Computation Alignment:

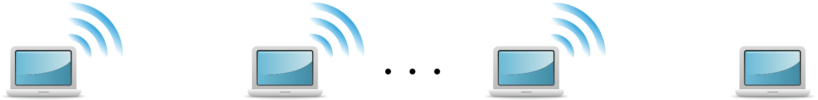
Capacity Approximation without Noise Accumulation

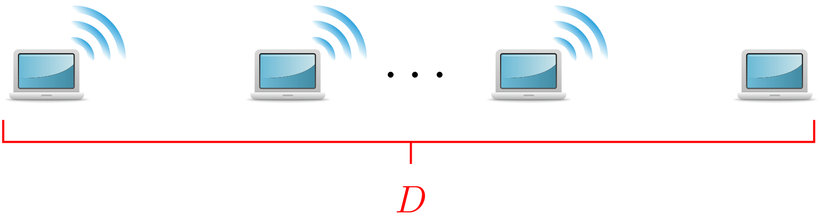
Urs Niesen
Bell Labs

Bobak Nazer
Boston University

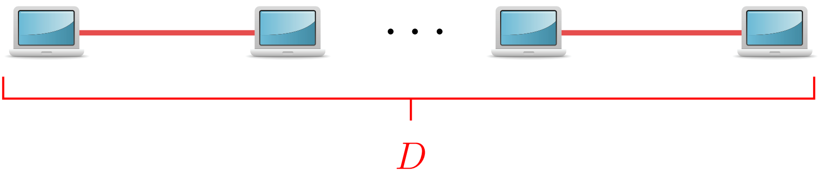
Phil Whiting
Bell Labs

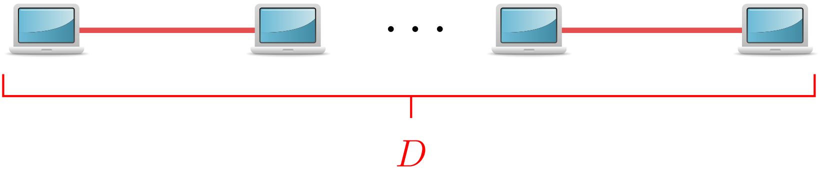
ISWCS 2011
November 7, 2011



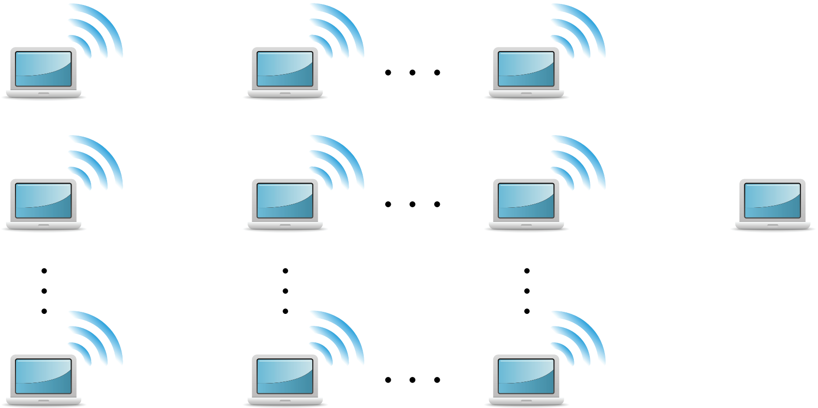


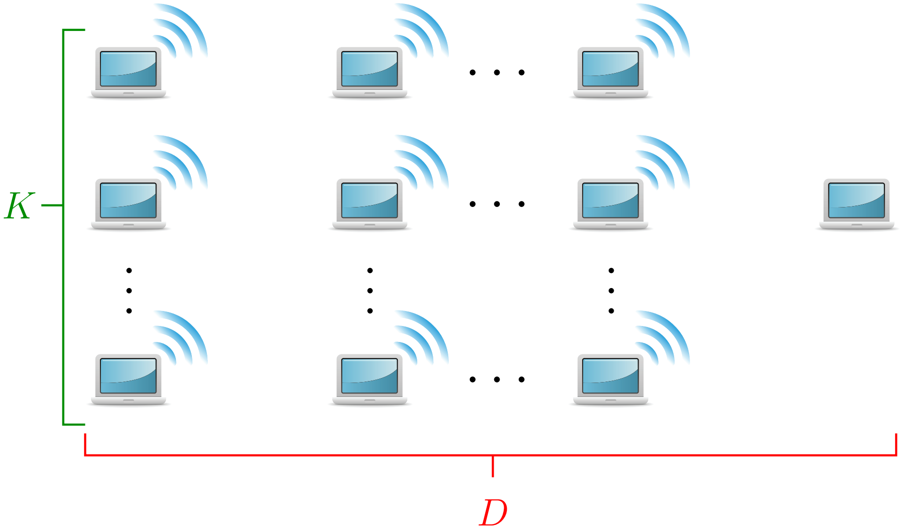
. . .
D


D

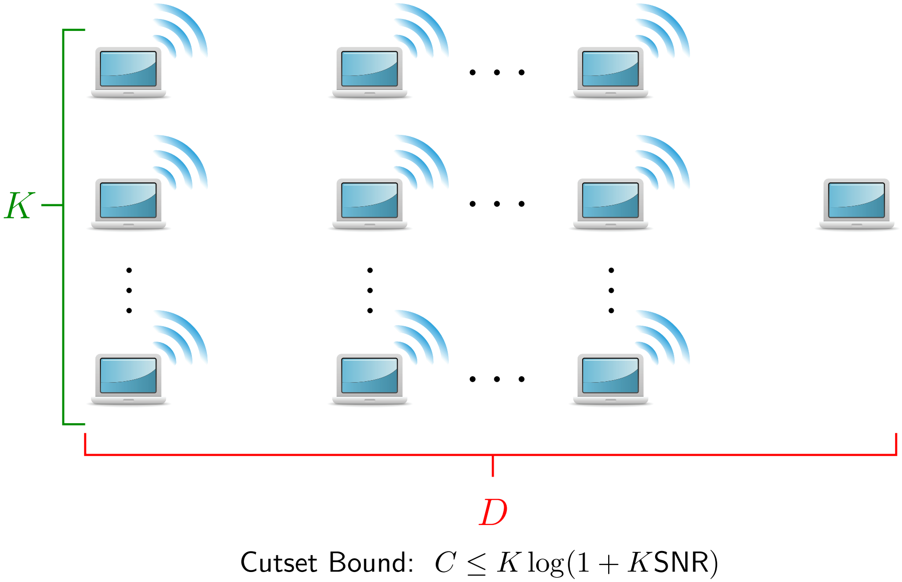


Capacity: $C = \log(1 + \mathsf{SNR})$





K
D


K
D
Cutset Bound: $C \leq K \log(1 + K\mathsf{SNR})$

K = Network Width

D = Network Depth

- **Decode-and-Forward** (**Cover-El Gamal '79**): Decode subsets of messages. Removes noise but interference-limited.

$$C \geq \log(1 + K\mathsf{SNR})$$

K = Network Width D = Network Depth

- **Decode-and-Forward** (**Cover-El Gamal '79**): Decode subsets of messages. Removes noise but interference-limited.

$$C \geq \log(1 + K\mathsf{SNR})$$

- **Compress-and-Forward** (**Cover-El Gamal '79**): Quantize analog observations. Captures signal interactions but accumulates noise.

Avestimehr-Diggavi-Tse '11: $C \geq K \log(1 + K\mathsf{SNR}) - KD$

K = Network Width D = Network Depth

- **Decode-and-Forward** (**Cover-El Gamal '79**): Decode subsets of messages. Removes noise but interference-limited.

$$C \geq \log(1 + K\mathsf{SNR})$$

- **Compress-and-Forward** (**Cover-El Gamal '79**): Quantize analog observations. Captures signal interactions but accumulates noise.

Avestimehr-Diggavi-Tse '11: $C \geq K \log(1 + K\mathsf{SNR}) - KD$

- **Compute-and-Forward** (**Nazer-Gastpar '11**): Decode equations of messages. Partially captures signal interactions and removes noise.

What rates are attainable?

Compute-and-Forward to a Single Relay

Finite field messages:

$$\mathbf{w}_\ell \in \mathbb{F}_q^m$$

Power-constrained inputs:

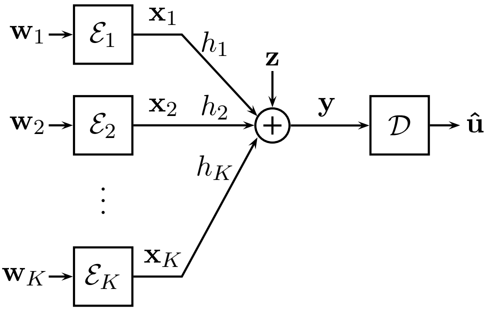
$$\frac{1}{n}\|\mathbf{x}_\ell\|^2 \leq \mathsf{SNR}$$

AWGN noise:

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Message rate:

$$R_\ell = \frac{m}{n}\log_2 q$$

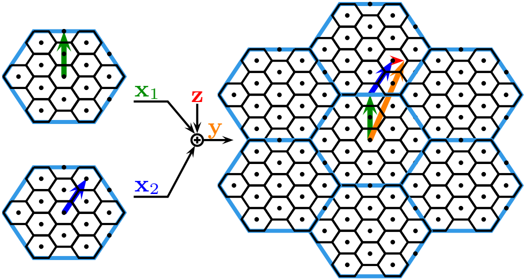


- Decoder makes an estimate $\hat{\mathbf{u}}$ of a linear equation $\mathbf{u} = \bigoplus_{\ell=1}^{K} a_\ell \mathbf{w}_\ell$
- Require that probability of error vanishes with blocklength.
- Receiver can use its knowledge of the channel gains to match the integer equation coefficients a_ℓ to the channel coefficients h_ℓ.

Compute-and-Forward – Single Receiver

Lattice codes: Sums of codewords are codewords.

Erez-Zamir '04: Nested lattice codes can approach AWGN capacity.

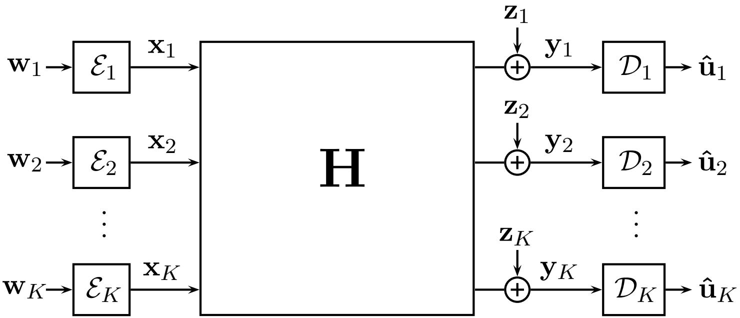


Theorem (Nazer-Gastpar IT Trans. '11)

A relay can reliably decode the linear equation with coefficients $\mathbf{a} \in \mathbb{Z}^K$ *from a channel with coefficients* $\mathbf{h} \in \mathbb{R}^K$ *if the message rates satisfy:*

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

For complex-valued channels, use real-valued decomposition.



Theorem (Nazer-Gastpar IT Trans. '11)

The relays can decode a full rank set of linear equations from a channel matrix with rows $\mathbf{h}_k^T \in \mathbb{C}^K$ *as long as for some full rank matrix with columns* $\mathbf{a}_k \in \{\mathbb{Z} + j\mathbb{Z}\}^K$ *the message rates satisfy:*

$$R_\ell < \min_{k:a_{k\ell}\neq 0} \log\left(\frac{1 + \mathsf{SNR}\|\mathbf{h}_k\|^2}{\|\mathbf{a}_k\|^2 + \mathsf{SNR}(\|\mathbf{h}_k\|^2\|\mathbf{a}_k\|^2 - (\mathbf{h}_k^*\mathbf{a}_k)^2)}\right)$$

Compute-and-Forward – Degrees-of-Freedom

- Compute-and-forward performs well in simulations at moderate SNR (e.g. 15dB). Does it scale like $K \log \text{SNR}$?

Compute-and-Forward – Degrees-of-Freedom

- Compute-and-forward performs well in simulations at moderate SNR (e.g. 15dB). Does it scale like $K \log \text{SNR}$?

Theorem (Niesen-Whiting ISIT '11, arXiv '11)

Compute-and-forward scheme from **Nazer-Gastpar '11** *has at most* 2 *degrees-of-freedom.*

Compute-and-Forward – Degrees-of-Freedom

- Compute-and-forward performs well in simulations at moderate SNR (e.g. 15dB). Does it scale like $K \log \mathsf{SNR}$?

Theorem (Niesen-Whiting ISIT '11, arXiv '11)

Compute-and-forward scheme from **Nazer-Gastpar '11** *has at most 2 degrees-of-freedom.*

- Is that the end of the story?

Compute-and-Forward – Degrees-of-Freedom

- Compute-and-forward performs well in simulations at moderate SNR (e.g. 15dB). Does it scale like $K \log \mathsf{SNR}$?

Theorem (Niesen-Whiting ISIT '11, arXiv '11)

Compute-and-forward scheme from **Nazer-Gastpar '11** *has at most* 2 *degrees-of-freedom.*

- Is that the end of the story?

Theorem (Niesen-Whiting ISIT '11, arXiv '11)

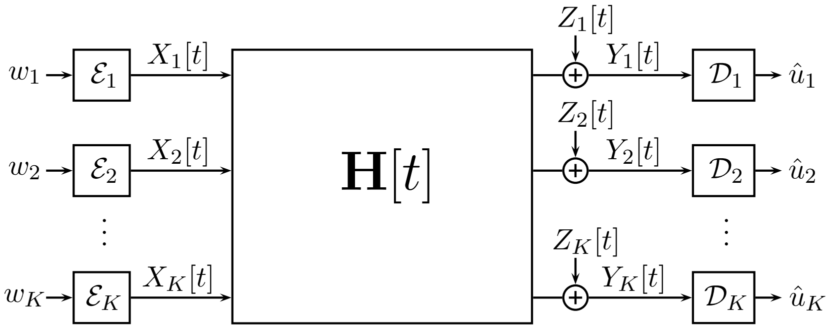
Coupling compute-and-forward with interference alignment can achieve K *degrees-of-freedom.*

- Key Idea: Use alignment scheme from **Motahari et al. '09** to create effective integer channels that are good for compute-and-forward.

Finite SNR Bounds

- In the high SNR limit, noise accumulation does not show up: we could just as well use compress-and-forward.
- **Goal for this talk:** Design a computation alignment scheme that can send equations with rates within a constant gap of $K \log \mathsf{SNR}$.
- Ergodic interference alignment schemes offer excellent finite SNR rates for the standard interference channel problem.
- Requires time-varying channel coefficients. Our constant gap will depend on the number of sources K and the fading statistics but not the network depth D.

Computation Alignment - Setup



- Relays recover an invertible set of functions

$$u_k = f_k(w_1, w_2, \ldots, w_K)$$

- Vanishing probability of error.
- Channel coefficients are independent from each other, stationary and ergodic across time, and have uniform phase.

Starting Point – Ergodic Interference Alignment

Basic ergodic alignment scheme:

- Assume each relay wants a single message, $u_k = w_k$.
- **Nazer-Gastpar-Jafar-Vishwanath '09:** The following rate is achievable for each user:

$$R_k = \frac{1}{2}\mathbb{E}\left[\log(1 + 2|h_{kk}|^2 \mathsf{SNR})\right]$$

- Corresponds to a sum rate that scales like $\dfrac{K}{2}\log \mathsf{SNR}$.

Starting Point – Ergodic Interference Alignment

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

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

Starting Point – Ergodic Interference Alignment

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

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

Starting Point – Ergodic Interference Alignment

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

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

Starting Point – Ergodic Interference Alignment

1. At time t with channel $\mathbf{H}$, user k transmits symbol 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 symbol 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 symbols and wait for their $\mathbf{H}_C$.

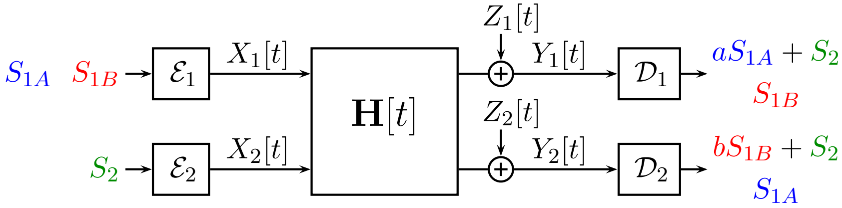
Starting Point – Ergodic Interference Alignment

- For long block lengths, nearly every matrix finds its quantized match.
- In general, we can successfully pair together matrices if their coefficients are phase rotations of each other,

$$h_{k\ell}[t_C] = e^{j\phi_{k\ell}} h_{k\ell}[t] \ .$$

- Computation alignment: Pair up coefficients to create effective integer channels.

Computation Alignment - Motivating Example

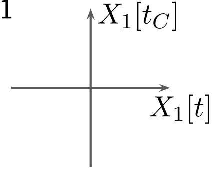


- Send 3 symbols over 2 channel uses. Sum rate is:

$$R_{\textsf{sum}} = \frac{3}{2}\mathbb{E}\left[\min_{k,\ell} \log(1 + |h_{k\ell}|^2\textsf{SNR})\right]$$

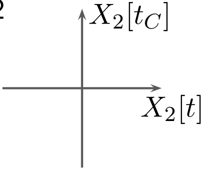
- Each relay gets 2 equations. Coefficients a and b are non-zero integers so equations are always invertible.
- Use compute-and-forward to code over resulting effective channels.

Tx 1



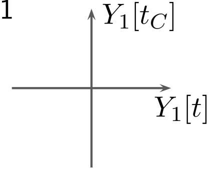
$$X_1[t] =$$
$$X_1[t_C] =$$

Tx 2



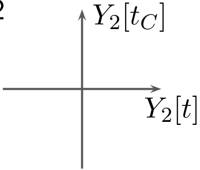
$$X_2[t] =$$
$$X_2[t_C] =$$

Rx 1



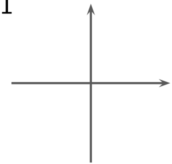
$$Y_1[t] + Y_1[t_C] =$$
$$Y_1[t] - Y_1[t_C] =$$

Rx 2



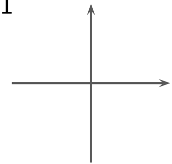
$$Y_2[t] + Y_2[t_C] =$$
$$Y_2[t] - Y_2[t_C] =$$

Tx 1



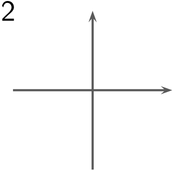
$$X_1[t] =$$
$$X_1[t_C] =$$

Rx 1



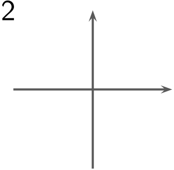
$$Y_1[t] + Y_1[t_C] =$$
$$Y_1[t] - Y_1[t_C] =$$

Tx 2



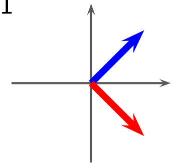
$$X_2[t] =$$
$$X_2[t_C] =$$

Rx 2



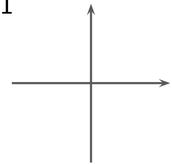
$$Y_2[t] + Y_2[t_C] =$$
$$Y_2[t] - Y_2[t_C] =$$

Tx 1



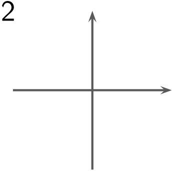
$$X_1[t] = S_{1A} + S_{1B}$$
$$X_1[t_C] = S_{1A} - S_{1B}$$

Rx 1



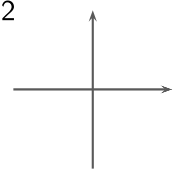
$$Y_1[t] + Y_1[t_C] =$$
$$Y_1[t] - Y_1[t_C] =$$

Tx 2



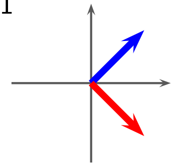
$$X_2[t] =$$
$$X_2[t_C] =$$

Rx 2



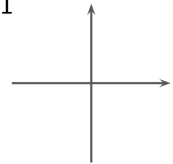
$$Y_2[t] + Y_2[t_C] =$$
$$Y_2[t] - Y_2[t_C] =$$

Tx 1



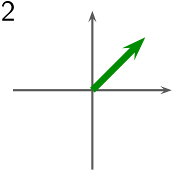
$$X_1[t] = S_{1A} + S_{1B}$$
$$X_1[t_C] = S_{1A} - S_{1B}$$

Rx 1



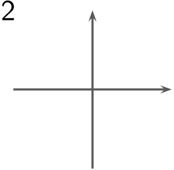
$$Y_1[t] + Y_1[t_C] =$$
$$Y_1[t] - Y_1[t_C] =$$

Tx 2

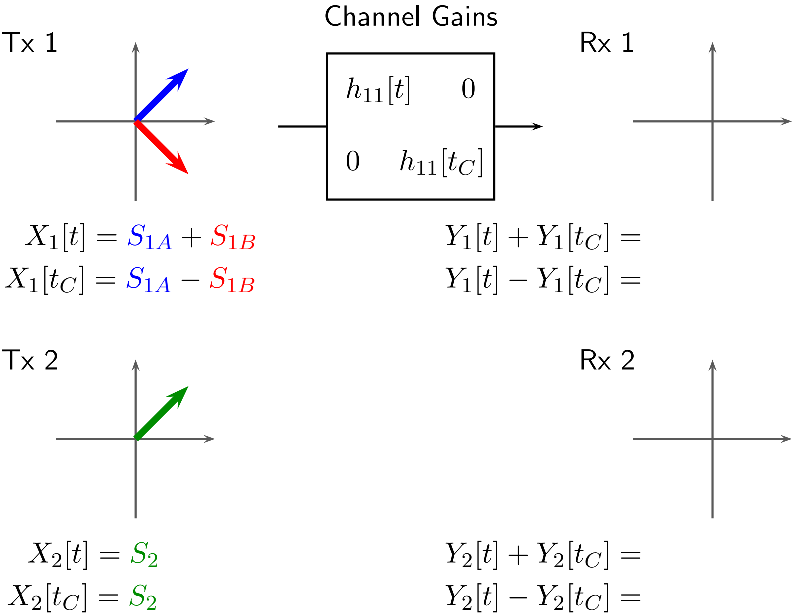


$$X_2[t] = S_2$$
$$X_2[t_C] = S_2$$

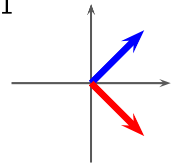
Rx 2

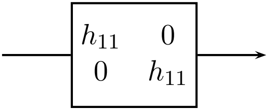


$$Y_2[t] + Y_2[t_C] =$$
$$Y_2[t] - Y_2[t_C] =$$

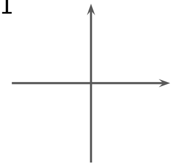

Channel Gains
Tx 1
$h_{11}[t]$ 0
0 $h_{11}[t_C]$
Rx 1
$X_1[t] = S_{1A} + S_{1B}$
$X_1[t_C] = S_{1A} - S_{1B}$
$Y_1[t] + Y_1[t_C] =$
$Y_1[t] - Y_1[t_C] =$
Tx 2
Rx 2
$X_2[t] = S_2$
$X_2[t_C] = S_2$
$Y_2[t] + Y_2[t_C] =$
$Y_2[t] - Y_2[t_C] =$

Tx 1





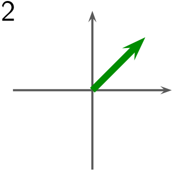
Rx 1



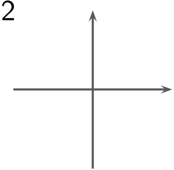
$$X_1[t] = S_{1A} + S_{1B}$$
$$X_1[t_C] = S_{1A} - S_{1B}$$

$$Y_1[t] + Y_1[t_C] =$$
$$Y_1[t] - Y_1[t_C] =$$

Tx 2

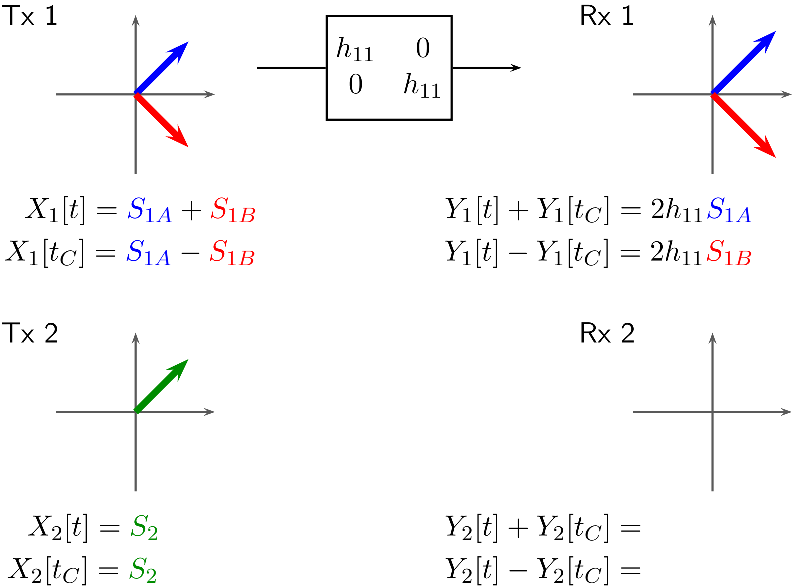


Rx 2

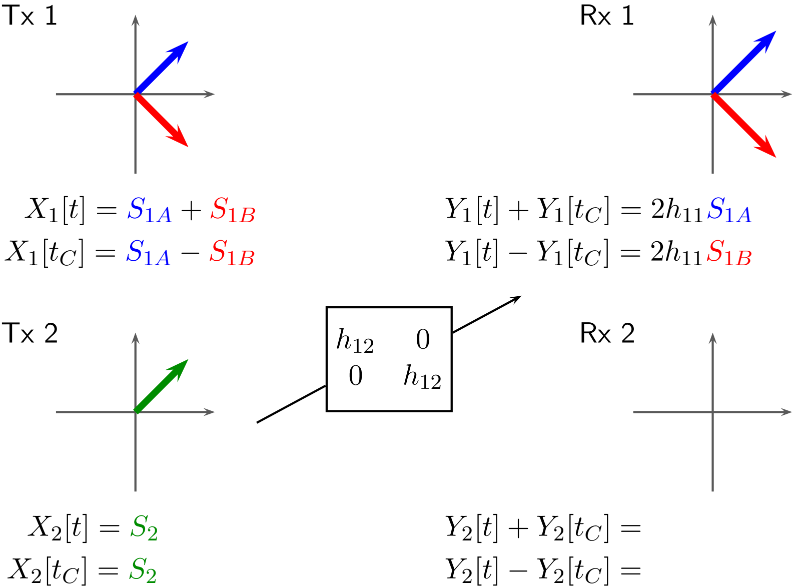


$$X_2[t] = S_2$$
$$X_2[t_C] = S_2$$

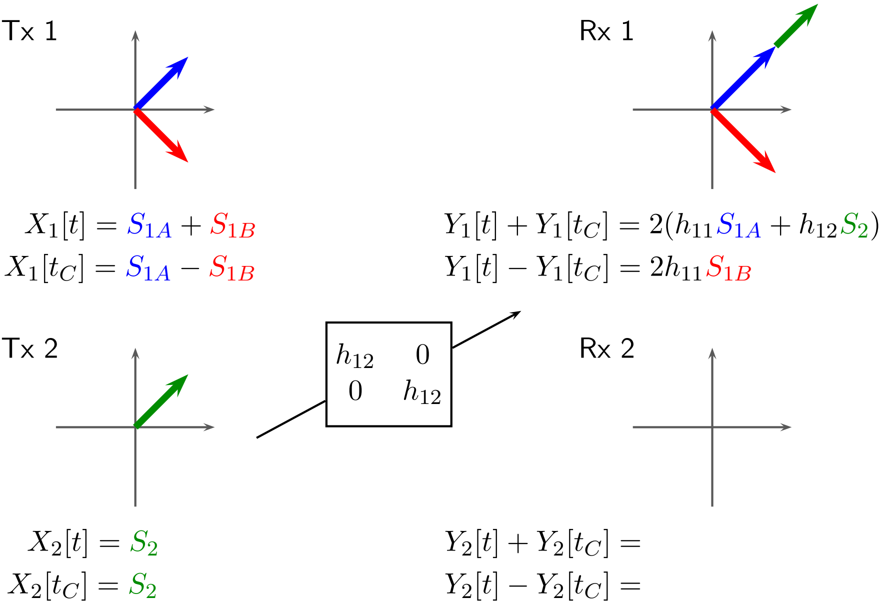
$$Y_2[t] + Y_2[t_C] =$$
$$Y_2[t] - Y_2[t_C] =$$



Tx 1
h_{11} 0
0 h_{11}
Rx 1
$X_1[t] = S_{1A} + S_{1B}$
$X_1[t_C] = S_{1A} - S_{1B}$
$Y_1[t] + Y_1[t_C] = 2h_{11}S_{1A}$
$Y_1[t] - Y_1[t_C] = 2h_{11}S_{1B}$
Tx 2
Rx 2
$X_2[t] = S_2$
$X_2[t_C] = S_2$
$Y_2[t] + Y_2[t_C] =$
$Y_2[t] - Y_2[t_C] =$

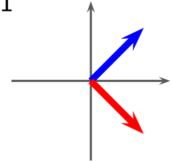


Tx 1
$X_1[t] = S_{1A} + S_{1B}$
$X_1[t_C] = S_{1A} - S_{1B}$
Rx 1
$Y_1[t] + Y_1[t_C] = 2h_{11}S_{1A}$
$Y_1[t] - Y_1[t_C] = 2h_{11}S_{1B}$
Tx 2
$\begin{matrix} h_{12} & 0 \\ 0 & h_{12} \end{matrix}$
Rx 2
$X_2[t] = S_2$
$X_2[t_C] = S_2$
$Y_2[t] + Y_2[t_C] =$
$Y_2[t] - Y_2[t_C] =$



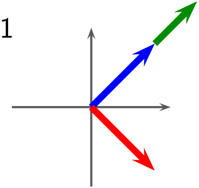
Tx 1
$X_1[t] = S_{1A} + S_{1B}$
$X_1[t_C] = S_{1A} - S_{1B}$
Rx 1
$Y_1[t] + Y_1[t_C] = 2(h_{11}S_{1A} + h_{12}S_2)$
$Y_1[t] - Y_1[t_C] = 2h_{11}S_{1B}$
Tx 2
h_{12} 0
0 h_{12}
Rx 2
$X_2[t] = S_2$
$X_2[t_C] = S_2$
$Y_2[t] + Y_2[t_C] =$
$Y_2[t] - Y_2[t_C] =$

Tx 1



Adjust strength of S_{1A} to get integer coefficients.

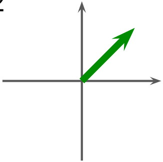
Rx 1



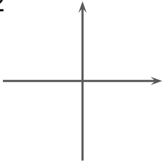
$$X_1[t] = S_{1A} + S_{1B}$$
$$X_1[t_C] = S_{1A} - S_{1B}$$

$$Y_1[t] + Y_1[t_C] = 2(h_{11}S_{1A} + h_{12}S_2)$$
$$Y_1[t] - Y_1[t_C] = 2h_{11}S_{1B}$$

Tx 2



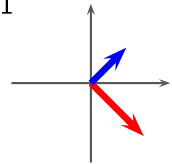
Rx 2



$$X_2[t] = S_2$$
$$X_2[t_C] = S_2$$

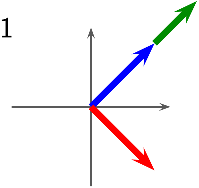
$$Y_2[t] + Y_2[t_C] =$$
$$Y_2[t] - Y_2[t_C] =$$

Tx 1



Adjust strength of S_{1A} to get integer coefficients.

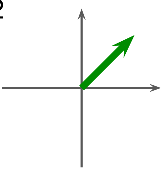
Rx 1



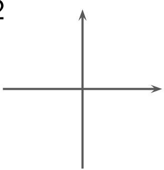
$$X_1[t] = \rho S_{1A} + S_{1B}$$
$$X_1[t_C] = \rho S_{1A} - S_{1B}$$

$$Y_1[t] + Y_1[t_C] = 2(h_{11} S_{1A} + h_{12} S_2)$$
$$Y_1[t] - Y_1[t_C] = 2h_{11} S_{1B}$$

Tx 2



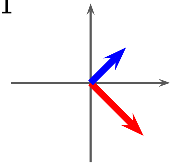
Rx 2



$$X_2[t] = S_2$$
$$X_2[t_C] = S_2$$

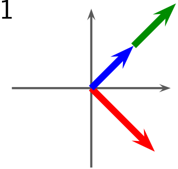
$$Y_2[t] + Y_2[t_C] =$$
$$Y_2[t] - Y_2[t_C] =$$

Tx 1



Adjust strength of S_{1A}
to get integer coefficients.

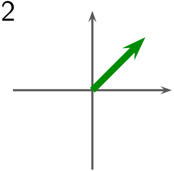
Rx 1



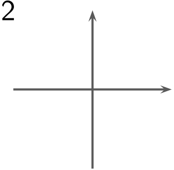
$$X_1[t] = \rho S_{1A} + S_{1B}$$
$$X_1[t_C] = \rho S_{1A} - S_{1B}$$

$$Y_1[t] + Y_1[t_C] = 2h_{12}(aS_{1A} + S_2)$$
$$Y_1[t] - Y_1[t_C] = 2h_{11}S_{1B}$$

Tx 2



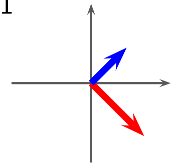
Rx 2



$$X_2[t] = S_2$$
$$X_2[t_C] = S_2$$

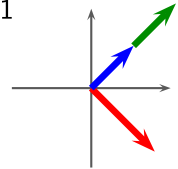
$$Y_2[t] + Y_2[t_C] =$$
$$Y_2[t] - Y_2[t_C] =$$

Tx 1



$$X_1[t] = \rho S_{1A} + S_{1B}$$
$$X_1[t_C] = \rho S_{1A} - S_{1B}$$

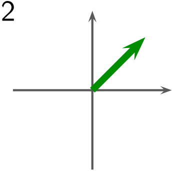
Rx 1



$$Y_1[t] + Y_1[t_C] = 2h_{12}(aS_{1A} + S_2)$$
$$Y_1[t] - Y_1[t_C] = 2h_{11}S_{1B}$$

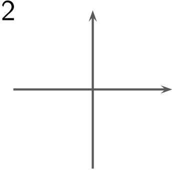
$$\begin{matrix} h_{21} & 0 \\ 0 & -h_{21} \end{matrix}$$

Tx 2

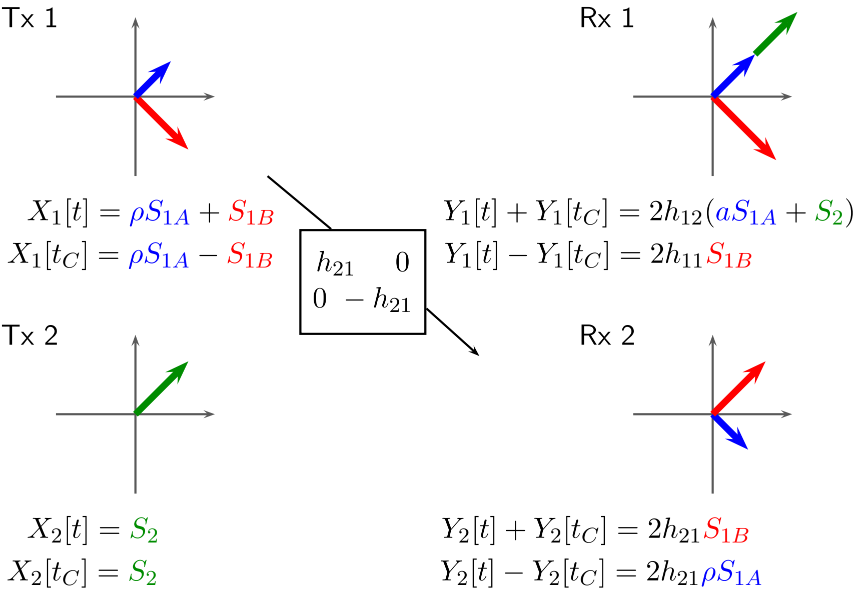


$$X_2[t] = S_2$$
$$X_2[t_C] = S_2$$

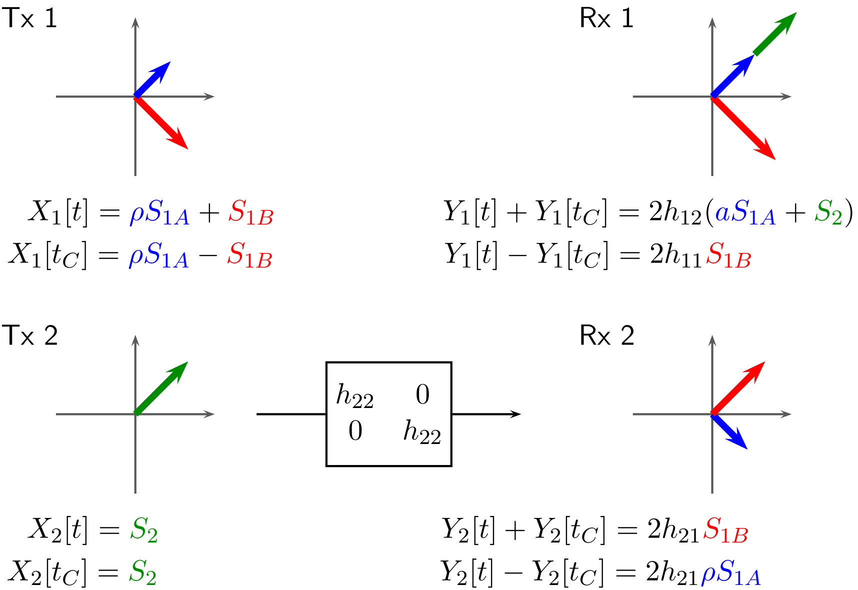
Rx 2

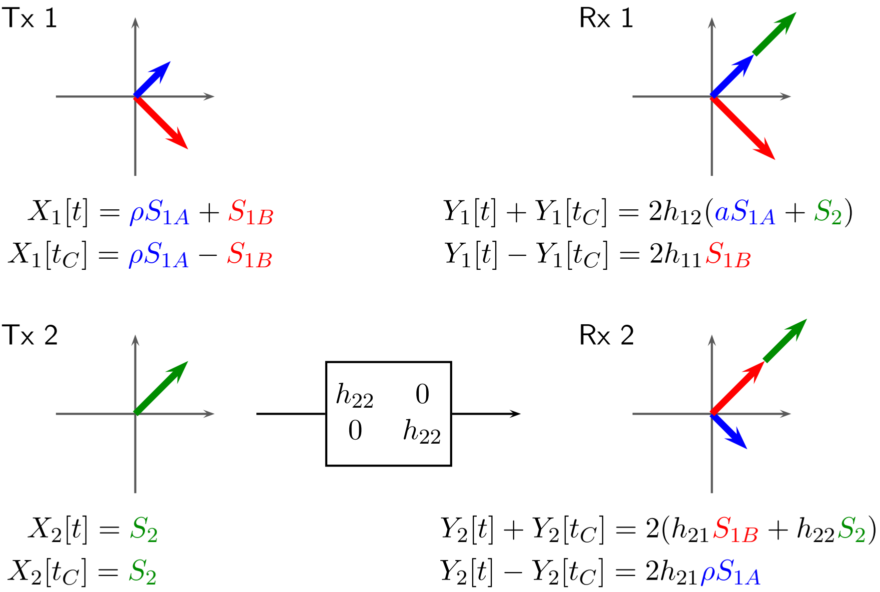


$$Y_2[t] + Y_2[t_C] =$$
$$Y_2[t] - Y_2[t_C] =$$



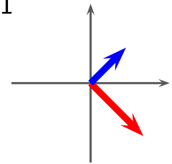
Tx 1
$X_1[t] = \rho S_{1A} + S_{1B}$
$X_1[t_C] = \rho S_{1A} - S_{1B}$
Tx 2
$X_2[t] = S_2$
$X_2[t_C] = S_2$
$\begin{matrix} h_{21} & 0 \\ 0 & -h_{21} \end{matrix}$
Rx 1
$Y_1[t] + Y_1[t_C] = 2h_{12}(aS_{1A} + S_2)$
$Y_1[t] - Y_1[t_C] = 2h_{11}S_{1B}$
Rx 2
$Y_2[t] + Y_2[t_C] = 2h_{21}S_{1B}$
$Y_2[t] - Y_2[t_C] = 2h_{21}\rho S_{1A}$


Tx 1
Rx 1
$X_1[t] = \rho S_{1A} + S_{1B}$
$X_1[t_C] = \rho S_{1A} - S_{1B}$
$Y_1[t] + Y_1[t_C] = 2h_{12}(aS_{1A} + S_2)$
$Y_1[t] - Y_1[t_C] = 2h_{11} S_{1B}$
Tx 2
h_{22} 0
0 h_{22}
Rx 2
$X_2[t] = S_2$
$X_2[t_C] = S_2$
$Y_2[t] + Y_2[t_C] = 2h_{21} S_{1B}$
$Y_2[t] - Y_2[t_C] = 2h_{21} \rho S_{1A}$



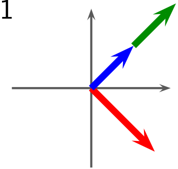
Tx 1
Rx 1
$X_1[t] = \rho S_{1A} + S_{1B}$
$X_1[t_C] = \rho S_{1A} - S_{1B}$
$Y_1[t] + Y_1[t_C] = 2h_{12}(aS_{1A} + S_2)$
$Y_1[t] - Y_1[t_C] = 2h_{11}S_{1B}$
Tx 2
h_{22} 0
0 h_{22}
Rx 2
$X_2[t] = S_2$
$X_2[t_C] = S_2$
$Y_2[t] + Y_2[t_C] = 2(h_{21}S_{1B} + h_{22}S_2)$
$Y_2[t] - Y_2[t_C] = 2h_{21}\rho S_{1A}$

Tx 1



Adjust strength of S_{1B}
to get integer coefficients.

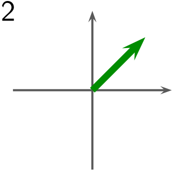
Rx 1



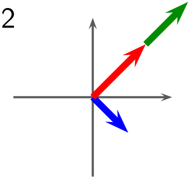
$$X_1[t] = \rho S_{1A} + S_{1B}$$
$$X_1[t_C] = \rho S_{1A} - S_{1B}$$

$$Y_1[t] + Y_1[t_C] = 2h_{12}(aS_{1A} + S_2)$$
$$Y_1[t] - Y_1[t_C] = 2h_{11}S_{1B}$$

Tx 2



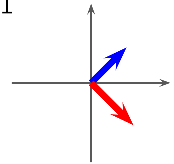
Rx 2



$$X_2[t] = S_2$$
$$X_2[t_C] = S_2$$

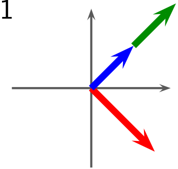
$$Y_2[t] + Y_2[t_C] = 2(h_{21}S_{1B} + h_{22}S_2)$$
$$Y_2[t] - Y_2[t_C] = 2h_{21}\rho S_{1A}$$

Tx 1



Adjust strength of S_{1B} to get integer coefficients.

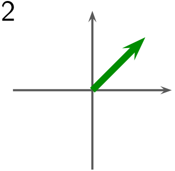
Rx 1



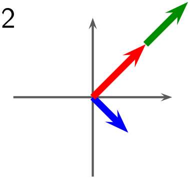
$$X_1[t] = \rho S_{1A} + \gamma S_{1B}$$
$$X_1[t_C] = \rho S_{1A} - \gamma S_{1B}$$

$$Y_1[t] + Y_1[t_C] = 2h_{12}(aS_{1A} + S_2)$$
$$Y_1[t] - Y_1[t_C] = 2h_{11}S_{1B}$$

Tx 2



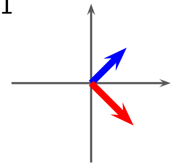
Rx 2



$$X_2[t] = S_2$$
$$X_2[t_C] = S_2$$

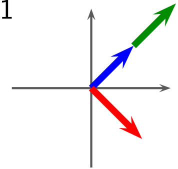
$$Y_2[t] + Y_2[t_C] = 2(h_{21}S_{1B} + h_{22}S_2)$$
$$Y_2[t] - Y_2[t_C] = 2h_{21}\rho S_{1A}$$

Tx 1



Adjust strength of S_{1B} to get integer coefficients.

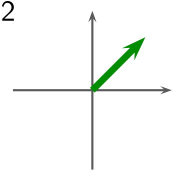
Rx 1



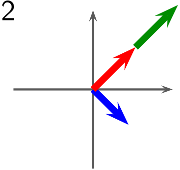
$$X_1[t] = \rho S_{1A} + \gamma S_{1B}$$
$$X_1[t_C] = \rho S_{1A} - \gamma S_{1B}$$

$$Y_1[t] + Y_1[t_C] = 2h_{12}(aS_{1A} + S_2)$$
$$Y_1[t] - Y_1[t_C] = 2h_{11}\gamma S_{1B}$$

Tx 2



Rx 2



$$X_2[t] = S_2$$
$$X_2[t_C] = S_2$$

$$Y_2[t] + Y_2[t_C] = 2h_{22}(bS_{1B} + S_2)$$
$$Y_2[t] - Y_2[t_C] = 2h_{21}\rho S_{1A}$$

Computation Alignment - Main Result

$K = 2$ case:

- User 1 sends L symbols using L DFT vectors.
- User 2 sends $L - 1$ symbols using $L - 1$ DFT vectors.
- Relays get $2L$ equations. Sum rate is approximately $\frac{2L-1}{L} \log \mathsf{SNR}$.

Computation Alignment - Main Result

$K = 2$ case:

- User 1 sends L symbols using L DFT vectors.
- User 2 sends $L-1$ symbols using $L-1$ DFT vectors.
- Relays get $2L$ equations. Sum rate is approximately $\frac{2L-1}{L} \log \mathsf{SNR}$.

$K \geq 2$ case: More sophisticated set of alignment vectors.

Theorem (Niesen-Nazer-Whiting arXiv '11)

For Rayleigh fading, $h_{k\ell}[t] \sim \mathcal{CN}(0,1)$, the following sum rate is achievable using computation alignment:

$$R_{sum} = K \log \mathsf{SNR} - 7K^3 \ .$$

Computation Alignment - Main Result

$K = 2$ case:

- User 1 sends L symbols using L DFT vectors.
- User 2 sends $L-1$ symbols using $L-1$ DFT vectors.
- Relays get $2L$ equations. Sum rate is approximately $\frac{2L-1}{L} \log \mathsf{SNR}$.

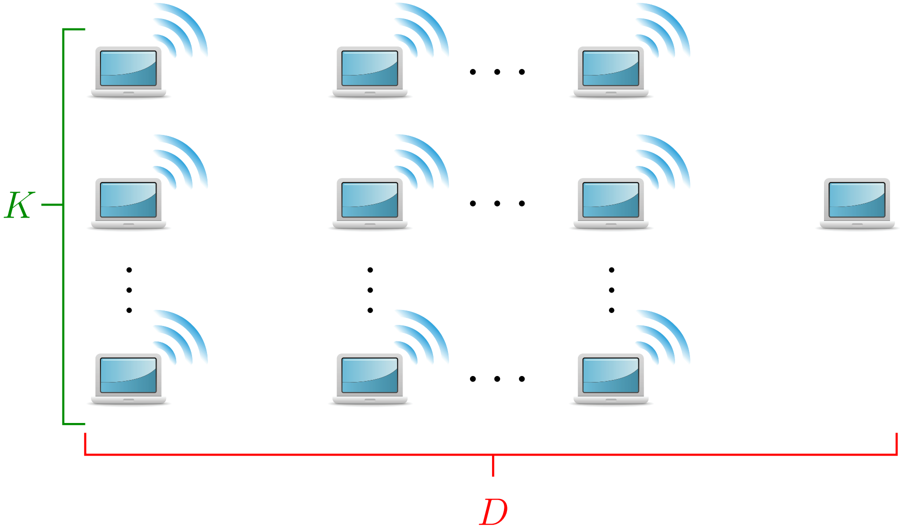
$K \geq 2$ case: More sophisticated set of alignment vectors.

Theorem (Niesen-Nazer-Whiting arXiv '11)

For Rayleigh fading, $h_{k\ell}[t] \sim \mathcal{CN}(0,1)$, the following sum rate is achievable using computation alignment:

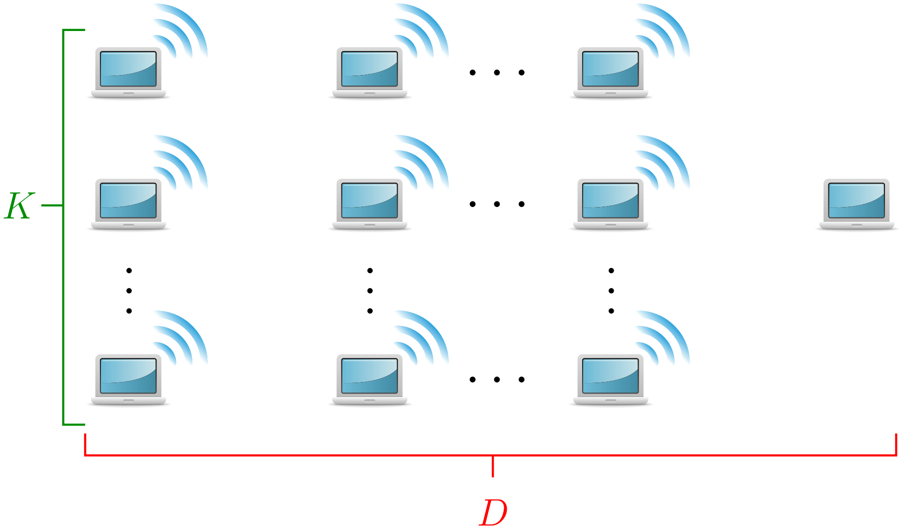
$$R_{sum} = K \log \mathsf{SNR} - 7K^3 \ .$$

- Penalty term is due to setting rates using the weakest coefficient.
- For equal magnitudes, $|h_{k\ell}| = 1$, we can achieve $R_{\mathsf{sum}} = K \log \mathsf{SNR}$.



Avestimehr-Diggavi-Tse '11: $C \geq K \log \mathsf{SNR} - g_1(K, D)$

This talk: $C \geq K \log \mathsf{SNR} - g_2(K, \mathsf{fading})$



Avestimehr-Diggavi-Tse '11: $C \geq K \log \mathsf{SNR} - KD$

This talk: $C \geq K \log \mathsf{SNR} - 7K^3$

Conclusions and Future Work

- Compute-and-forward combined with alignment as a new tool for capacity approximation.
- More work is needed to get a constant gap that does not depend on the fading statistics.
- Multiple receivers. Layered interference channels.
- Fixed channel coefficients.
- For more info: ISIT '11 tutorial slides on Algebraic Structure in Network Information Theory available at `iss.bu.edu/bobak`