Lattices in AWGN Networks: What's missing?

Bobak Nazer

Boston University

BIRS 2011: Algebraic Structure in Network Information Theory

August 16, 2011

Some Disclaimers

- This talk will include some conjectures...

Some Disclaimers

- This talk will include some speculation...

Some Disclaimers

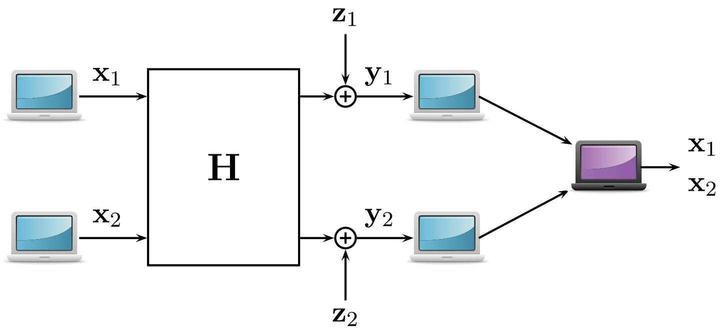
- This talk will include some speculation...
- with the aim of making it a bit more interactive.

Some Disclaimers

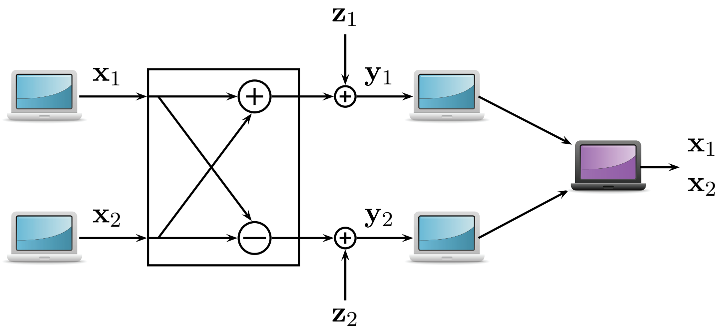
- This talk will include some speculation...
- with the aim of making it a bit more interactive.
- Some of this will be tied to some (local) foods.

Some Disclaimers

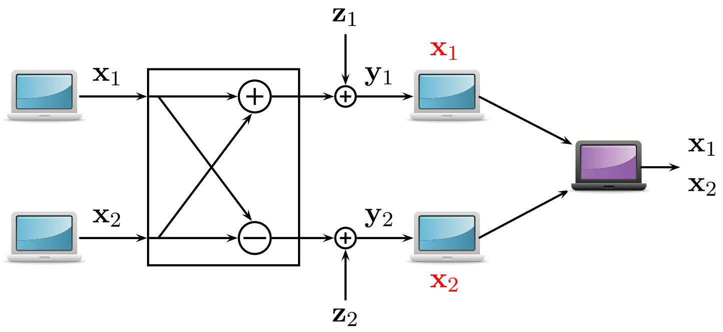
- This talk will include some speculation...
- with the aim of making it a bit more interactive.
- Some of this will be tied to some (local) foods.
- ISIT '11 Tutorial co-taught with Michael Gastpar now available at ISIT website (and `iss.bu.edu/bobak`).



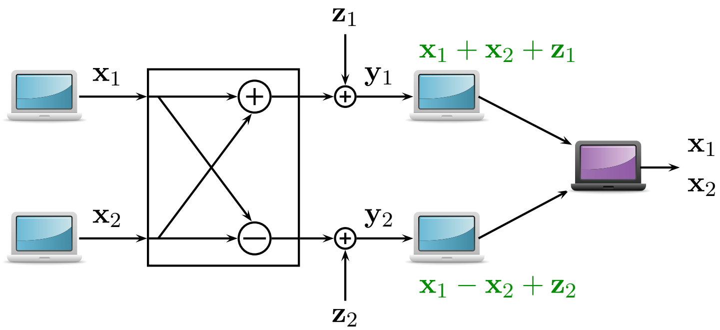
- Two users want to send messages across the network with the help of two relays.



- Two users want to send messages across the network with the help of two relays.

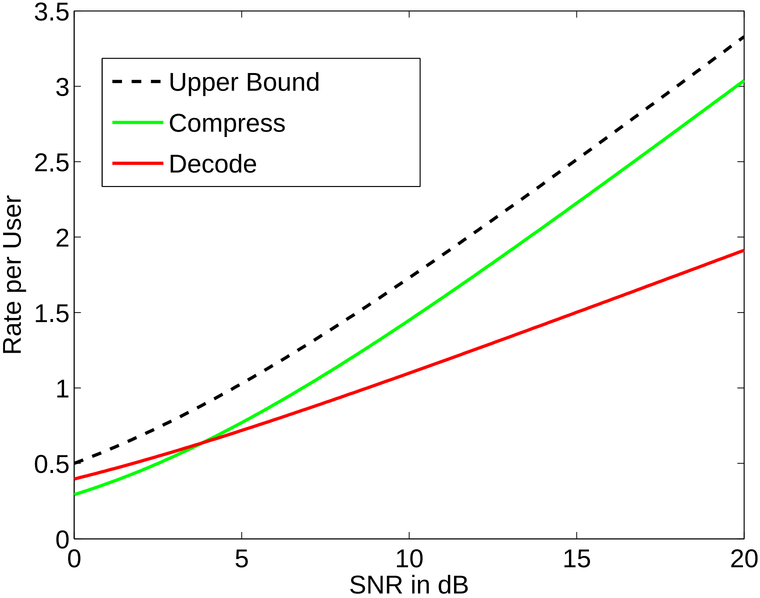


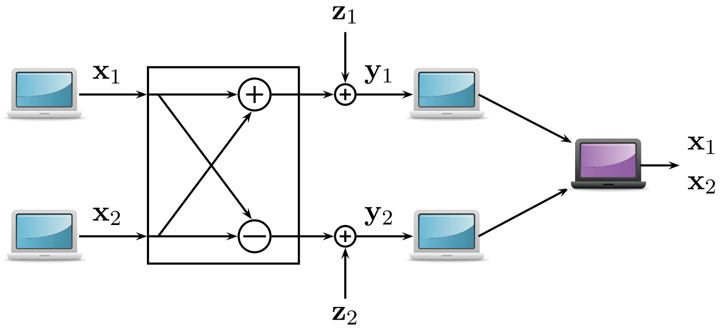
- Two users want to send messages across the network with the help of two relays.
- Decode-and-Forward: Each relay decodes one message.



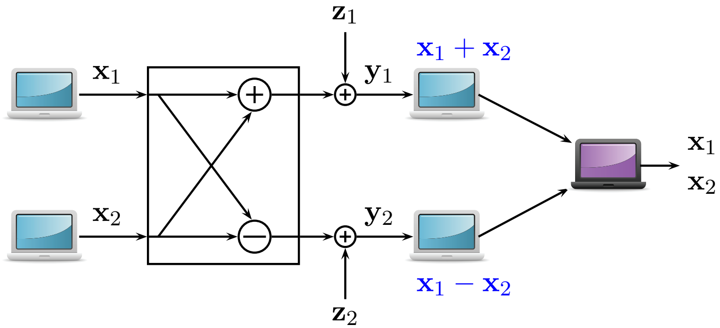
- Two users want to send messages across the network with the help of two relays.
- Decode-and-Forward: Each relay decodes one message.
- Compress-and-Forward: Relays send their observed signal to the destination without decoding.

Compute-and-Forward



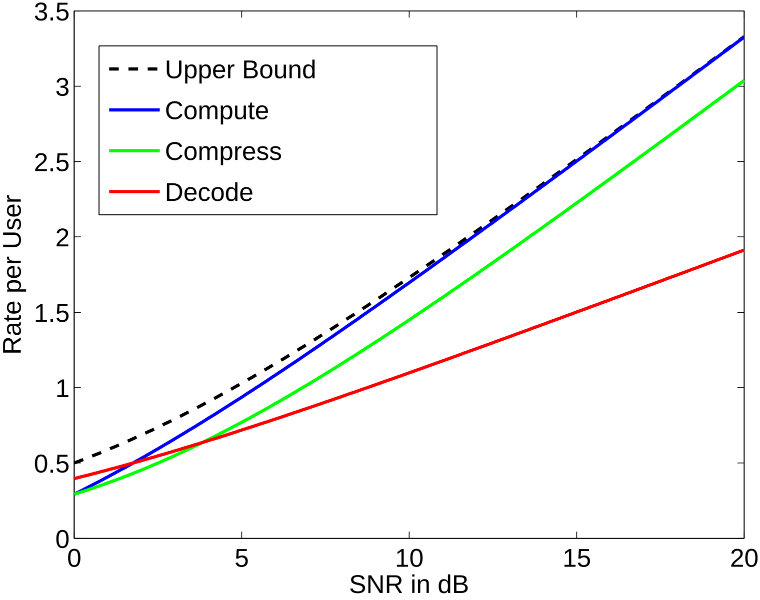


- What if each relay could decode a linear equation?

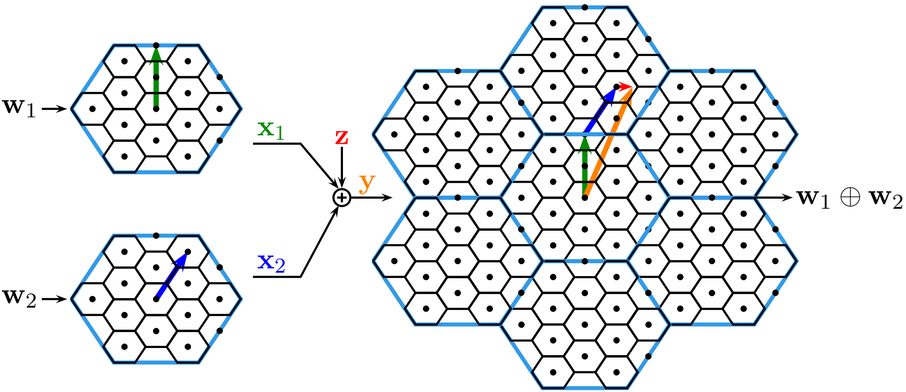


- What if each relay could decode a linear equation?
- Compute-and-Forward: One relay decodes the sum of codewords.
  Other relay decodes the difference.

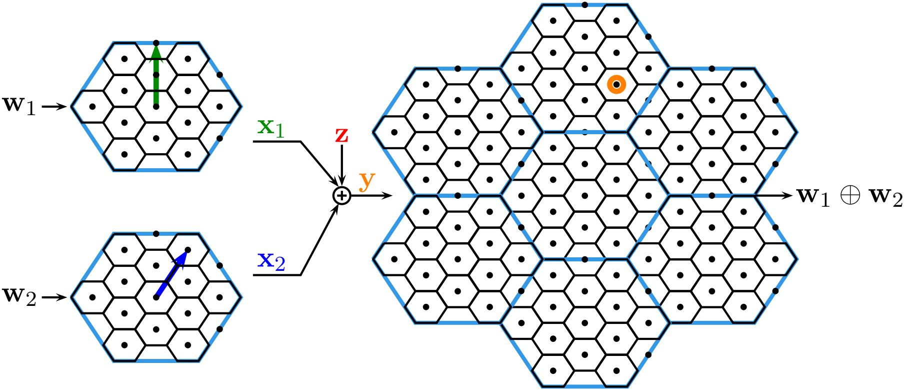
Compute-and-Forward



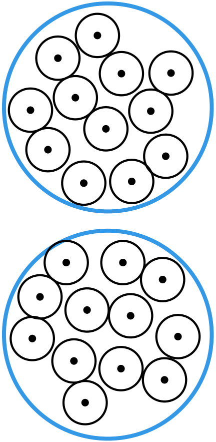
Compute-and-Forward Illustration



Compute-and-Forward Illustration



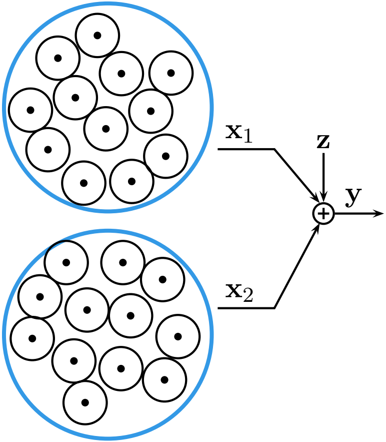
Random i.i.d. codes are not good for computation



2^{nR} codewords each.

2^{n2R} possible sums of codewords.

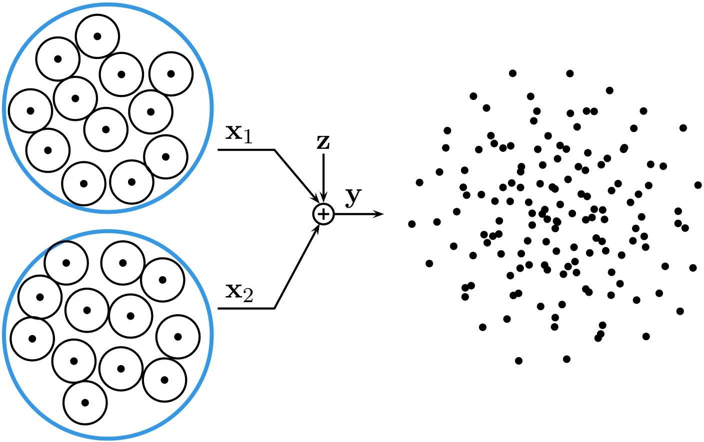
Random i.i.d. codes are not good for computation



2^{nR} codewords each.

2^{n2R} possible sums of codewords.

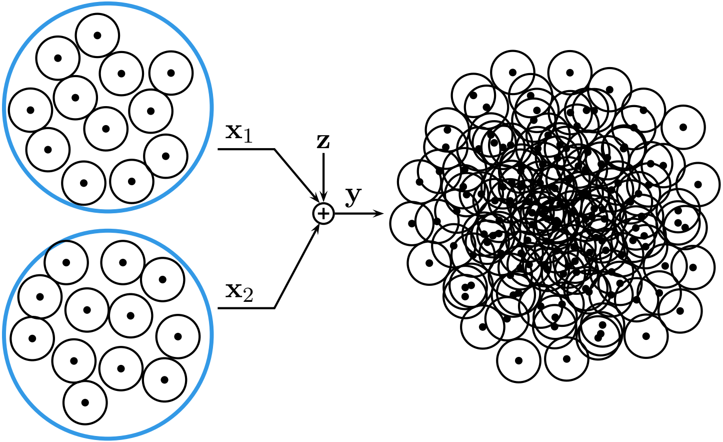
Random i.i.d. codes are not good for computation



2^{nR} codewords each.

2^{n2R} possible sums of codewords.

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



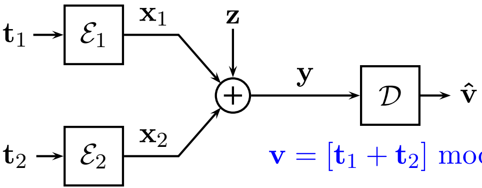
2^{nR} codewords each.

2^{n2R} possible sums of codewords.

Encoders use the same nested lattice codebook from **Erez-Zamir '04.**

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.

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

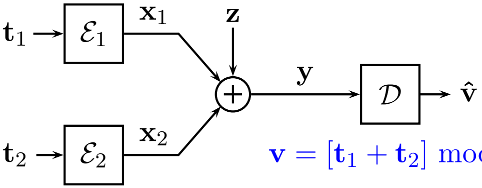
$$R = \frac{1}{2} \log \left(\frac{P}{N} \right)$$

Decoding the Sum of Lattice Codewords – MMSE Scaling

Encoders use the same nested lattice codebook from **Erez-Zamir '04.**

Transmit dithered codewords:

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



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

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

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

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

Decoding the Sum of Lattice Codewords – MMSE Scaling

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

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

- Plugging in, we get

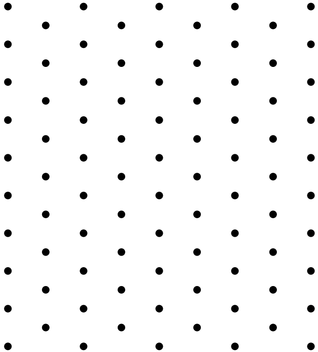
$$N_{\mathsf{EFFEC}} = \frac{2NP}{N + 2P}$$

- Resulting rate is

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

- What happened to the “one plus” term?

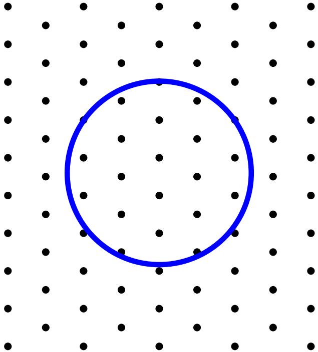
Where is the "one plus"?



Set fine lattice density using:

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

Where is the "one plus"?

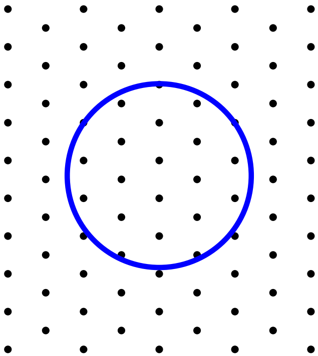


Set fine lattice density using:

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

Cut out codebook at power P.

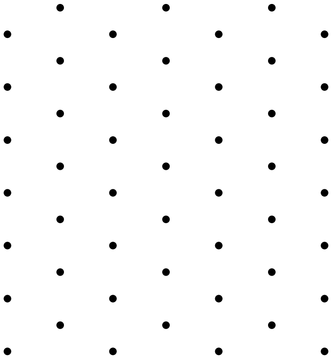
Where is the "one plus"?



Set fine lattice density using:

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

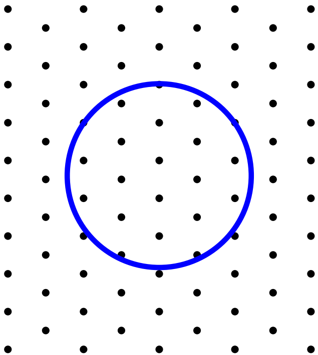
Cut out codebook at power P.



Set fine lattice density using:

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

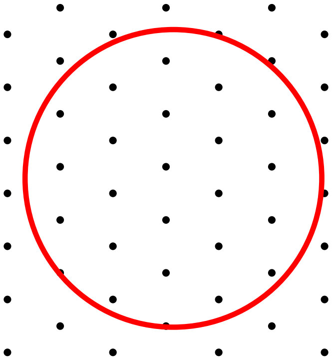
Where is the "one plus"?



Set fine lattice density using:

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

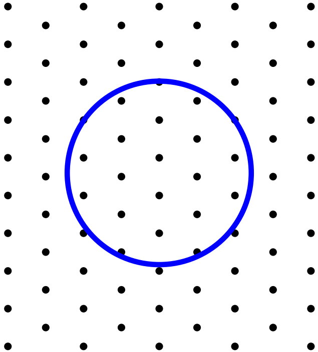
Cut out codebook at power P.



Set fine lattice density using:

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

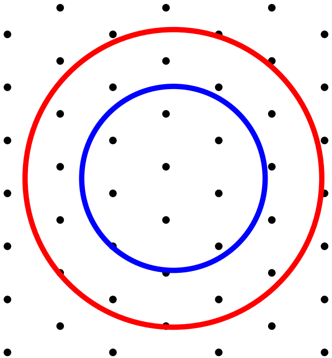
Where is the “one plus”?



Set fine lattice density using:

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

Cut out codebook at power P.

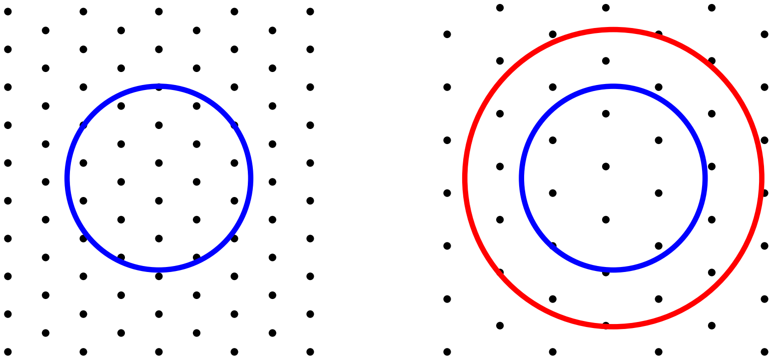


Set fine lattice density using:

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

Cut out codebook at power P.

Where is the "one plus"?



Set fine lattice density using: $R = \frac{1}{2}\log\left(1 + \frac{P}{N}\right)$ Cut out codebook at power P.	Set fine lattice density using: $R = \frac{1}{2}\log\left(1 + \frac{2P}{N}\right)$ Cut out codebook at power P.

Resulting codebook only has $R = \frac{1}{2}\log\left(\frac{1}{2} + \frac{P}{N}\right)$!

Where is the “one plus”?

- This limitation seems inherent to nested lattice codes combined with lattice decoding.
- Also seems inherent to any scheme that treats all codewords as “the same” (e.g. ML decoding). Connected to decoding analysis in **Wilson-Narayanan-Pfister-Sprintson ’10**.
- What about MAP decoding?
- Ice Wine Problem: Prove that the sum of codewords $\mathbf{x}_1 + \mathbf{x}_2$ can be recovered from $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}$ at rate

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

Where is the "one plus"?

- This limitation seems inherent to nested lattice codes combined with lattice decoding.
- Also seems inherent to any scheme that treats all codewords as "the same" (e.g. ML decoding). Connected to decoding analysis in **Wilson-Narayanan-Pfister-Sprintson '10**.
- What about MAP decoding?
- Ice Wine Problem: Prove that the sum of codewords $\mathbf{x}_1 + \mathbf{x}_2$ can be recovered from $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}$ at rate

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

Or prove that this is impossible.

Where is the "one plus"?

- Loss is at most $1/2$ of a bit. Is this such a big deal?

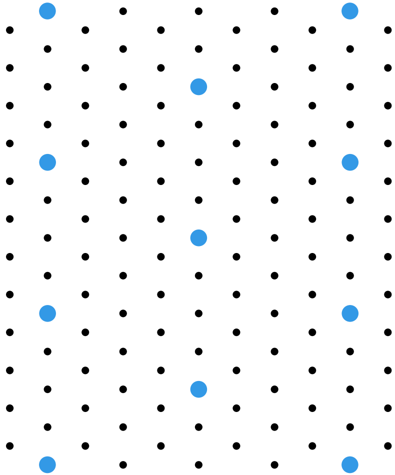
Where is the "one plus"?

- Loss is at most $1/2$ of a bit. Is this such a big deal? Yes, especially for layering.

Where is the "one plus"?

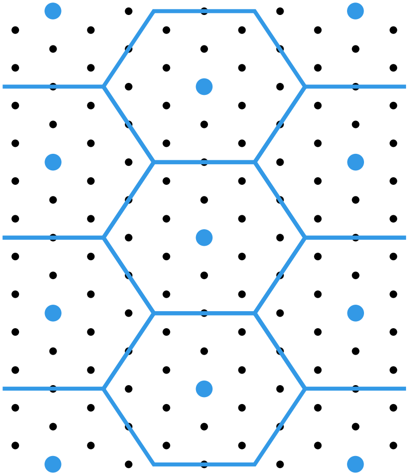
- Loss is at most $1/2$ of a bit. Is this such a big deal? Yes, especially for layering.
- Consider employing a superposition of many lattice codewords.
- As the number of layers increases, the effective SNR of each layer decreases.
- Hard to analyze layered lattice codebooks outside the high SNR regime.
- Shows up in interference channels (e.g. **Sridharan et al. '08**)

- What if the power constraints are not equal?
- Idea from **Nam-Chung-Lee '10**:
- Draw the codewords from the same fine lattice Λ_{FINE}.
- Use two nested coarse lattices Λ_1 and Λ_2 to enforce the power constraints P_1 and P_2.

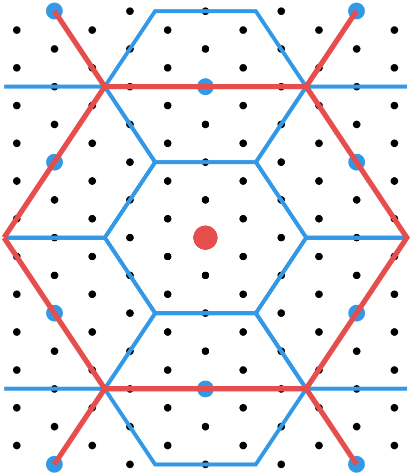


Unequal Power Constraints – Double Nesting

- What if the power constraints are not equal?
- Idea from **Nam-Chung-Lee '10**:
- Draw the codewords from the same fine lattice Λ_{FINE}.
- Use two nested coarse lattices Λ_1 and Λ_2 to enforce the power constraints P_1 and P_2.

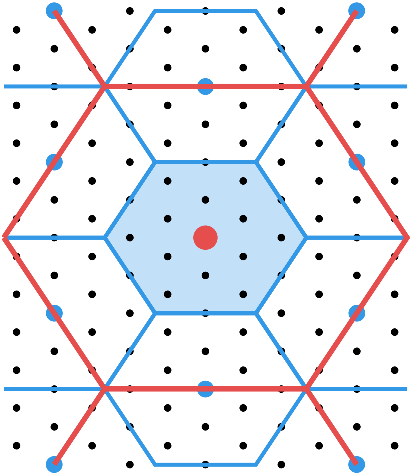


- What if the power constraints are not equal?
- Idea from **Nam-Chung-Lee '10**:
- Draw the codewords from the same fine lattice Λ_{FINE}.
- Use two nested coarse lattices Λ_1 and Λ_2 to enforce the power constraints P_1 and P_2.

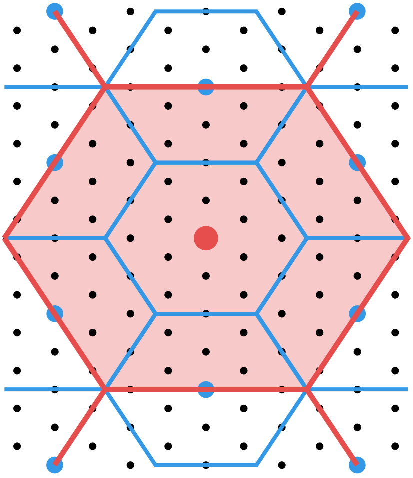


Unequal Power Constraints – Double Nesting

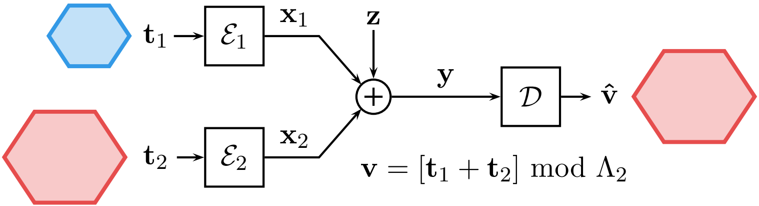
- What if the power constraints are not equal?
- Idea from **Nam-Chung-Lee '10**:
- Draw the codewords from the same fine lattice Λ_{FINE}.
- Use two nested coarse lattices Λ_1 and Λ_2 to enforce the power constraints P_1 and P_2.



- What if the power constraints are not equal?
- Idea from **Nam-Chung-Lee '10**:
- Draw the codewords from the same fine lattice Λ_{FINE}.
- Use two nested coarse lattices Λ_1 and Λ_2 to enforce the power constraints P_1 and P_2.

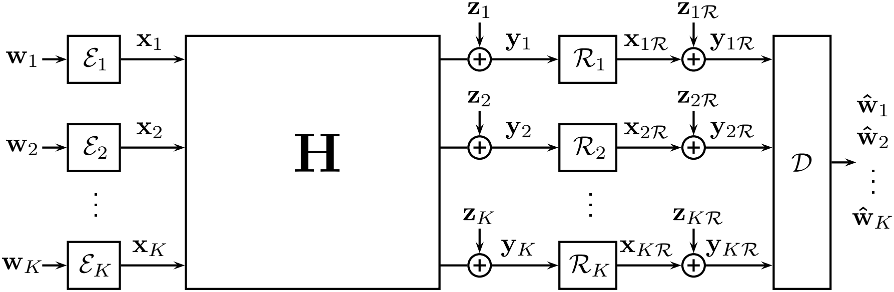


Unequal Power Constraints – Double Nesting



- Encoder 1 sends $\mathbf{x}_1 = [\mathbf{t}_1 + \mathbf{d}_1] \bmod \Lambda_1$. Coarse lattice Λ_1 has second moment P_1.
- Encoder 2 sends $\mathbf{x}_2 = [\mathbf{t}_2 + \mathbf{d}_2] \bmod \Lambda_2$. Coarse lattice Λ_2 has second moment $P_2 > P_1$.
- Decoder performs MMSE scaling, remove dithers, recovers $\bmod \Lambda_2$ sum.

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



- 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

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

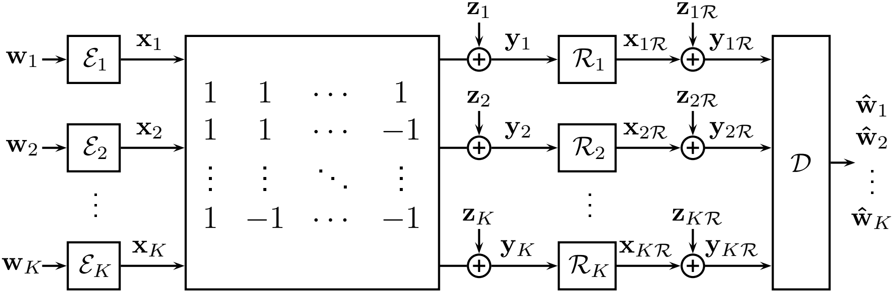
Compress-and-Forward

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

Decode-and-Forward

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

Case Study – Hadamard Relay Network



- 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

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

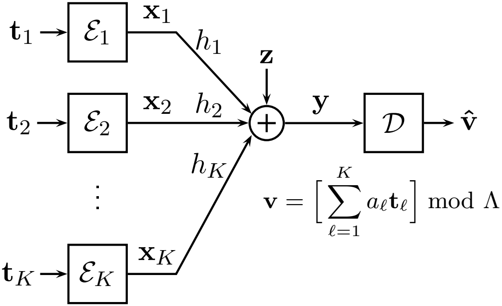
Compute-and-Forward: Fading Channels

Transmitters do not know channel realization.

Encoders use the same nested lattice codebook.

Transmit dithered codewords:

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



- Decoder removes dithers and recovers integer combination

$$\mathbf{v} = \left[\sum_{\ell=1}^{K} a_\ell \mathbf{t}_\ell\right] \bmod \Lambda$$

- Receiver can use its knowledge of the channel gains to match the equation coefficients a_ℓ to the channel coefficients h_ℓ.

Distributive Law

- Distributive Law also holds for integer combinations. Let $a, b \in \mathbb{Z}$.

$$
\begin{aligned}
&\Big[a[\mathbf{x}_1] \bmod \Lambda + b[\mathbf{x}_2] \bmod \Lambda\Big] \bmod \Lambda \\
&= \Big[a\Big(\mathbf{x}_1 - Q_\Lambda(\mathbf{x}_1)\Big) + b\Big(\mathbf{x}_2 - Q_\Lambda(\mathbf{x}_2)\Big)\Big] \bmod \Lambda \\
&= \Big[a\mathbf{x}_1 + b\mathbf{x}_2 - aQ_\Lambda(\mathbf{x}_1) - bQ_\Lambda(\mathbf{x}_2)\Big] \bmod \Lambda \\
&= [a\mathbf{x}_1 + b\mathbf{x}_2] \bmod \Lambda
\end{aligned}
$$

- Last step follows since since $aQ_\Lambda(\mathbf{x}_1)$ and $bQ_\Lambda(\mathbf{x}_2)$ are elements of the lattice Λ.

- Transmit dithered codewords $\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda$
- Decoder removes dithers and recovers integer combination

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

Compute-and-Forward: Fading Channels – Effective Noise

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

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

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

Compute-and-Forward: Fading Channels – Effective Noise

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

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

- Can do better with MMSE scaling.

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

- See **Nazer-Gastpar ’11** for more details.

- The rate expression simplifies in some special cases.

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

- Integer channels: $\mathbf{h} = \mathbf{a}$.

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

- Recovering a single message: Set $\mathbf{a} = \delta_m$, the m^{th} unit vector.

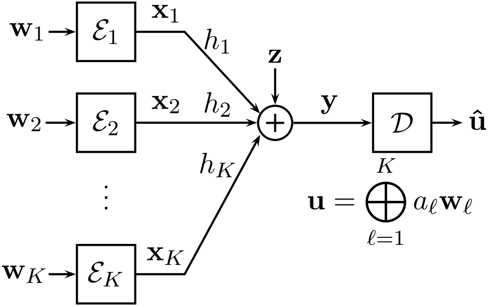
$$R = \frac{1}{2}\log\left(1 + \frac{h_m^2 P}{N + P\sum_{\ell \neq m} h_\ell^2}\right)$$

Transmitters do not know channel realization.

Encoders use the same nested lattice codebook.

Transmit dithered codewords:

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

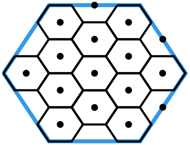


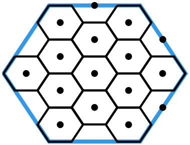
- Due to Construction A, mapping $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$ between messages and lattice points preserves linearity.

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

- Digital interface that fits well with **network coding**.

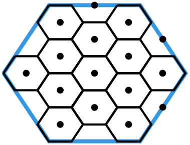
All users pick the same nested lattice code:

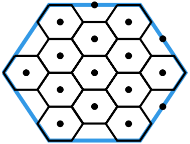




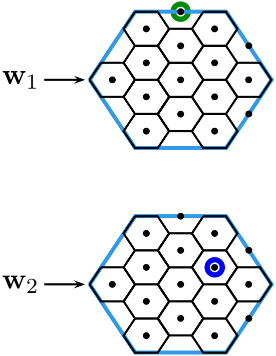
Compute-and-Forward: Fading Channels – Illustration

Choose messages over field $\mathbf{w}_\ell \in \mathbb{F}_q^k$:

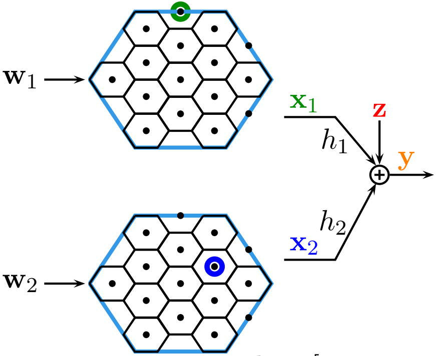
$\mathbf{w}_1 \longrightarrow$ 

$\mathbf{w}_2 \longrightarrow$ 

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



Transmit lattice points over the channel:

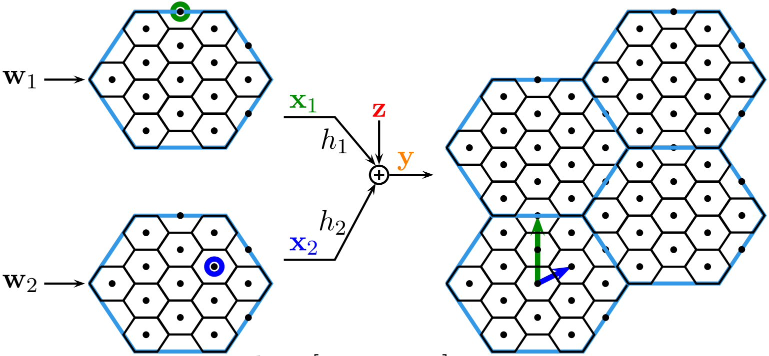


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

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

Compute-and-Forward: Fading Channels – Illustration

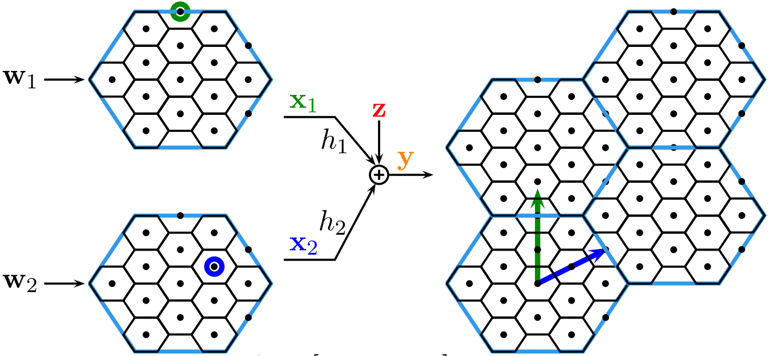
Transmit lattice points over the channel:



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

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

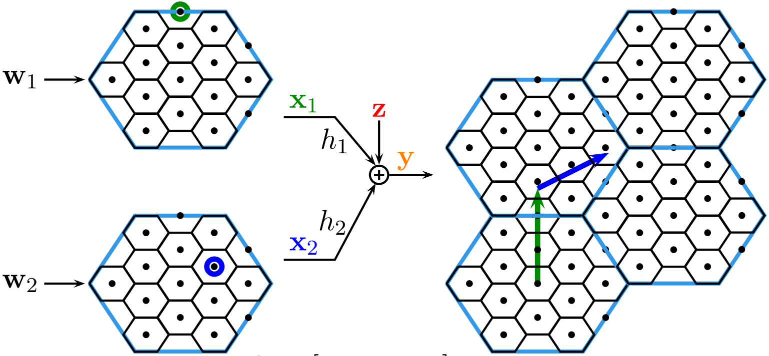
Lattice codewords are scaled by channel coefficients:



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

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

Scaled codewords added together plus noise:

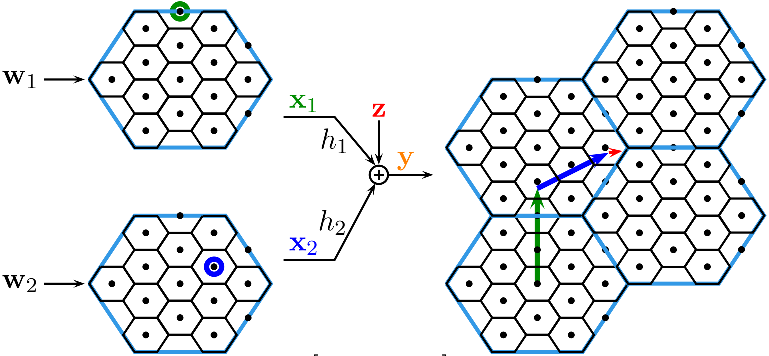


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

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

Compute-and-Forward: Fading Channels – Illustration

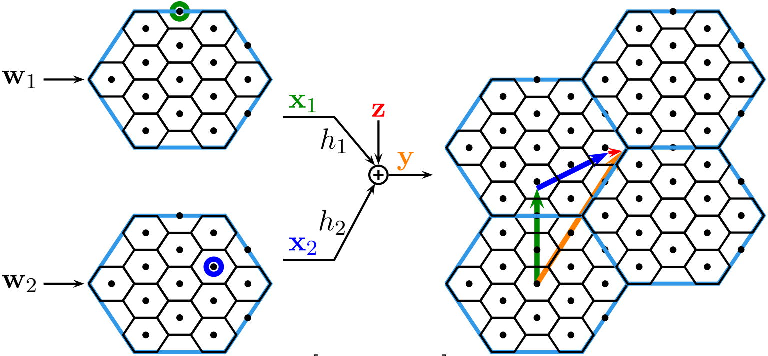
Scaled codewords added together plus noise:



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

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

Extra noise penalty for non-integer channel coefficients:

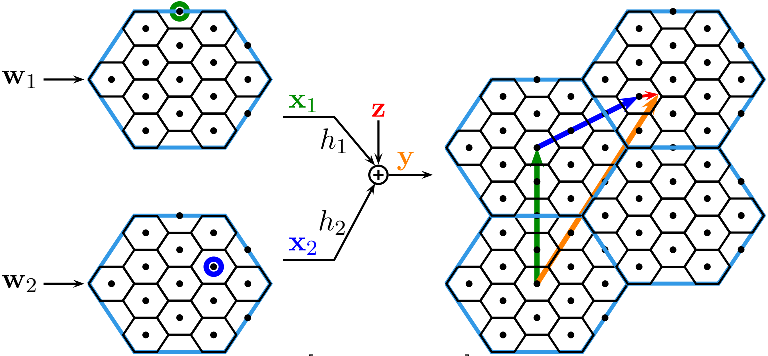


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

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

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

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

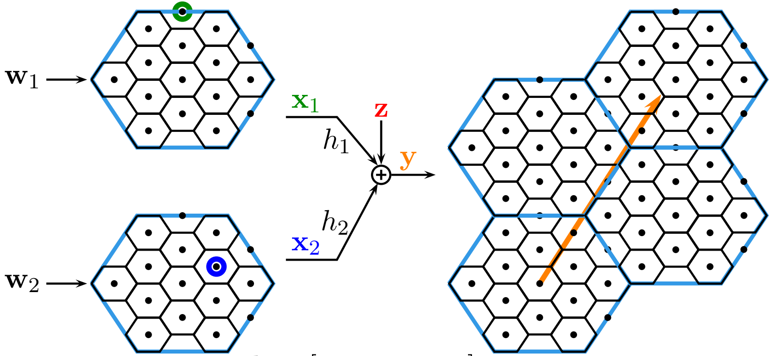


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

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

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

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

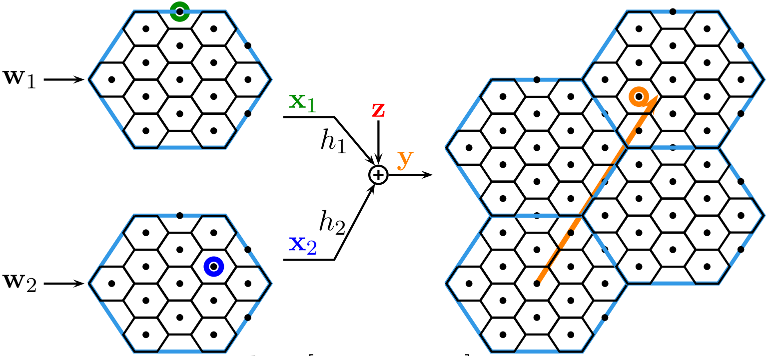


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

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

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

Decode to closest lattice point:

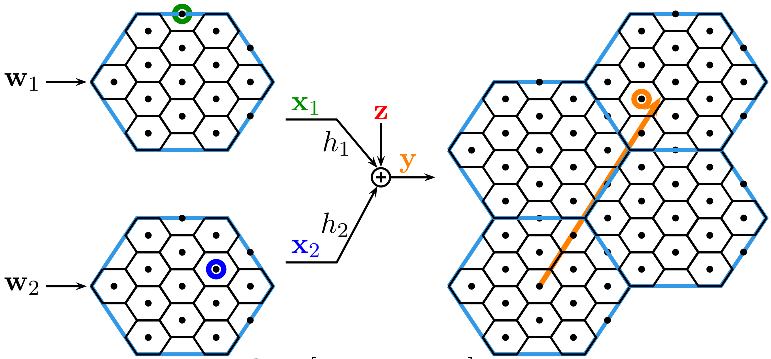


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

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

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

Compute sum of lattice points modulo the coarse lattice:

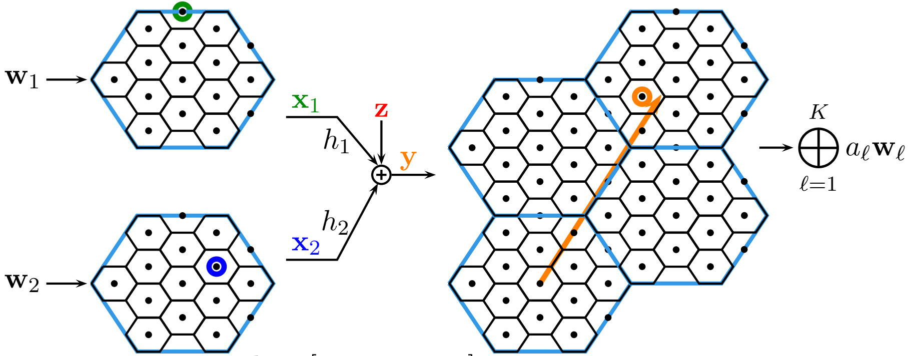


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

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

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

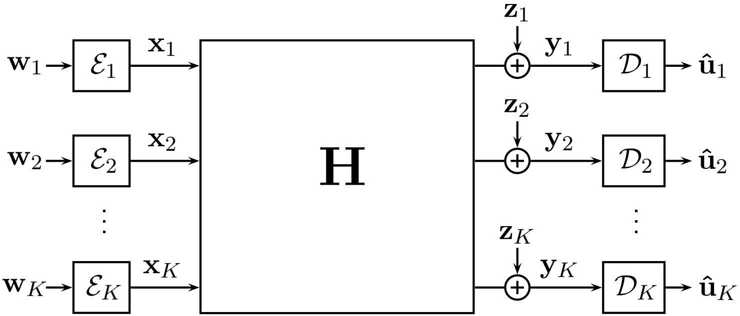
Map back to equation of message symbols over the field:



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

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

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

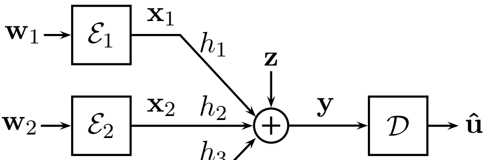


- Equal rates R. No channel state information (CSI) at transmitters.
- Receivers use their CSI to select coefficients, decode linear equation

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

- Reliable decoding possible if

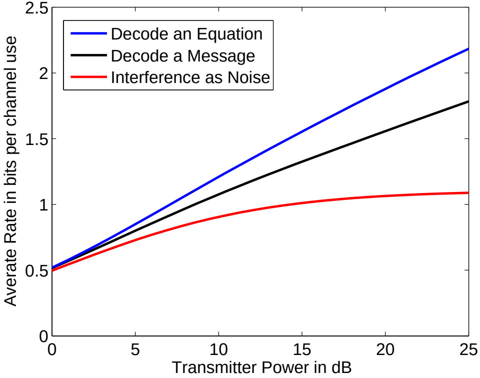
$$R < \min_{k : a_{k\ell} \neq 0} \frac{1}{2} \log\left(\frac{N + P\|\mathbf{h}_k\|^2}{N\|\mathbf{a}_k\|^2 + P(\|\mathbf{h}_k\|^2\|\mathbf{a}_k\|^2 - (\mathbf{h}_k^T\mathbf{a}_k)^2)}\right)$$



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

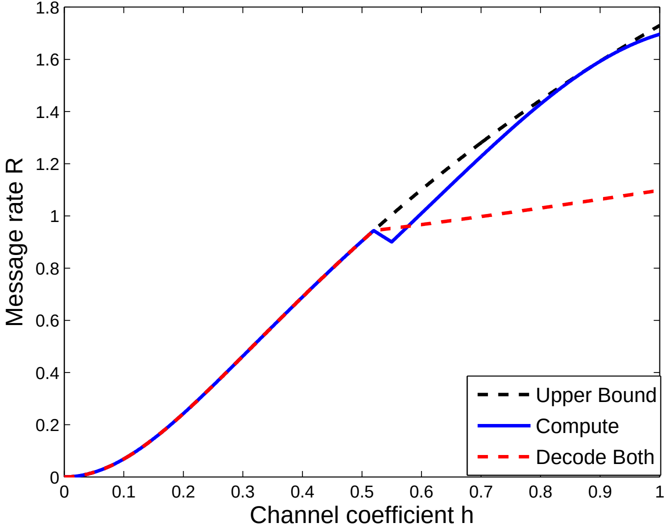
Relay either decodes some linear function of messages or an individual message.

- Three transmitters that do not know the fading coefficients.
- Average rate plotted for i.i.d. Gaussian fading.



- Receiver observes $\mathbf{y} = \mathbf{x}_1 + h\mathbf{x}_2 + \mathbf{z}$.
- Recovers $a\mathbf{w}_1 \oplus b\mathbf{w}_2$ for $a, b \neq 0$.

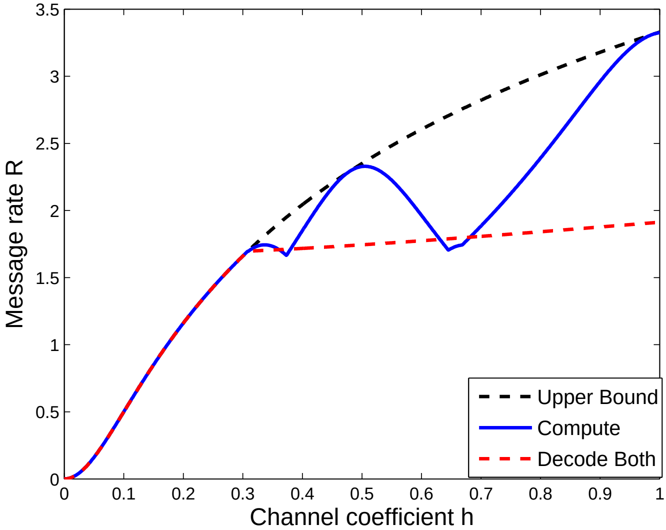
10dB



Computation over Fading Channels – No CSIT

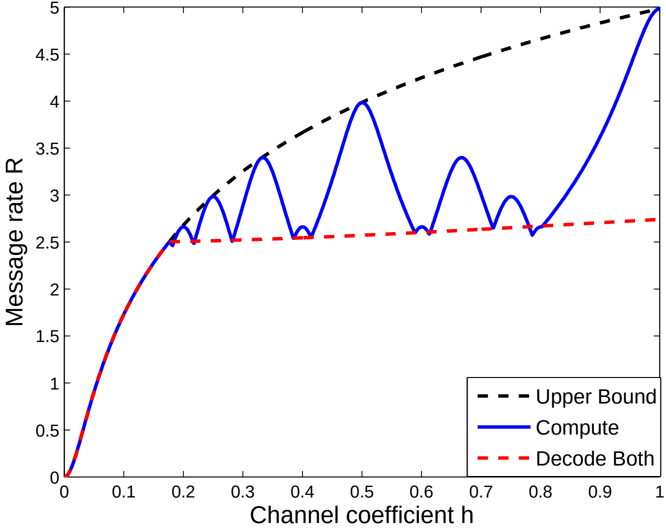
- Receiver observes $\mathbf{y} = \mathbf{x}_1 + h\mathbf{x}_2 + \mathbf{z}$.
- Recovers $a\mathbf{w}_1 \oplus b\mathbf{w}_2$ for $a, b \neq 0$.

20dB



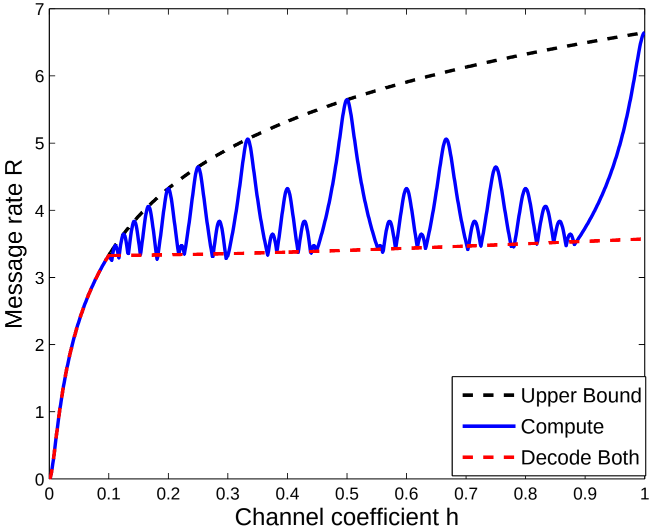
- Receiver observes $\mathbf{y} = \mathbf{x}_1 + h\mathbf{x}_2 + \mathbf{z}$.
- Recovers $a\mathbf{w}_1 \oplus b\mathbf{w}_2$ for $a, b \neq 0$.

30dB



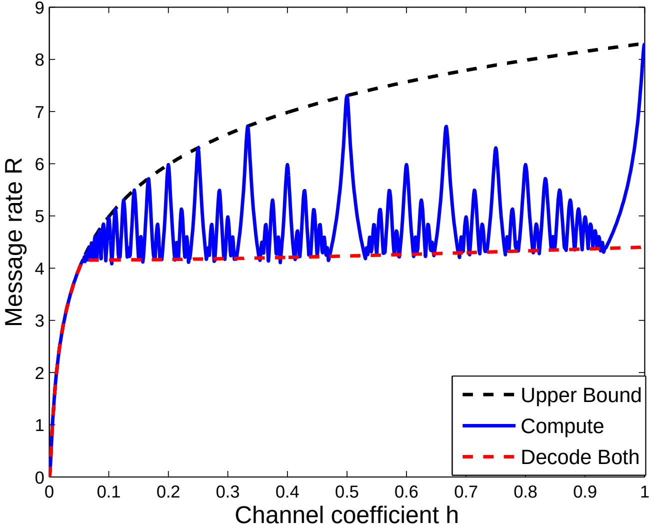
- Receiver observes $\mathbf{y} = \mathbf{x}_1 + h\mathbf{x}_2 + \mathbf{z}$.
- Recovers $a\mathbf{w}_1 \oplus b\mathbf{w}_2$ for $a, b \neq 0$.

40dB



- Receiver observes $\mathbf{y} = \mathbf{x}_1 + h\mathbf{x}_2 + \mathbf{z}$.
- Recovers $a\mathbf{w}_1 \oplus b\mathbf{w}_2$ for $a, b \neq 0$.

50dB



Can we fill in the valleys?

- Compute-and-forward does well at rational coefficients and poorly at irrational coefficients.
- This is the opposite of the behavior observed in "real interference alignment" (**Motahari et al. '09**).
- As demonstrated in the previous talk, we can do better using superposition. This alters the effective channel gains.
- Next talk covers this issue in depth at high SNR.
- How about finite SNR? Similar issues as encountered in static interference channels.

Successive Cancellation

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

Successive cancellation:

- Decode $\mathbf{x}_i$.
- Calculate $\mathbf{y} - h_i \mathbf{x}_i$.
- Receiver now has

$$\sum_{\ell \neq i} h_\ell \mathbf{x}_\ell + \mathbf{z}$$

Successive ~~Cancellation~~ Computation

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

Successive cancellation:

- Decode $\mathbf{x}_i$.
- Calculate $\mathbf{y} - h_i \mathbf{x}_i$.
- Receiver now has

$$\sum_{\ell \neq i} h_\ell \mathbf{x}_\ell + \mathbf{z}$$

Successive computation:

- Decode $\sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell$.
- Calculate $\mathbf{y} + \beta \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell$.
- Receiver now has

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

Successive Computation

- So far, we have only decoded a modulo sum of the lattice points:

$$\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell \Big] \mod \Lambda \ .$$

Successive Computation

- So far, we have only decoded a modulo sum of the lattice points:

$$\Big[\sum_\ell a_\ell \mathbf{t}_\ell\Big] \mod \Lambda \ .$$

- First, add back in the dithers to get the modulo sum of codewords:

$$\bigg[\Big[\sum_\ell a_\ell \mathbf{t}_\ell\Big] \mod \Lambda + \Big[\sum_\ell a_\ell \mathbf{d}_\ell\Big] \mod \Lambda\bigg] \mod \Lambda = \Big[\sum_\ell a_\ell \mathbf{x}_\ell\Big] \mod \Lambda$$

Successive Computation

- So far, we have only decoded a modulo sum of the lattice points:

$$\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell \Big] \mod \Lambda \ .$$

- First, add back in the dithers to get the modulo sum of codewords:

$$\bigg[\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell \Big] \mod \Lambda + \Big[\sum_{\ell} a_\ell \mathbf{d}_\ell \Big] \mod \Lambda \bigg] \mod \Lambda = \Big[\sum_{\ell} a_\ell \mathbf{x}_\ell \Big] \mod \Lambda$$

- Subtract this from $\mathbf{y}$ to expose the coarse lattice point nearest to the real sum:

$$\mathbf{y} - \Big[\sum_{\ell} a_\ell \mathbf{x}_\ell \Big] \mod \Lambda = Q_\Lambda \Big(\sum_{\ell} a_\ell \mathbf{x}_\ell \Big) + \sum_{\ell} (h_\ell - a_\ell) \mathbf{x}_\ell + \mathbf{z}$$

Successive Computation

- So far, we have only decoded a modulo sum of the lattice points:

$$\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell\Big] \mod \Lambda \ .$$

- First, add back in the dithers to get the modulo sum of codewords:

$$\bigg[\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell\Big] \mod \Lambda + \Big[\sum_{\ell} a_\ell \mathbf{d}_\ell\Big] \mod \Lambda\bigg] \mod \Lambda = \Big[\sum_{\ell} a_\ell \mathbf{x}_\ell\Big] \mod \Lambda$$

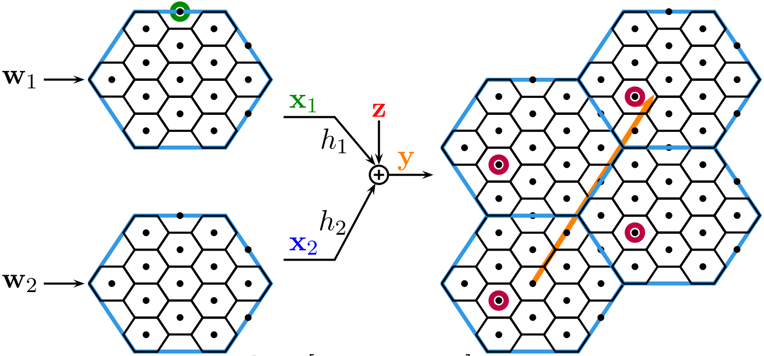
- Subtract this from $\mathbf{y}$ to expose the coarse lattice point nearest to the real sum:

$$\mathbf{y} - \Big[\sum_{\ell} a_\ell \mathbf{x}_\ell\Big] \mod \Lambda = Q_\Lambda\Big(\sum_{\ell} a_\ell \mathbf{x}_\ell\Big) + \sum_{\ell} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}$$

- Coarse lattice point easier to decode than fine lattice point:

$$Q_\Lambda\left(Q_\Lambda\Big(\sum_{\ell} a_\ell \mathbf{x}_\ell\Big) + \sum_{\ell} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}\right) = Q_\Lambda\Big(\sum_{\ell} a_\ell \mathbf{x}_\ell\Big) \quad \text{w.h.p.}$$

We have the modulo sum.

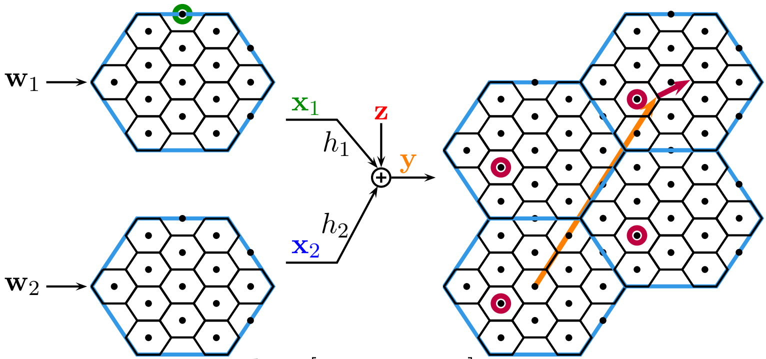


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

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

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

Subtract modulo sum from the received signal.

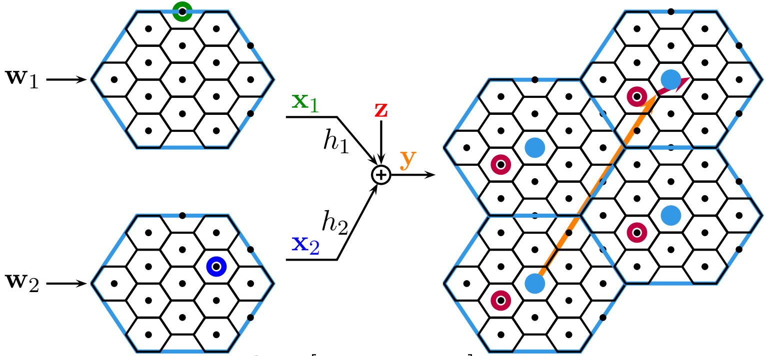


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

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

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

Decode to the closest coarse lattice point.

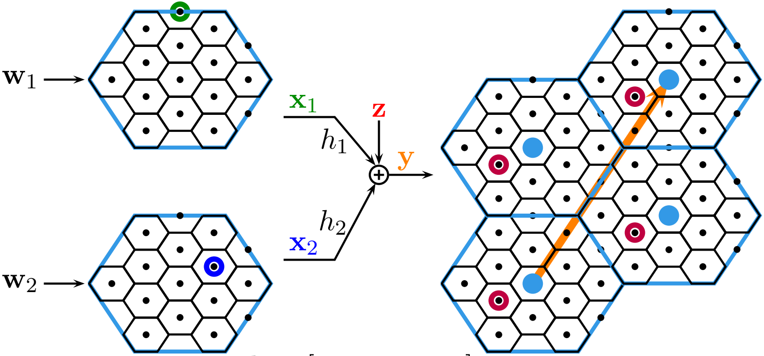


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

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

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

Decode to the closest coarse lattice point.

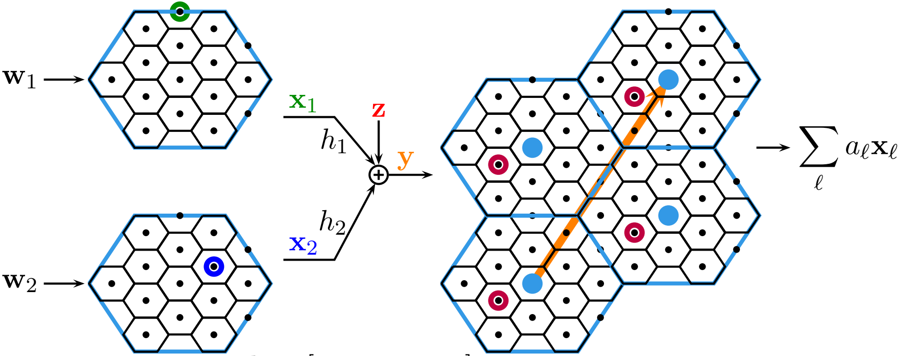


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

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

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

Now we can infer the real sum.



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

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

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

Successive Computation

- Finally, we get back the real sum:

$$\Big[\sum_{\ell} a_\ell \mathbf{x}_\ell\Big] \bmod \Lambda + Q_\Lambda\Big(\sum_{\ell} a_\ell \mathbf{x}_\ell\Big) = \sum_{\ell} a_\ell \mathbf{x}_\ell$$

- What can we do with this?

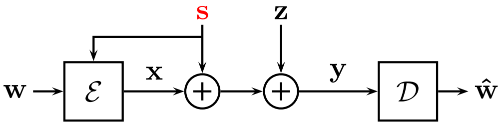
- Change the effective channel gains and decode a new equation.

- Recover some existing results on interference alignment.

Alignment via Successive Computation

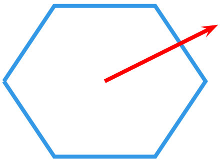
- Assume each receiver observes $\mathbf{y}_k = h\mathbf{x}_k + \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k$.
- Decode equation of the form $p\mathbf{x}_k + \sum_{\ell \neq k} q\mathbf{x}_\ell$.
- Rate $\frac{1}{2}\log\left(\frac{\mathsf{SNR}}{q^2 + \mathsf{SNR}|qh - p|^2}\right) \leq \frac{1}{2}\log\left(\frac{\mathsf{SNR}}{q^2 + \mathsf{SNR}/q^2}\right)$
- Plug in $q \approx \mathsf{SNR}^{1/4}$ to get $R \approx \frac{1}{4}\log(P)$.
- Calculate $\mathbf{y}_k - \frac{p}{q}\mathbf{x}_k + \sum_{\ell \neq k} \mathbf{x}_\ell = \left(h - \frac{p}{q}\right)\mathbf{x}_k + \mathbf{z}_k$
- By Khinchin's Theorem, residual channel coefficient allows $R \approx \frac{1}{4}\log(P)$.

$\mathbf{s}$ is interference known noncausally to the encoder.



Assume $\mathbf{s}$ i.i.d. Gaussian, very large variance P_S.

Erez-Shamai-Zamir '05:
Encoder subtracts $\alpha\mathbf{s}$, dithers, and takes $\bmod \Lambda$.

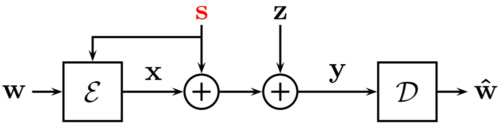


$$\mathbf{x} = [\mathbf{t} - \alpha\mathbf{s} + \mathbf{d}] \mod \Lambda$$

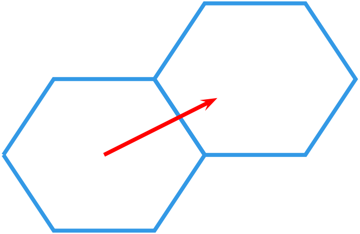
Decoder scales by α, removes dither, takes $\bmod \Lambda$, and recovers $\mathbf{t}$.
Interference is cancelled.

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

$\mathbf{s}$ is interference known noncausally to the encoder.



Assume $\mathbf{s}$ i.i.d. Gaussian, very large variance P_S.



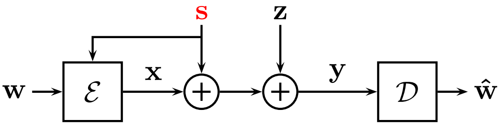
Erez-Shamai-Zamir '05:
Encoder subtracts $\alpha\mathbf{s}$, dithers, and takes $\mod \Lambda$.

$$\mathbf{x} = [\mathbf{t} - \alpha\mathbf{s} + \mathbf{d}] \mod \Lambda$$

Decoder scales by α, removes dither, takes $\mod \Lambda$, and recovers $\mathbf{t}$.
Interference is cancelled.

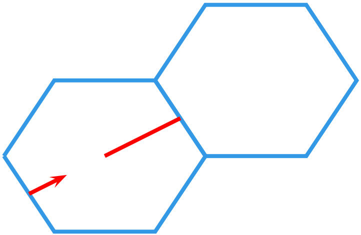
$$\begin{aligned}
[\alpha\mathbf{y} - \mathbf{d}] \bmod \Lambda &= [\mathbf{x} + \alpha\mathbf{s} - \mathbf{d} + \mathbf{z} - (1-\alpha)\mathbf{x}] \bmod \Lambda \\
&= \Big[[\mathbf{t} - \alpha\mathbf{s} + \mathbf{d}] \bmod \Lambda + \alpha\mathbf{s} - \mathbf{d} + \mathbf{z} - (1-\alpha)\mathbf{x}\Big]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

Dirty Paper Coding



$\mathbf{s}$ is interference known noncausally to the encoder.

Assume $\mathbf{s}$ i.i.d. Gaussian, very large variance P_S.



Erez-Shamai-Zamir '05:
Encoder subtracts $\alpha\mathbf{s}$, dithers, and takes $\bmod \Lambda$.

$$\mathbf{x} = [\mathbf{t} - \alpha\mathbf{s} + \mathbf{d}] \mod \Lambda$$

Decoder scales by α, removes dither, takes $\bmod \Lambda$, and recovers $\mathbf{t}$.
Interference is cancelled.

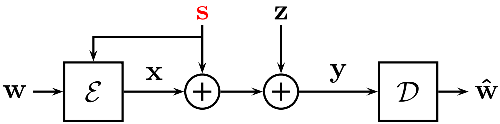
$$\begin{aligned}[\alpha\mathbf{y} - \mathbf{d}] \bmod \Lambda &= [\mathbf{x} + \alpha\mathbf{s} - \mathbf{d} + \mathbf{z} - (1-\alpha)\mathbf{x}] \bmod \Lambda \\ &= \Big[[\mathbf{t} - \alpha\mathbf{s} + \mathbf{d}] \bmod \Lambda + \alpha\mathbf{s} - \mathbf{d} + \mathbf{z} - (1-\alpha)\mathbf{x}\Big]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

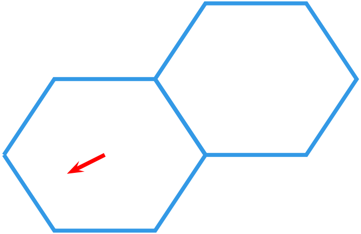
$\mathbf{s}$ is interference known noncausally to the encoder.

Assume $\mathbf{s}$ i.i.d. Gaussian, very large variance P_S.

Erez-Shamai-Zamir '05:
Encoder subtracts $\alpha\mathbf{s}$, dithers, and takes $\bmod \Lambda$.

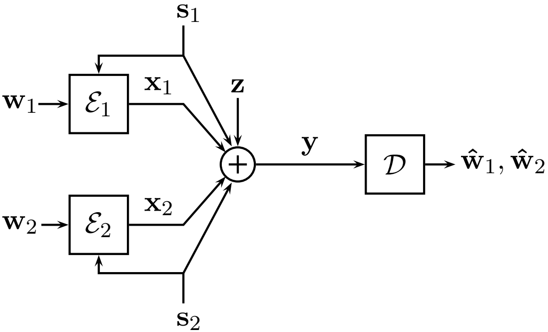
$$\mathbf{x} = [\mathbf{t} - \alpha\mathbf{s} + \mathbf{d}] \mod \Lambda$$





Decoder scales by α, removes dither, takes $\bmod \Lambda$, and recovers $\mathbf{t}$.
Interference is cancelled.

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



Philosof-Zamir-Erez-Khisti '11:

- Encoder 1 knows interference $\mathbf{s}_1$.
- Encoder 2 knows interference $\mathbf{s}_2$.
- Need to cancel out interference in a distributed fashion.
- Assume i.i.d. Gaussian interference with very large variance P_S. Random i.i.d. methods yield rate that goes to 0 as P_S goes to infinity.

Dirty Gaussian Multiple-Access Channel

Subtract (part of) the interference signals ahead of time:

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

Decoder removes dithers:

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

Select $\alpha = 2P/(2P+N)$ to obtain

$$R_1 + R_2 \leq \frac{1}{2} \log \left(\frac{1}{2} + \frac{P}{N} \right)$$

Maple Syrup Problem: Prove this is the best possible

.

Dirty Gaussian Multiple-Access Channel

Subtract (part of) the interference signals ahead of time:

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

Decoder removes dithers:

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

Select $\alpha = 2P/(2P+N)$ to obtain

$$R_1 + R_2 \leq \frac{1}{2} \log \left(\frac{1}{2} + \frac{P}{N} \right)$$

Maple Syrup Problem: Prove this is the best possible (or find a better acheivable scheme).

AWGN Techniques

- Superposition (**Shlomo's Talk**)
- Fading
- Successive Cancellation
- Dirty Paper Coding
- Joint Decoding (**Uri's Talk**)
- List Decoding (**Natasha's Talk**)

AWGN Techniques

- Superposition (**Shlomo's Talk**)
- Fading
- Successive Cancellation
- Dirty Paper Coding
- Joint Decoding (**Uri's Talk**)
- List Decoding (**Natasha's Talk**)
- Timbit Problem: Are there are any AWGN encoding/decoding techniques that are not available to lattice codes? That is, is there a (simple, linear, AWGN, etc.) network where lattices are outperformed by i.i.d. random codes?

AWGN Techniques

- Superposition (**Shlomo's Talk**)
- Fading
- Successive Cancellation
- Dirty Paper Coding
- Joint Decoding (**Uri's Talk**)
- List Decoding (**Natasha's Talk**)
- Timbit Problem: Are there are any AWGN encoding/decoding techniques that are not available to lattice codes? That is, is there a (simple, linear, AWGN, etc.) network where lattices are outperformed by i.i.d. random codes?
- Outer bounds?