-\alpha)(\mathbf{x}_1 + \mathbf{x}_2) + \alpha\mathbf{z}\Big] \bmod \Lambda \\
&= [\mathbf{v} \underbrace{- (1-\alpha)(\mathbf{x}_1 + \mathbf{x}_2) + \alpha\mathbf{z}}_{\text{Effective Noise}}] \bmod \Lambda
\end{aligned}$$

$$N_{\mathsf{EFFEC}} = (1-\alpha)^2 2P + \alpha^2 N$$

Decoding the Sum of Lattice Codewords – MMSE Scaling

- Effective noise after scaling is $N_{\text{EFFEC}} = (1-\alpha)^2 2P + \alpha^2 N$.
- Minimized by setting α to be the MMSE coefficient:

$$\alpha_{\text{MMSE}} = \frac{2P}{N + 2P}$$

- Plugging in, we get

$$N_{\text{EFFEC}} = \frac{2NP}{N + 2P}$$

- Resulting rate is

$$R = \frac{1}{2}\log\left(\frac{P}{N_{\text{EFFEC}}}\right) = \frac{1}{2}\log\left(\frac{1}{2} + \frac{P}{N}\right)$$

- Getting the full "one plus" term is an open challenge. Does not seem possible with nested lattices.

- Map messages to lattice points

$$\mathbf{t}_1 = \phi(\mathbf{w}_1) = [\mathbf{B}\gamma\mathbf{G}\mathbf{w}_1] \bmod \Lambda$$
$$\mathbf{t}_2 = \phi(\mathbf{w}_2) = [\mathbf{B}\gamma\mathbf{G}\mathbf{w}_2] \bmod \Lambda$$

- Mapping between finite field messages and lattice codewords preserves linearity:

$$\phi^{-1}\Big([\mathbf{t}_1 + \mathbf{t}_2] \mod \Lambda\Big) = \mathbf{w}_1 \oplus \mathbf{w}_2$$

- This means that after decoding a $\mod \Lambda$ equation of lattice points we can immediately recover the finite field equation of the messages. See **Nazer-Gastpar '11** for more details.

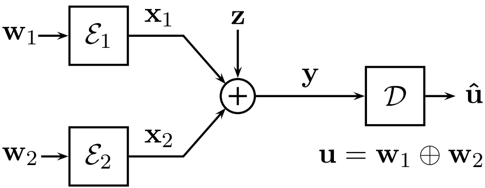
Summary: Finite Field Computation over a Gaussian MAC

Map messages to lattice points:

$$\mathbf{t}_1 = \phi(\mathbf{w}_1)$$
$$\mathbf{t}_2 = \phi(\mathbf{w}_2)$$

Transmit dithered codewords:

$$\mathbf{x}_1 = [\mathbf{t}_1 + \mathbf{d}_1] \bmod \Lambda$$
$$\mathbf{x}_2 = [\mathbf{t}_2 + \mathbf{d}_2] \bmod \Lambda$$

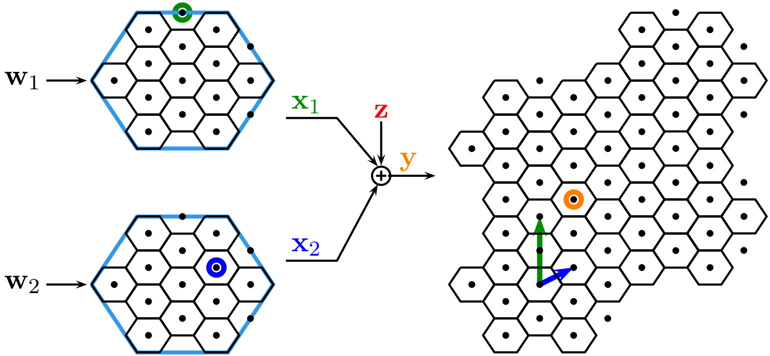


- If decoder can recover $[\mathbf{t}_1 + \mathbf{t}_2] \bmod \Lambda$, it also can get the sum of the messages

$$\mathbf{w}_1 \oplus \mathbf{w}_2 = \phi^{-1}\Big([\mathbf{t}_1 + \mathbf{t}_2] \mod \Lambda\Big) \, .$$

- Achievable rate $R = \frac{1}{2}\log\left(\frac{1}{2} + \frac{P}{N}\right)$.

All users pick the same nested lattice code: Choose messages over field $\mathbf{w}_i \in \mathbb{F}_p^k$: Map $\mathbf{w}_i$ to lattice point in Λ_{FINE} mod Λ_{COARSE}: Transmit lattice points over the channel: Decode the sum:



Decoding is successful whenever $R \leq \frac{1}{2}\log_2\left(\frac{1}{2} + \mathsf{SNR}\right)$

Theorem

For the K-user Gaussian MAC with unit gains, a receiver can decode $\sum \mathbf{w}_i$ at rate:

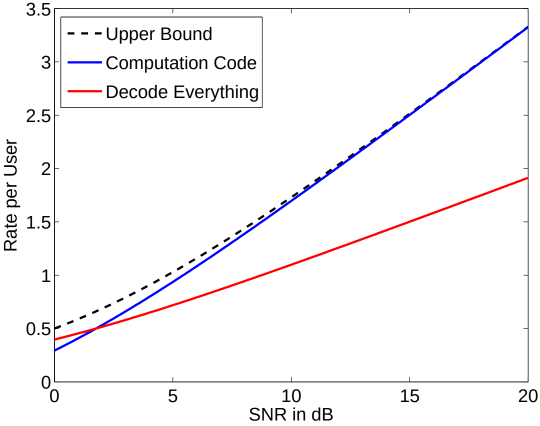
$$R = \frac{1}{2} \log \left(\frac{1}{K} + \frac{P}{N} \right)$$

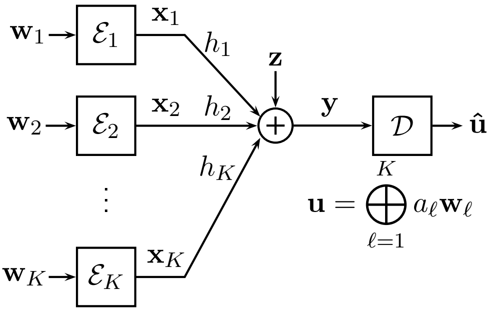
Note: Constructive proof requires lattices generated from q-ary codes, where q is generally arbitrarily large.

Lattice Codes for Compute-and-Forward: Direct Sum

- Want sum of messages $\sum_{i=1}^{M} \mathbf{w}_i$
- Channel is perfectly matched $\mathbf{y} = \sum_{i=1}^{M} \mathbf{x}_i + \mathbf{z}$

$$M = 2$$




$\mathbf{w}_1$
$\mathcal{E}_1$
$\mathbf{x}_1$
h_1
$\mathbf{w}_2$
$\mathcal{E}_2$
$\mathbf{x}_2$
h_2
$\mathbf{w}_K$
$\mathcal{E}_K$
$\mathbf{x}_K$
h_K
$\mathbf{z}$
$\mathbf{y}$
$\mathcal{D}$
$\hat{\mathbf{u}}$
$\mathbf{u} = \bigoplus_{\ell=1}^{K} a_\ell \mathbf{w}_\ell$

Computation over Fading Channels

- 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 removes dithers and decodes an integer combination which can be mapped back to $\mathbb{F}_q$ to recover $\bigoplus_\ell a_\ell \mathbf{w}_\ell$.

$$\begin{aligned}
&\Big[\mathbf{y} - \sum_{\ell=1}^{L} a_\ell \mathbf{d}_\ell\Big] \bmod \Lambda \\
&= \Big[\sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z} - \sum_{\ell=1}^{L} a_\ell \mathbf{d}_\ell\Big] \bmod \Lambda \\
&= \Big[\sum_{\ell=1}^{L} a_\ell(\mathbf{x}_\ell - \mathbf{d}_\ell) + \sum_{\ell=1}^{L}(h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}\Big] \bmod \Lambda \\
&= \Big[\Big[\sum_{\ell=1}^{L} a_\ell \mathbf{t}_\ell\Big] \bmod \Lambda + \underbrace{\sum_{\ell=1}^{L}(h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}}_{\text{Effective Noise}}\Big] \bmod \Lambda \qquad \text{Distributive Law}
\end{aligned}$$

Computation over Fading Channels – Effective Noise

- Effective noise due to mismatch between channel coefficients $\mathbf{h} = [h_1 \cdots h_L]^T$ and equation coefficients $\mathbf{a} = [a_1 \cdots a_L]^T$.

$$N_{\mathsf{EFFEC}} = 1 + \mathsf{SNR}\|\mathbf{h} - \mathbf{a}\|^2$$

$$R = \frac{1}{2}\log\left(\frac{\mathsf{SNR}}{1 + \mathsf{SNR}\|\mathbf{h} - \mathbf{a}\|^2}\right)$$

- Can do better with MMSE scaling.

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

$$R = \max_{\alpha} \frac{1}{2}\log\left(\frac{\mathsf{SNR}}{\alpha^2 + \mathsf{SNR}\|\alpha\mathbf{h} - \mathbf{a}\|^2}\right)$$

$$= \frac{1}{2}\log\left(\frac{1 + \mathsf{SNR}\|\mathbf{h}\|^2}{\|\mathbf{a}\|^2 + \mathsf{SNR}(\|\mathbf{h}\|^2\|\mathbf{a}\|^2 - (\mathbf{h}^T\mathbf{a})^2)}\right)$$

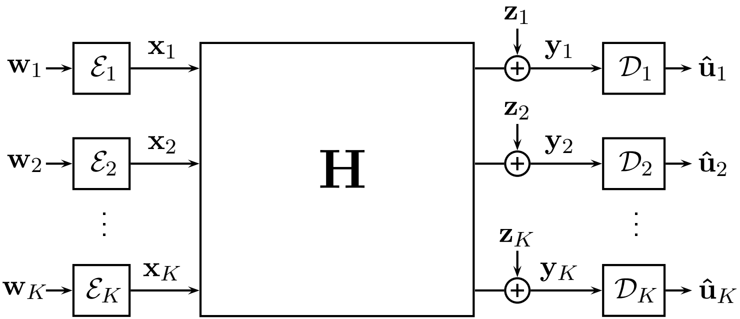
- Practical codes and constellations: **Feng-Silva-Kschischang '10**, **Hern and Narayanan '11**, **Ordentlich and Erez '10**, **Osmane and Belfiore '11**

Theorem (Nazer-Gastpar 2009, IT Trans 2011)

For the Gaussian MAC with coefficients $\mathbf{h} = [h_1 \cdots h_L]^T$, unknown *to the transmitters, it is possible to decode the finite-field sum of the messages with coefficients* $\mathbf{a} = [a_1 \cdots a_L]^T$ *at rate*

$$R = \max_{\alpha} \frac{1}{2} \log \left(\frac{\mathsf{SNR}}{\alpha^2 + \mathsf{SNR}\|\alpha \mathbf{h} - \mathbf{a}\|^2} \right)$$

$$= \frac{1}{2} \log \left(\frac{1 + \mathsf{SNR}\|\mathbf{h}\|^2}{\|\mathbf{a}\|^2 + \mathsf{SNR}(\|\mathbf{h}\|^2\|\mathbf{a}\|^2 - (\mathbf{h}^T\mathbf{a})^2)} \right)$$



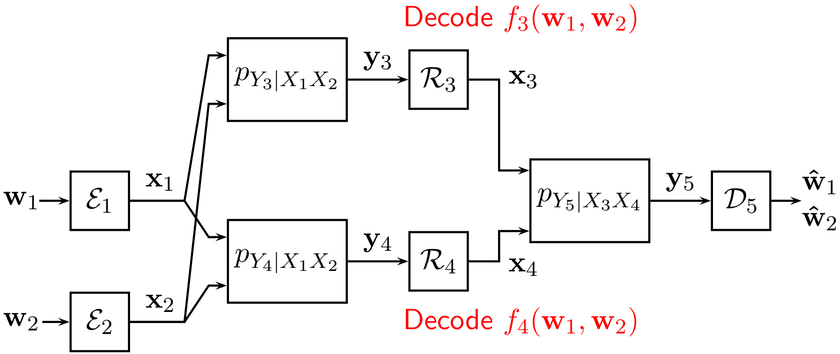
- No channel state information (CSI) at transmitters.
- Receivers use CSI to select coefficients, decode linear equation

$$\mathbf{u}_k = \bigoplus_{\ell=1}^{K} a_{k\ell}\mathbf{w}_\ell$$

- Reliable decoding possible if

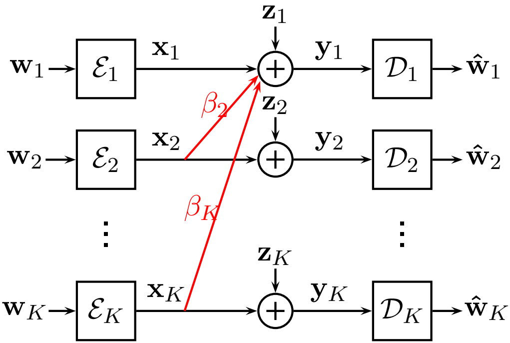
$$R < \min_{k: a_{k\ell} \neq 0} \frac{1}{2} \log\left(\frac{N + P\|\mathbf{h}_k\|^2}{N\|\mathbf{a}_k\|^2 + P(\|\mathbf{h}_k\|^2\|\mathbf{a}_k\|^2 - (\mathbf{h}_k^T\mathbf{a}_k)^2)} \right)$$

Interference: The Compute-and-Forward Perspective



- Only receiver 1 sees interference:

$$\mathbf{y}_1 = \mathbf{x}_1 + \sum_{\ell=2}^{K} \beta_\ell \mathbf{x}_\ell + \mathbf{z}_1$$



- "Compute-and-Forward" Approach: Encoders use the same nested lattice codebook.
- Decoder $\mathcal{D}_1$ decodes linear equations of the messages.
- Additional twist: After decoding an equation, we can (partially) remove it from the received signal.

Many-to-One Interference Channel

- Only receiver 1 sees interference:

$$\mathbf{y}_1 = \mathbf{x}_1 + \sum_{\ell=2}^{K} \beta_\ell \mathbf{x}_\ell + \mathbf{z}_1$$

Let us denote $\mathbf{b} = (1, \beta_2, \ldots, \beta_K)$.

- It first decodes the equation

$$q_1^{(1)}\mathbf{w}_1 + q_2^{(1)}\mathbf{w}_2 + \ldots + q_K^{(1)}\mathbf{w}_K$$

where we collect the integer coefficients into the vector $\mathbf{q}^{(1)}$.

- As we have seen, this works if the rate R is chosen to satisfy

$$R \leq \max_{\alpha} \log^+\left(\frac{P}{\|\alpha\mathbf{b} - \mathbf{q}^{(1)}\|^2 P + \alpha^2 N}\right)$$

- Next, we form

$$\mathbf{y}_1^{(2)} = \mathbf{x}_1 + \sum_{\ell=2}^{K} \beta_\ell \mathbf{x}_\ell + \mathbf{z}_1 - \alpha_1' \left(\sum_{\ell=1}^{K} \mathbf{q}_\ell^{(1)} \mathbf{x}_\ell \right)$$

- From this, we decode

$$q_1^{(2)} \mathbf{w}_1 + q_2^{(2)} \mathbf{w}_2 + \ldots + q_K^{(2)} \mathbf{w}_K$$

  where we collect the integer coefficients into the vector $\mathbf{q}^{(2)}$.
- Again, this works if the rate R is chosen to satisfy

$$R \leq \max_{\alpha_1', \alpha_2} \log^+ \left(\frac{P}{\|\alpha_2(\mathbf{b} - \alpha_1' \mathbf{q}^{(1)}) - \mathbf{q}^{(2)}\|^2 P + \alpha_2^2 N} \right)$$

  which we prefer to trivially rewrite as

$$R \leq \max_{\alpha_1, \alpha_2} \log^+ \left(\frac{P}{\|\alpha_2 \mathbf{b} - \alpha_1 \mathbf{q}^{(1)} - \mathbf{q}^{(2)}\|^2 P + \alpha_2^2 N} \right)$$

- Next, we form

$$\mathbf{y}_1^{(3)} = \mathbf{x}_1 + \sum_{\ell=2}^{K} \beta_\ell \mathbf{x}_\ell + \mathbf{z}_1 - \alpha'_1 \left(\sum_{\ell=1}^{K} \mathbf{q}_\ell^{(1)} \mathbf{x}_\ell \right) - \alpha'_2 \left(\sum_{\ell=1}^{K} \mathbf{q}_\ell^{(2)} \mathbf{x}_\ell \right)$$

- From this, we decode

$$q_1^{(3)} \mathbf{w}_1 + q_2^{(3)} \mathbf{w}_2 + \ldots + q_K^{(3)} \mathbf{w}_K$$

  where we collect the integer coefficients into the vector $\mathbf{q}^{(3)}$.

- Again, this works if the rate R is chosen to satisfy

$$R \leq \max_{\alpha'_1, \alpha'_2, \alpha_3} \frac{1}{2} \log^+ \left(\frac{P}{\|\alpha_3(\mathbf{b} - \alpha'_2 \mathbf{q}^{(2)} - \alpha'_1 \mathbf{q}^{(1)}) - \mathbf{q}^{(3)}\|^2 P + \alpha_3^2 N} \right)$$

  which we prefer to trivially rewrite as

$$R \leq \max_{\alpha_1, \alpha_2, \alpha_3} \frac{1}{2} \log^+ \left(\frac{P}{\|\alpha_3 \mathbf{b} - \alpha_2 \mathbf{q}^{(2)} - \alpha_1 \mathbf{q}^{(1)} - \mathbf{q}^{(3)}\|^2 P + \alpha_3^2 N} \right)$$

- At this point, we have decoded three equations, with coeffcients $\mathbf{q}^{(1)}, \mathbf{q}^{(2)}$, and $\mathbf{q}^{(3)}$, respectively.
- Of course, this is only useful if we can now use these to recover the message $\mathbf{w}_1$.
- Suppose we have

$$\begin{aligned} \mathbf{q}^{(1)} &= (1, 1, 2) \\ \mathbf{q}^{(2)} &= (1, -5, 1) \\ \mathbf{q}^{(3)} &= (-1, 3, -5) \end{aligned}$$

  Can we recover $\mathbf{w}_1$?
- Construct the $3 \times K$ matrix

$$\mathbf{Q} = \begin{pmatrix} \mathbf{q}^{(1)} \\ \mathbf{q}^{(2)} \\ \mathbf{q}^{(3)} \end{pmatrix}$$

  Let us denote the set of those matrices for which one can recover the first component (i.e., $\mathbf{w}_1$) by $\mathcal{Q}_1$.

- Construct the $3 \times K$ matrix

$$\mathbf{Q} \;=\; \begin{pmatrix} \mathbf{q}^{(1)} \\ \mathbf{q}^{(2)} \\ \mathbf{q}^{(3)} \end{pmatrix}$$

Definition

Let $\mathcal{Q}_1$ be the set of those matrices for which one can recover the first component (i.e., $\mathbf{w}_1$).

- **Exercise:** Give an explicit characterization of $\mathcal{Q}_1$. (For the $3 \times K$ case, and then for the general $L \times K$ case.)
- *Hint:* Consider the matrix $\mathbf{Q}'$, obtained from $\mathbf{Q}$ by removing the first column.

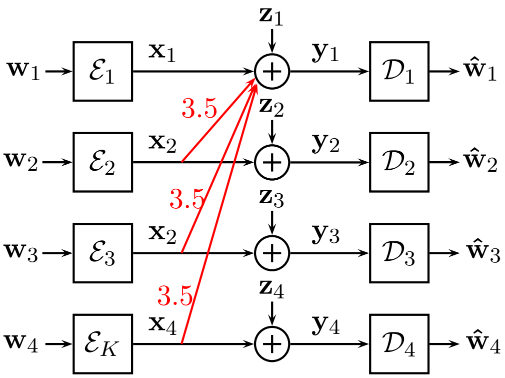
Theorem (Zhu-Gastpar, ISIT'13)

The following rates are achievable for the many-to-one interference networks

$$\begin{aligned} R_1 &\leq \max_{\substack{L\in[1:K] \\ \mathbf{Q}\in\mathcal{Q}_{\mathbf{1}}}} \min_{l\in[1:L]} R_{comp}(\mathbf{q}^{(\ell)}, \mathbf{q}^{(\ell-1)}, \ldots, \mathbf{q}^{(1)}) \\ R_k &\leq \min\left\{\frac{1}{2}\log\left(1+P\right), R_1\right\} \text{ for } k\in[2:K] \end{aligned}$$

where

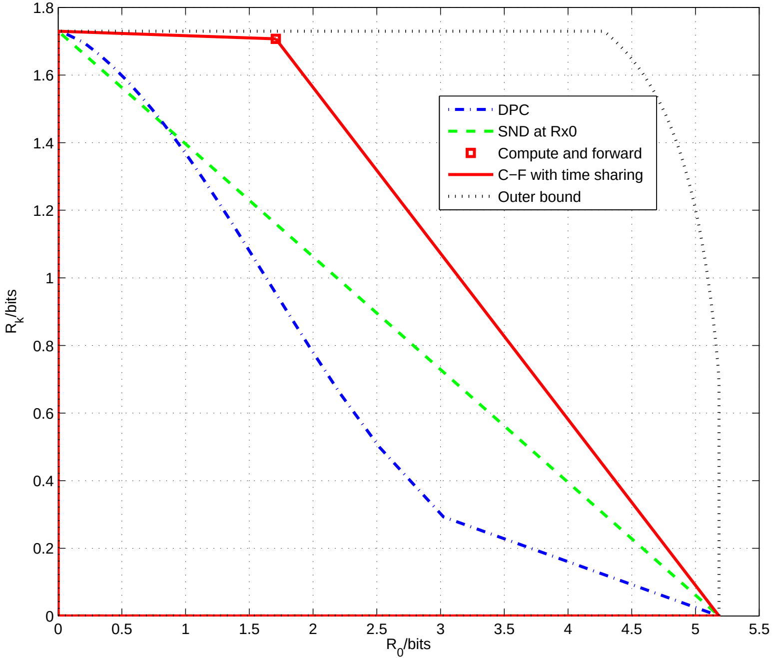
$$\begin{aligned} & R_{comp}(\mathbf{q}^{(\ell)}, \mathbf{q}^{(\ell-1)}, \ldots, \mathbf{q}^{(1)}) \\ &= \max_{\alpha_1,\ldots,\alpha_\ell} \frac{1}{2}\log^+\left(\frac{P}{\left\|\alpha_\ell\mathbf{b} - \sum_{j=1}^{\ell-1}\alpha_j\mathbf{q}^{(j)} - \mathbf{q}^{(\ell)}\right\|^2 P + \alpha_\ell^2 N}\right). \end{aligned}$$



- Now, let $P = 10$.
- Then, the best equations turn out to be (in this order!)

$$\mathbf{q}^{(1)} = (1, 3, 3, 3), \quad \text{leading to} \quad R_{comp} = 1.707$$
$$\mathbf{q}^{(2)} = (3, 10, 10, 10), \quad \text{leading to} \quad R_{comp} = 1.782.$$

Many-to-One Interference Channel

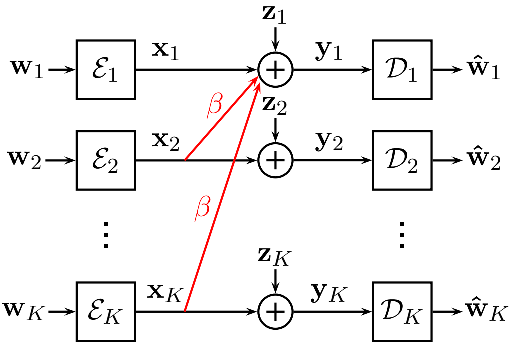


- Equal rates R.
- Good equations:

$$\mathbf{q}^{(1)} = (0, 1, 1, \ldots, 1),$$

$$\mathbf{q}^{(2)} = (1, 0, 0, \ldots, 0).$$

From the theorem, we find...



Many-to-One Interference Channel – Symmetric Very Strong Case

- How big does β have to be to achieve $R = \frac{1}{2}\log\left(1 + \frac{P}{N}\right)$? (i.e. "very strong" case)
- Baseline scheme: Decode $\mathbf{w}_2, \ldots, \mathbf{w}_K$ at receiver 1 and remove prior to decoding $\mathbf{w}_1$.

$$R \leq \frac{1}{2(K-1)} \log\left(1 + \frac{\beta^2 (K-1) P}{N+P}\right)$$

  Hence,

$$\beta^2 \geq \frac{\left((1 + \frac{P}{N})^{K-1} - 1\right)(N+P)}{(K-1)P}$$

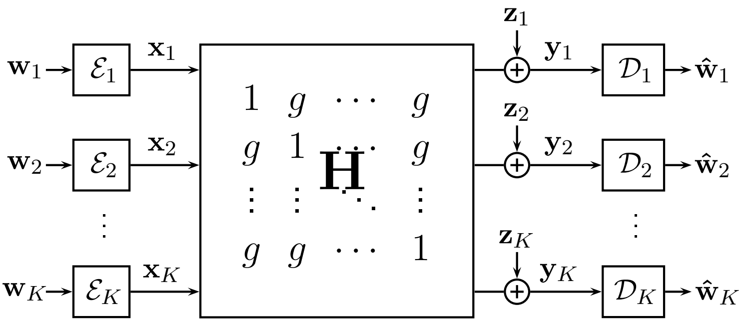
- By contrast, for the "compute-and-forward" scheme:

$$\beta^2 \geq \frac{(P+N)^2}{PN}$$

- Originally shown in **Sridharan-Jafarian-Vishwanath-Jafar '08** using spherical shaping region. Nested lattice scheme: **Nazer-Gastpar '11**.

- Further results can be obtained for the many-to-one interference channel.
- *Example:* Lattices codes combined with the deterministic model can approach the capacity region to within $(3K+3)(1+\log(K+1))$ bits per user. (Bresler-Parekh-Tse '10).

Symmetric K-User Interference Channel



- Each transmitter wants to send a message to a single receiver.
- Possibility of interference alignment **Cadambe-Jafar '08, Motahari et al. '09**.
- 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**.
- Focus on the special case of symmetric cross-gains.

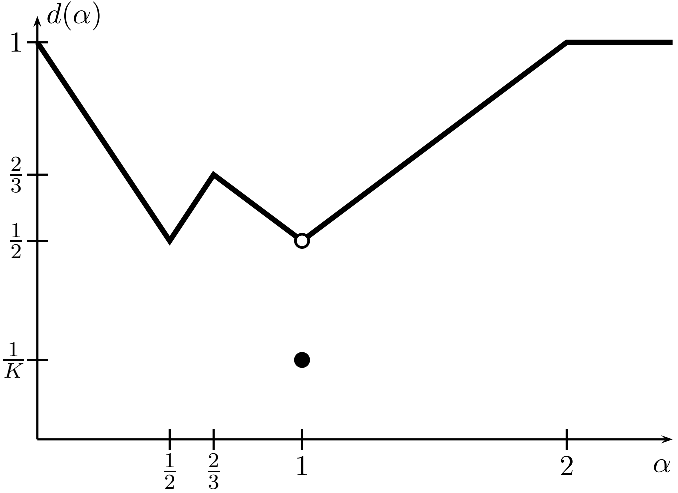
Effective Multiple-Access Channel

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

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

- Idea: 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**.
- With the compute-and-forward transform we can approximate the sum capacity in all regimes.

Generalized Degrees-of-Freedom



- 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,

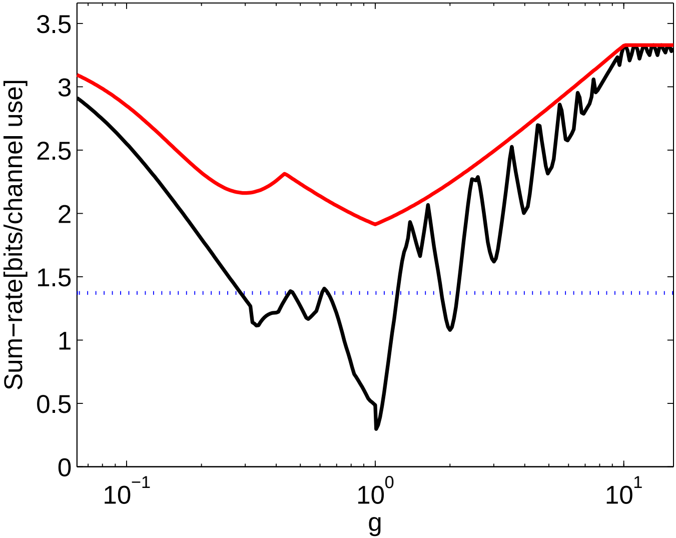
$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

- Decode two equations:

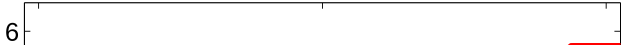
$$a_1 \mathbf{x}_k + a_2 \sum_{\ell \neq k} \mathbf{x}_\ell \qquad\qquad b_1 \mathbf{x}_k + b_2 \sum_{\ell \neq l} \mathbf{x}_\ell$$

- This allows us to operate close to the symmetric capacity, unlike successive cancellation.

20dB



35dB



- Using this technique, we can characterize the approximate sum capacity of the symmetric K-user interference channel. See **Ordentlich-Erez-Nazer '12 arXiv:1206.0197**.

- Typical result:
  Strong Interference Regime, $1 \leq \alpha < 2$,

$$\frac{1}{4}\log(\mathsf{INR}) - \frac{c+5}{2} \leq C_{\mathsf{sym}} \leq \frac{1}{4}\log(\mathsf{INR}) + \frac{1}{2}$$

  for all channel gains except for an outage set of measure $\mu < 2^{-c}$ for any $c > 0$.

I. Interference

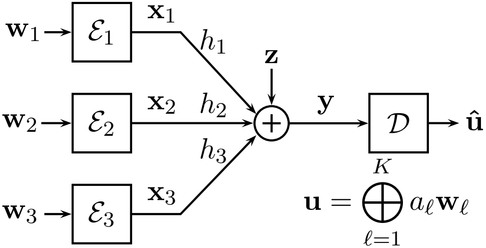
II. Compute-and-Forward

III. Interference: The Compute-and-Forward Perspective

IV. Single-Hop Networks
  (a) Fixed Channel Characteristics ("Single Channel")
  (b) Varying Channel Characteristics ("Parallel Channels")

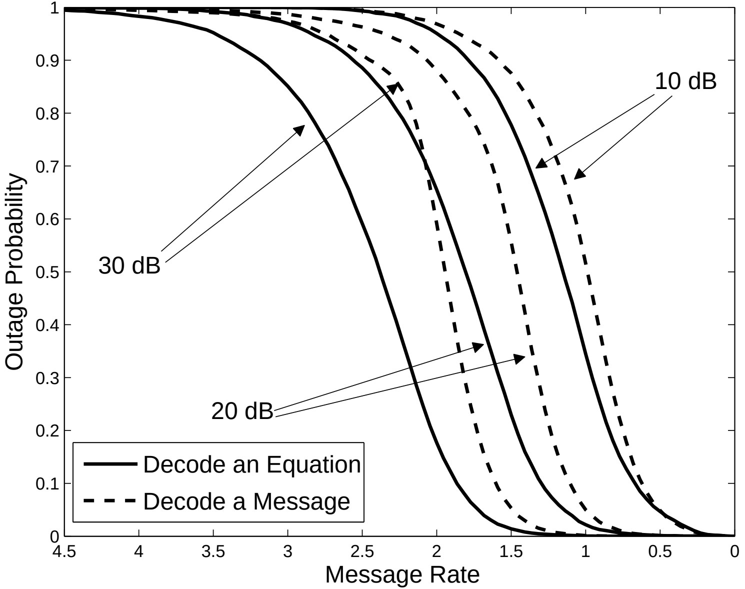
V. Multi-hop Networks

Time-Varying Channels, Unknown at Tx



- Fading coefficients h_1, h_2, h_3 are iid Gaussian, unknown to the transmitters.
- Fix a certain rate R. Decode *either* the equation of your choice *or* one of the messages (which is a special case of an equation...).
- With what probability does the channel *not* support the rate R that you fixed? ("Outage probability")

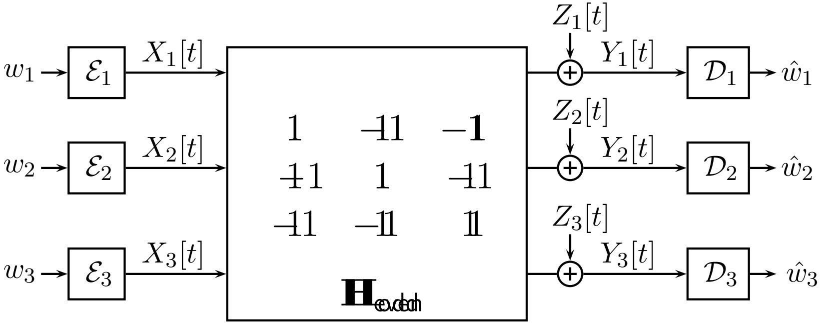
Time-Varying Channels, Unknown at Tx



Time-Varying Channels, Unknown at Tx

- One can also study the *average* rate, averaged over the fading behavior.
- But we here proceed to the case when the channel is **known** at the Tx.

Time-Varying Channels, Known at Tx



- Decoder 1: $w_1 + w_2 - w_3$ and $w_1 - w_2 + w_3$ at $R = \frac{1}{2}\log_2(\frac{1}{3} + \frac{P}{N})$
- Decoder 2: $-w_1 + w_2 + w_3$ and $w_1 + w_2 - w_3$ at $R = \frac{1}{2}\log_2(\frac{1}{3} + \frac{P}{N})$
- Decoder 3: $w_1 - w_2 + w_3$ and $-w_1 + w_2 + w_3$ at $R = \frac{1}{2}\log_2(\frac{1}{3} + \frac{P}{N})$

- Hence, each user gets a rate of

$$R = \frac{1}{2} \cdot \frac{1}{2} \log_2(\frac{1}{3} + \frac{P}{N}).$$

- Actually, we can do a little better: Simply *add up* the analog channel outputs from even and odd channel. This leads to a new interference channel:

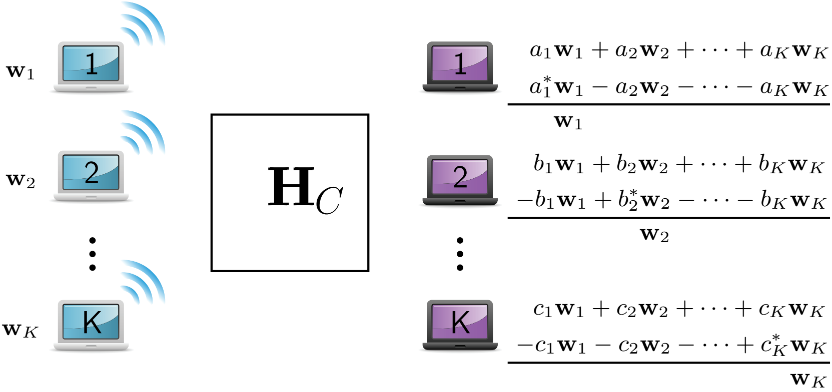
$$\begin{aligned} Y_1 &= 2X_1 + Z_1 + Z_1' \\ Y_2 &= 2X_2 + Z_2 + Z_2' \\ Y_3 &= 2X_3 + Z_3 + Z_3' \end{aligned}$$

The per-user rate is now

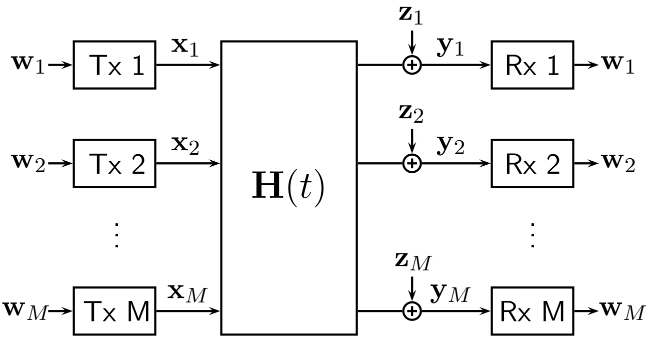
$$R = \frac{1}{2} \cdot \frac{1}{2} \log_2(1 + \frac{2P}{N}),$$

which can be shown to be exactly the (sum-rate) capacity of the considered network.

Time-Varying Channels, Known at Tx



For example, when the channel matrix changes over time...



- Time-varying fading with i.i.d. uniform phases.
- Transmitters know $\mathbf{H}(t)$ before time t.

Key Idea

1. At time t with channel $\mathbf{H}$, user k transmits signal X_k.

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1K} \\ h_{21} & h_{22} & \cdots & h_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ h_{K1} & h_{K2} & \cdots & h_{KK} \end{bmatrix}$$

2. When complementary matrix $\mathbf{H}_C$ occurs, retransmit signals X_k.

$$\mathbf{H}_C = \begin{bmatrix} h_{11} & -h_{12} & \cdots & -h_{1K} \\ -h_{21} & h_{22} & \cdots & -h_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ -h_{K1} & -h_{K2} & \cdots & h_{KK} \end{bmatrix} \pm \delta$$

3. Otherwise, transmit new signals and wait for their $\mathbf{H}_C$.

Pairing Up Channels

- We need to match up almost every matrix with its complement.
- Want a finite set of possible matrices $\mathcal{H}$ for analysis:
  1. Quantize each channel coefficient to precision δ (closest point in $\delta(\mathbb{Z} + j\mathbb{Z})$).
  2. Set threshold h_{MAX}. Throw out any matrix with $|h_{k\ell}| > h_{\mathsf{MAX}}$.
- Choose δ, h_{MAX} to get desired rate gap.
- Since phase is i.i.d. uniform, $P(\mathbf{H}) = P(\mathbf{H}_C)$.

Convergence in Type

Sequence of channel matrices $\mathbf{H}^n$ is ϵ-typical if:

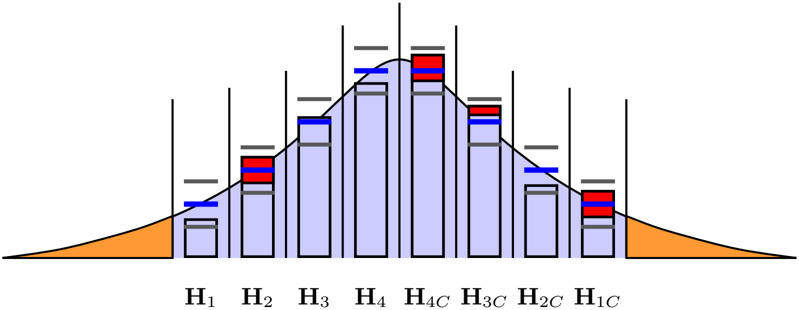
$$\left| \frac{1}{n} N(\mathsf{H} | \mathbf{H}^n) - P(\mathsf{H}) \right| \leq \epsilon \quad \forall \mathsf{H} \in \mathcal{H}$$

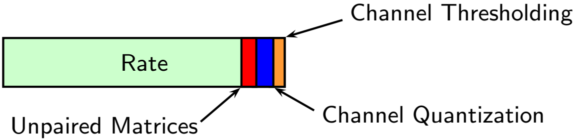
Lemma (Csiszar-Körner 2.12)

For any i.i.d. sequence, $\mathbf{H}^n$, the probability of the set of all ϵ-typical sequences, A_ϵ^n, is lower bounded by:

$$P(A_\epsilon^n) \geq 1 - \frac{|\mathcal{H}|}{4n\epsilon^2}$$

Convergence in Type





1. At time t with channel $\mathbf{H}$, user k transmits signal X_k.

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1K} \\ h_{21} & h_{22} & \cdots & h_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ h_{K1} & h_{K2} & \cdots & h_{KK} \end{bmatrix}$$

2. When complementary matrix $\mathbf{H}_C$ occurs, retransmit signals X_k.

$$\mathbf{H}_C = \begin{bmatrix} h_{11} & -h_{12} & \cdots & -h_{1K} \\ -h_{21} & h_{22} & \cdots & -h_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ -h_{K1} & -h_{K2} & \cdots & h_{KK} \end{bmatrix} \pm\delta$$

3. Otherwise, transmit new signals and wait for their $\mathbf{H}_C$.

In this special case, there is an even simpler solution...

$$\begin{bmatrix} Y_1(t) \\ Y_2(t) \\ \vdots \\ Y_K(t) \end{bmatrix}$$

$$\begin{bmatrix} Y_1(t_C) \\ Y_2(t_C) \\ \vdots \\ Y_K(t_C) \end{bmatrix} \qquad \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ \vdots & \vdots \\ h_{K1} & h_{K2} \end{bmatrix}$$

$$+ \left(\begin{bmatrix} h_{11} & -h_{12} & \cdots & -h_{1K} \\ -h_{21} & h_{22} & \cdots & -h_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ -h_{K1} & -h_{K2} & \cdots & h_{KK} \end{bmatrix} \pm \delta \right) \mathbf{X} \quad + \quad \mathbf{Z}(t_C)$$

$$\begin{bmatrix} Y_1(t) + Y_1(t_C) \end{bmatrix} \quad \left(\begin{bmatrix} 2h_{11} & 0 & \cdots & 0 \end{bmatrix} \right)$$

Interference Channel Ergodic Capacity

Theorem (Nazer-Gastpar-Jafar-Vishwanath *IEEE Trans Info Theory,* **2012 (ISIT '09))**

Each user can achieve at least half its interference-free capacity at **any signal-to-noise ratio***:*

$$R = \frac{1}{2} E\left[\log\left(1 + 2|h_{mm}|^2 \mathsf{SNR}_m\right)\right] > \frac{1}{2} R_{\textit{FREE}}$$

- "Everybody gets half the cake!"
- For uniform phase fading and a large number of users, scheme achieves the ergodic capacity.
- Can also show this approach achieves the ergodic capacity region for finite field channel models.

I. Interference

II. Compute-and-Forward

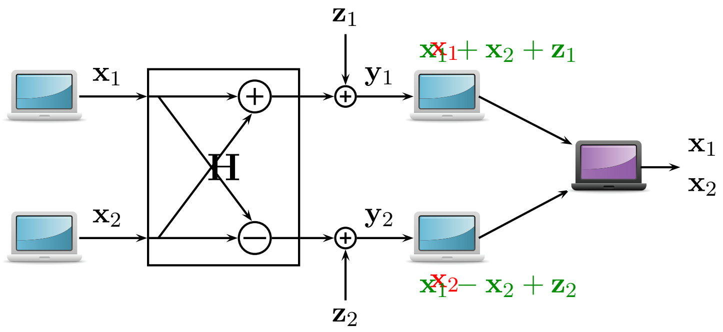
III. Interference: The Compute-and-Forward Perspective

IV. Single-Hop Networks

V. Multi-hop Networks

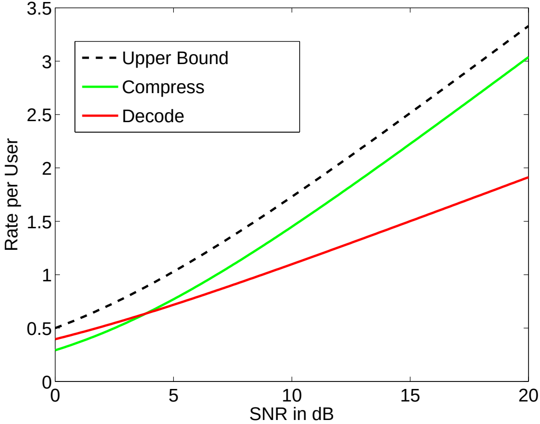
(a) Fixed Channel Characteristics
(b) Varying Channel Characteristics

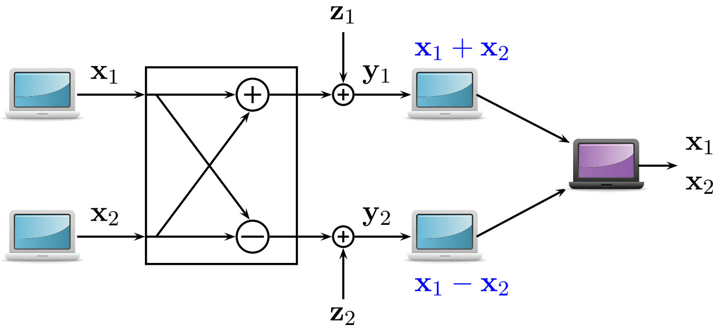
Multi-Hop Networks



- Two users want to send messages across the network with the help of two relays.
- Strategy 1: Each relay decodes one message.
- Strategy 2: Relays send their observed signal to the destination without decoding.

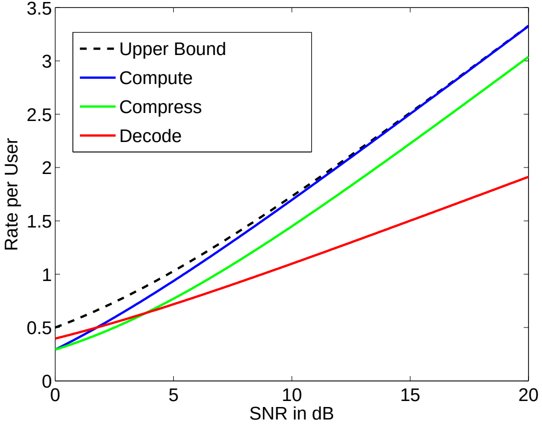
- Interference can be useful!
- Not captured by bit pipe approach.

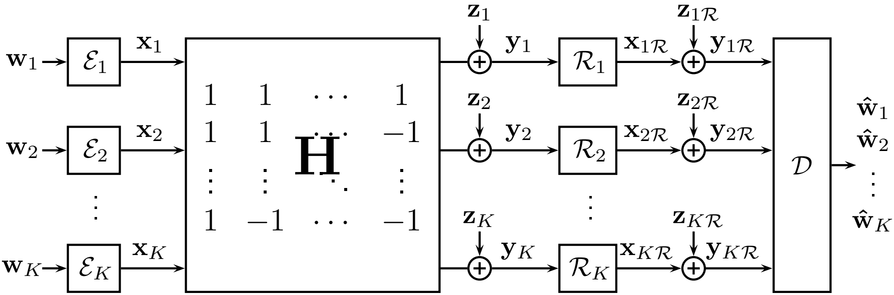




- What if each relay could decode a linear equation?
- Compute-and-Forward: One relay decodes the sum of codewords. Other relay decodes the difference.

- Compute-and-Forward is nearly optimal!





- Equal rates R. $\mathbf{H}$ is a Hadamard matrix, $\mathbf{H}\mathbf{H}^T = K\mathbf{I}$

Upper Bound

$$\frac{1}{2}\log\left(1+\frac{P}{N}\right)$$

Compute-and-Forward

$$\frac{1}{2}\log\left(\frac{1}{K}+\frac{P}{N}\right)$$

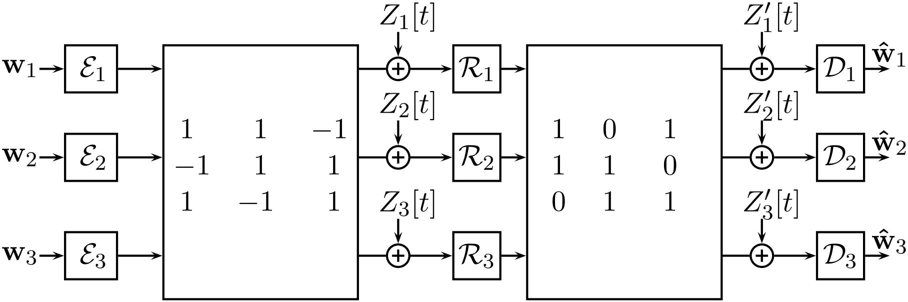
Compress-and-Forward

$$\frac{1}{2}\log\left(1+\frac{P}{N}\frac{P}{N+KP}\right)$$

Decode-and-Forward

$$\frac{1}{2K}\log\left(1+\frac{KP}{N}\right)$$

Multi-Hop Networks



- Relay 1 decodes $\mathbf{w}1 + \mathbf{w}2 - \mathbf{w}3$, etc.
- In this example, because the two matrices are inverses of each other, things work out perfectly. $R = \frac{1}{2}\log_2(\frac{1}{3} + \frac{P}{N})$.
- *Remark:* We could also simply use amplify-and-forward, at the expense of *noise amplification.* Called *Interference Neutralization* (Jeon *et al*, 2011). $R = \frac{1}{2}\log_2(1 + \frac{2P}{3N})$.

I. Interference

II. Compute-and-Forward

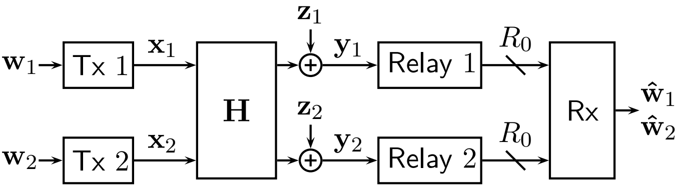
III. Interference: The Compute-and-Forward Perspective

IV. Single-Hop Networks

V. Multi-hop Networks

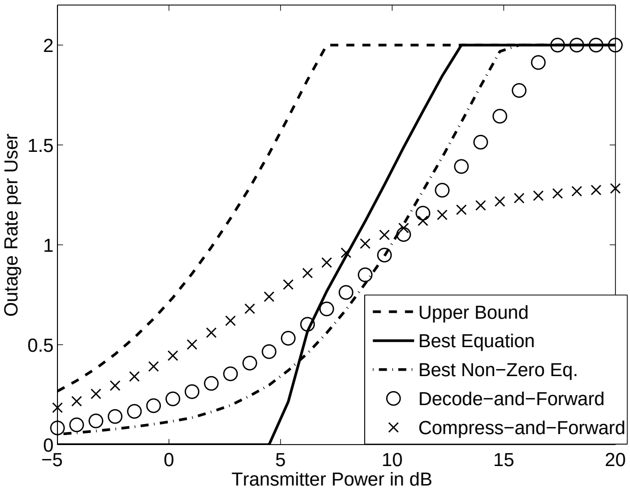
(a) Fixed Channel Characteristics
(b) Varying Channel Characteristics

Rayleigh fading, unknown at Tx

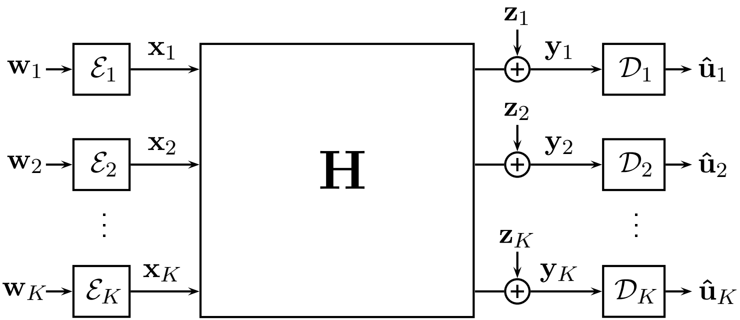


- Rayleigh fading
- No channel state information (CSI) at transmitters.
- Compute-and-Forward strategy: Given CSI, each relay *independently* selects the coefficients for an equation to decode, and forwards this equation.
- Fix transmission rates, what is the probability that the channel cannot support them? ("Outage probability")
- Here, we flip the perspective: fix the outage probability (0.25 in our example), maximize the rate.

Rayleigh fading, unknown at Tx



Rayleigh fading, unknown at Tx



- Rayleigh fading
- No channel state information (CSI) at transmitters.
- **Goal:** K *linearly independent* equations are decoded.

Rayleigh fading, unknown at Tx

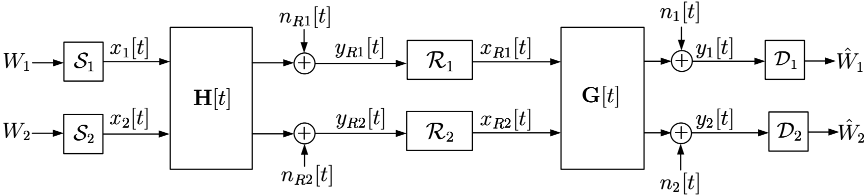
Some "negative" results for the **high SNR** behavior:

- It can be shown that this strategy achieves no more than two (computation) degrees of freedom, irrespective of the number of transmitter/receiver pairs (Niesen, Whiting, 2011).
- This is by contrast to one (message) degree of freedom for Han-Kobayashi.
- It is also by contrast to K degrees of freedom when instead of Rayleigh, the channel matrices are rational.

Time-Varying, **known** *at Tx*

- Now, suppose that the channel is known at the Tx ahead of time.
- Then, we can do interference alignment.

Example:



- We can pair up each matrix $\mathbf{H}$ with its inverse $\mathbf{G}$. (That is, find appropriate time slots.)
- Then, we can apply the interference neutralization trick.
  - Either via compute-and-forward
  - Or via amplify-and-forward, if we are not worried about the noise accumulation.

For the amplify-and-forward strategy under uniform phase fading, we can show the following:

Theorem (**Wang-Jeon-Gastpar, ISIT'12**)

$$R_{\mathrm{MIMO}} = \log(1 + 4P + 2P^2) + \log\left(1 + \sqrt{1 - (C(P))^2}\right) - 1,$$

$$R_{\mathrm{IN}} = 2\log\left(1 + \frac{2P^2}{1 + 4P}\right) + 2\log\left(1 + \sqrt{1 - (C(P))^2}\right) - 2,$$

where $C(P) = 2P^2/(1 + 4P + 2P^2)$. *Furthermore, for any* $P \geq 0$,

$$C_{\mathrm{sum}} - R_{\mathrm{IN}} \leq 4.$$

Note: For Rayleigh fading, we can show that the gap is around 4.7 bits.

Concluding Remarks

- Compute-and-Forward is one quite natural approach to managing interference:
  - The mantra is: "Whenever signals collide/interfere, decode a function of the messages, rather than the messages themselves."
  - If the function to be decoded is "similar" to the channel, there is hope that the resulting rate will be interesting.
- There exist networks where it attains optimal performance (and no other known strategy does).
- There exist practically relevant networks where it attains the best known performance (e.g., distributed antenna systems).
- ...but: so far, the story is pretty much limited to *linearly* colliding signals.
- On the positive side: for the linear case, the practical implementation of Compute-and-Forward is possible essentially with off-the-shelf components!

- B. Nazer and M. Gastpar. Reliable Physical-Layer Network Coding. *Proceedings of the IEEE*, March 2011.
- B. Nazer and M. Gastpar. Compute-and-Forward: Harnessing Interference Through Structured Codes. *IEEE Transactions on Information Theory*, October 2011. (2013 Communications Society & Information Theory Society Joint Paper Award)
- B. A. Nazer and M. Gastpar. Computing over multiple-access channels with connections to wireless network coding. In *Proceedings of the 2006 IEEE International Symposium on Information Theory (ISIT)*, Seattle, WA, July 2006.