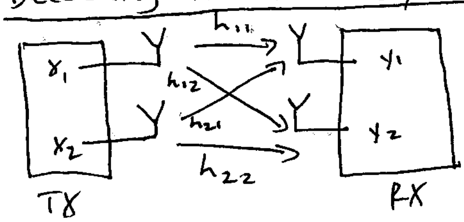
MIMO: multiple input - multiple output
SISO: single input, single output
MISO: multiple input, single output
SIMO: single input, multiple output

M x N MIMO – TX has M antennas
RX has N antennas

<u>Decoding in MIMO systems:</u>



$y_1 = h_{11}x_1 + h_{21}x_2$

$y_2 = h_{12}x_1 + h_{22}x_2$

$\vec{y} = H\vec{x}$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{21} \\ h_{12} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

↑ Channel matrix H

If H is known $\Rightarrow H^{-1}\vec{y} = H^{-1}H\vec{x} = \vec{x}$

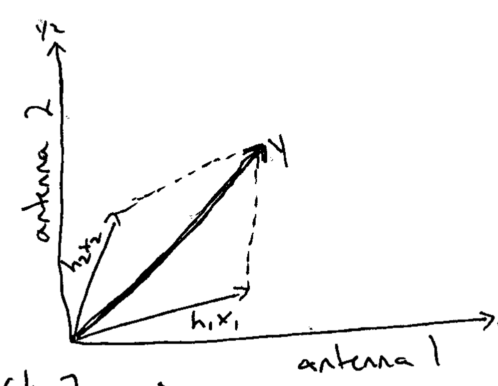
With a 1x1 system, we can deliver 1 packet.
With a 2x2 system, we can deliver 2 packets at once $\Rightarrow$ multiplexing gain

The matrix H is known to be invertible if the inter-antenna spacing at the transmitter and receiver exceeds a certain # of wavelengths. This ensures that the h_{ij}'s are different and hence the matrix is almost definitely invertible.

Since matrix inversion is an $O(n^3)$ operation, we incur large overhead as we increase the # of antennas.

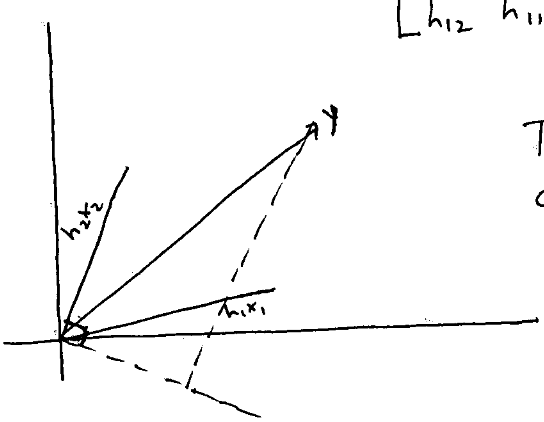
To avoid calculating H^{-1}, we use another approach:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} \\ h_{12} \end{bmatrix} x_1 + \begin{bmatrix} h_{21} \\ h_{22} \end{bmatrix} x_2 \qquad \vec{y} = \vec{h}_1 x_1 + \vec{h}_2 x_2$$



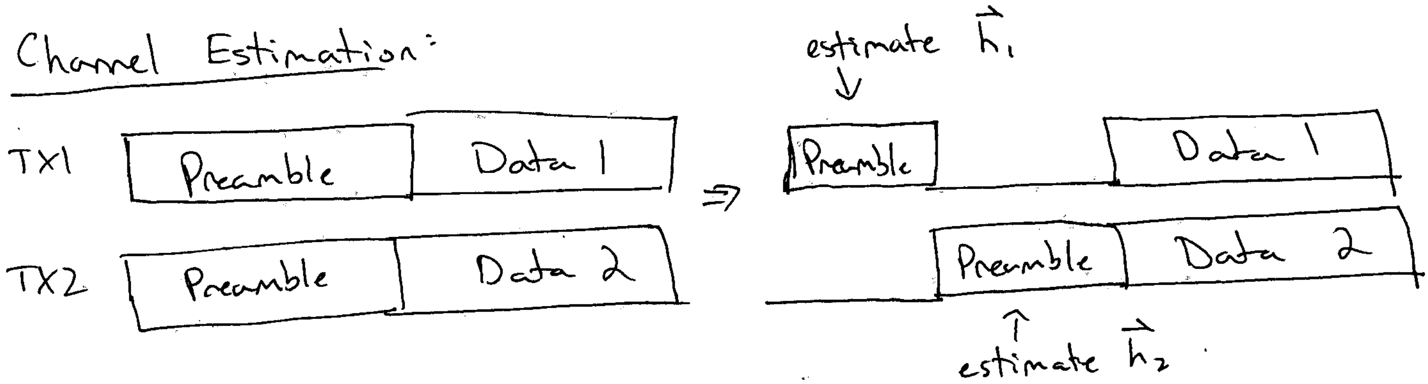
We can acquire x_1 by projecting $\vec{y}$ on a direction perpendicular to $\vec{h}_2$. x_2 can be gotten by projecting similarly on a direction perpendicular to $\vec{h}_1$.

$$[h_{12}\ \ h_{11}]\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \overset{0}{\cancel{[h_{12}\ \ -h_{11}]\begin{bmatrix} h_{11} \\ h_{12} \end{bmatrix}}} x_1 + \cancel{h_{12}} + [h_{12}\ \ -h_{11}]\begin{bmatrix} h_{21} \\ h_{22} \end{bmatrix} x_2$$



This approach scales linearly, but projection causes a loss of SNR.

Channel Estimation:



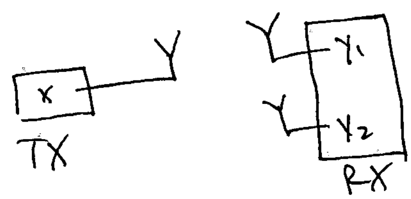
Preambles need to be transmitted separately which causes slight overhead.

MIMO Gain:

Multiplexing gain: $M \times M$ MIMO $\Rightarrow$ M packets at the same time

Diversity gain: $M \times M$ MIMO $\Rightarrow$ 1 packet M times $\rightarrow$ increases SNR

Diversity gain:



Let us initially assume the channel is 1.

$$y_1 = x + n_1 \Rightarrow 2x + n_1 + n_2$$
$$y_2 = x + n_2$$

$$SNR = \frac{Power(x)}{Power(noise)} = \frac{E[(2x)^2]}{E[(n_1+n_2)^2]} = \frac{4E[x^2]}{E[n_1^2]+E[n_2^2]} = \frac{2E[x^2]}{\sigma^2}$$

This is a doubling of the SNR as a SISO system with a channel of 1 has a $SNR = \frac{E[x^2]}{\sigma^2}$.

Now let us assume there is a non-trivial channel:

$$y_1 = h_1 x_1 + n_1 \qquad h_1^* y_1 = |h_1|^2 x_1 + h_1^* n_1 \Rightarrow (|h_1|^2 + |h_2|^2) x_1 + h_1^* n_1 + h_2^* n_2$$
$$y_2 = h_2 x_1 + n_1 \qquad h_2^* y_2 = |h_2|^2 x_1 + h_2^* n_2$$

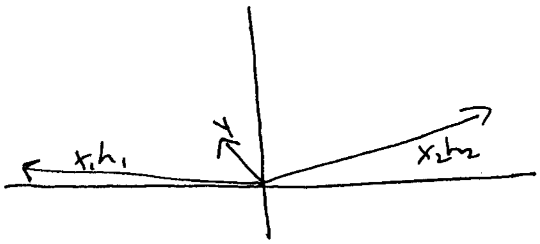
$$SNR = \frac{(|h_1|^2+|h_2|^2)^2 E[x^2]}{|h_1|^2\sigma^2 + |h_2|^2\sigma^2} = (|h_1|^2+|h_2|^2) \cdot \frac{E[x^2]}{\sigma^2} \qquad gain = \frac{|h_1|^2+|h_2|^2}{|h_1|^2}$$

We know capacity is proportional to $\log(SNR)$, with two antennas on the receiver, that becomes $\log 2 + \log(SNR)$. The lower the SNR regime, the larger influence $\log 2$ has on the capacity.

Alamouti Codes

With a 2x2 MIMO system, we have: $\vec{y} = x_1\vec{h}_1 + x_2\vec{h}_2$.

A problem that can arise is if $x_1\vec{h}_1$ and $x_2\vec{h}_2$ nearly cancel.



To account for this, instead of transmitting

	t	$t+1$
TX1	x_1	x_2
TX2	x_1	x_2

instead we transmit

	t	$t+1$
TX1	x_1	$-x_2^*$
TX2	x_2	x_1^*

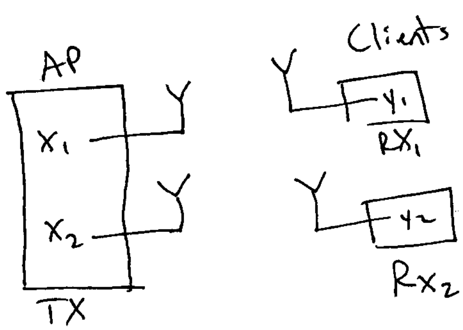
$$\vec{y}_t = \vec{h}_1 x_1 + \vec{h}_2 x_2 \Rightarrow h_1^*\vec{y}_t = h_1^*\vec{h}_1 x_1 + h_1^*\vec{h}_2 x_2$$

$$\vec{y}_{t+1} = \vec{h}_1(-x_2^*) + \vec{h}_2 x_1^* \Rightarrow h_2\vec{y}_{t+1} = h_2^*\vec{h}_1(-x_2^*) + h_2^*\vec{h}_2 x_1^*$$

$$h_1^*\vec{y}_t + \vec{h}_2 y_t^* = |h_1|^2 x_1 + h_1^*\vec{h}_2 x_2 - \vec{h}_2 h_1^* x_2 + |h_2|^2 x_1 = (|h_1|^2 + |h_2|^2)x_1$$

$$h_2^* y_{t+1} + h_1 y_t^* = (|h_1|^2 + |h_2|^2)x_2$$

MU-MIMO: Multiple User MIMO



$$y_1 = h_{11}x_1 + h_{21}x_2$$

$$y_2 = h_{12}x_1 + h_{22}x_2$$

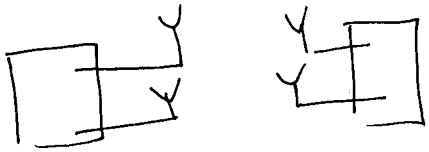
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = H\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Beamforming: (Precoding)

$$\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = H^{-1}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = HH^{-1}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

TX must know H

Frequency Offset:



$$y_1 = h_{11} x_1 e^{j2\pi \Delta f_{11} t} + h_{21} x_2 e^{j2\pi \Delta f_{21} t}$$

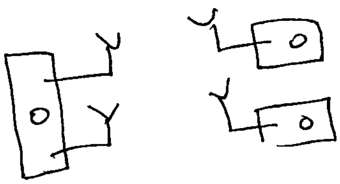
$$y_2 = h_{12} x_1 e^{j2\pi \Delta f_{22} t} + h_{22} x_2 e^{j2\pi \Delta f_{22} t}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} e^{j2\pi \Delta f_{11} t} & h_{21} e^{j2\pi \Delta f_{21} t} \\ h_{12} e^{j2\pi \Delta f_{12} t} & h_{22} e^{j2\pi \Delta f_{22} t} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

As antennas are on the same board, they use the same oscillator:

$\Delta f_{11} = \Delta f_{12} = \Delta f_{21} = \Delta f_{22}$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = e^{j2\pi \Delta f t} \begin{bmatrix} h_{11} & h_{21} \\ h_{12} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



$\Delta f_{11} = \Delta f_{21}$

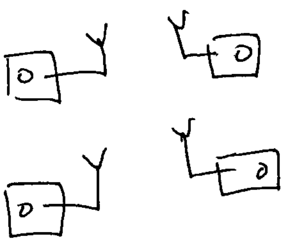
$\Delta f_{12} = \Delta f_{22}$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} e^{j2\pi \Delta f_{11} t} & 0 \\ 0 & e^{j2\pi \Delta f_{22} t} \end{bmatrix} \begin{bmatrix} h_{11} & h_{21} \\ h_{12} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$y_1 = e^{j2\pi \Delta f_{11} t} x_1$

$y_2 = e^{j2\pi \Delta f_{22} t} x_2$

$\Rightarrow$ still independent, can still transmit $\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = H^{-1} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$



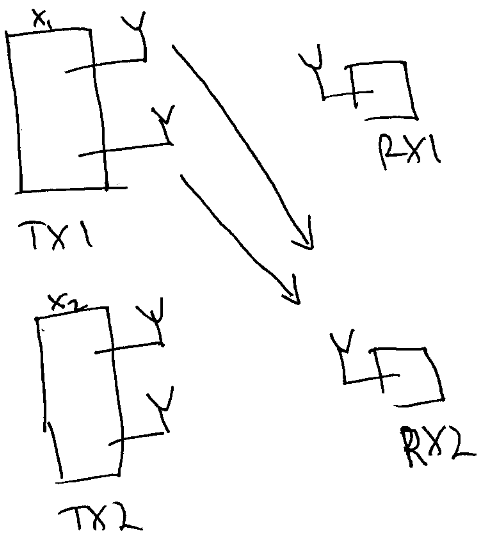
Virtual MIMO: $\Delta f_{11} \neq \Delta f_{12} \neq \Delta f_{21} \neq \Delta f_{22}$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} e^{j2\pi f_{11} t} & h_{21} e^{j2\pi f_{21} t} \\ h_{12} e^{j2\pi f_{12} t} & h_{22} e^{j2\pi f_{22} t} \end{bmatrix} \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}$$

functions of time, extremely hard to track in real-time, error will accumulate

Interference Nulling:

Each transmitter uses 2nd antenna to null his transmission at the other receiver



$$y_2 = h_{11}x_1 + h_{21}x_1$$

$$\Downarrow$$

instead ~~of~~ transmit $-h_{21}x_1$ and $h_{11}x_1$

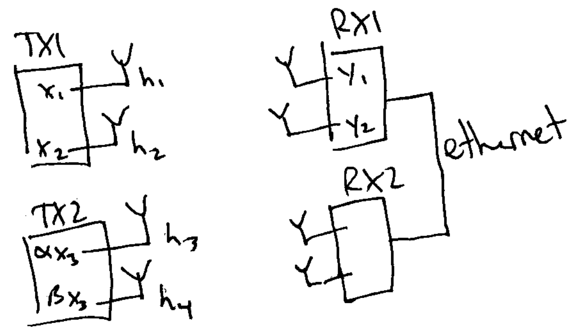
$$\Downarrow$$

$$y_2 = h_{11}(-h_{21})x_1 + h_{21}h_{11}x_1 = 0$$

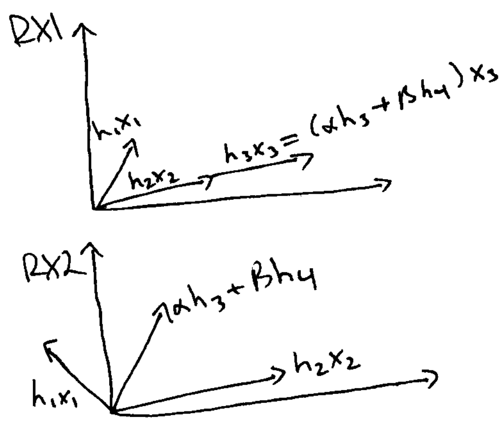
Each additional antenna on the receiver needs two additional on the transmitter to perform interference nulling.

Interference Alignment:

Uplink:



Aligning 2 of the vectors



$(\alpha \vec{h}_3 + \beta \vec{h}_4)x_3$

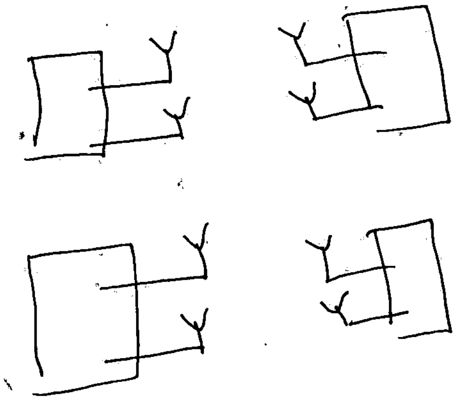
set $\alpha \vec{h}_3 + \beta \vec{h}_4$ proportional to $\vec{h}_2$

(h's are vectors, 2 equations and 2 unknowns)

RX2 receives $\vec{h}_1 x_1$ over ethernet, subtracts it, and solves for remaining two, sends back to RX1

With this, can transmit 3 packets instead of 2 at once

Downlink:



On downlink you can only send 1 and 1 packet.

While this is initially not as good as interference nulling, it scales better.