STANFORD

Lecture 16

Optimal IC Design

May 29, 2026

JOHN M. CIOFFI

Hitachi Professor Emeritus of Engineering

Instructor EE379B – Spring 2026

Announcements & Agenda

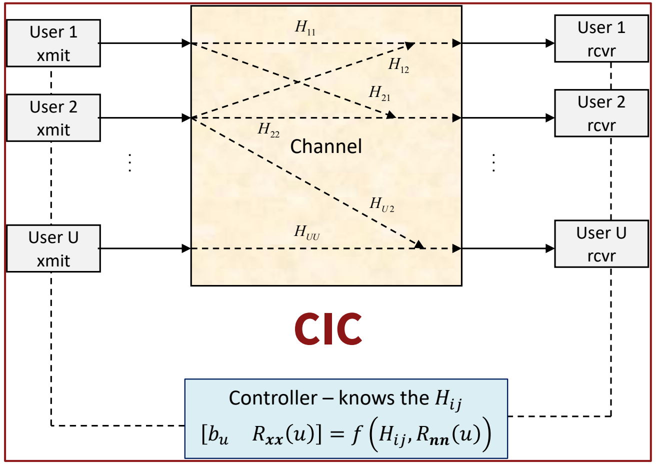
- Announcements
  - hihkjh

- Agenda
  - IC Review
  - minPIC
  - Spectrum Balancing and Iterative Waterfilling

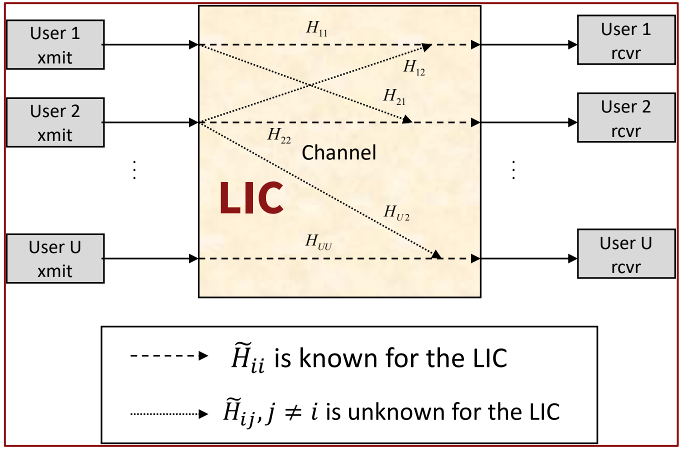
IC Review

Central or Local Control?

- **Centrally controlled IC (CIC):** controller directs users, &
  - centrally sets $b_{u,l,n}$, $\mathcal{E}_{u,l,n}$, $R_{\boldsymbol{xx}}(u,n)$, users' codes,
  - knows $H_{u \leftrightarrow u',l,n}$, $R_{\boldsymbol{nn}}(u,n)$ $\forall\ u \in \boldsymbol{u}$.
  - Examples include:
    - Distributed Antenna Systems (DAS)
    - Cell-free with mobile-edge computation &
    - CIC is often associated with licensed spectra (e.g., cellular).
  - User receivers use successive decoding (GDFE).
  - Basically, this is the IC you know already.

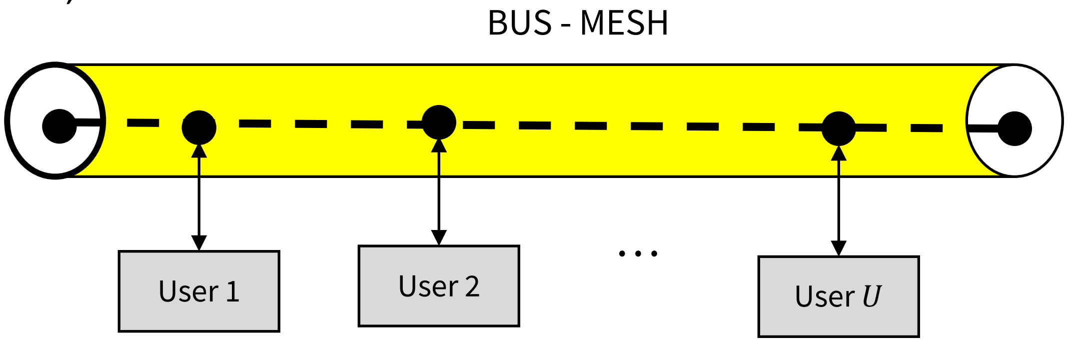


- **Locally controlled IC (LIC):** each user transmitter can only control its own transmission.
  - Locally (transceiver u) sets $b_{u,l,n}$, $\mathcal{E}_{u,l,n}$, $R_{\boldsymbol{xx}}(u,n)$, & codes.
  - Local user knows only $H_{u \leftrightarrow u,l,n}$, $\mathcal{R}_{\boldsymbol{nn}}(u,n)$.
  - Examples include:
    - Collision detection, avoidance are popular.
    - Unlicensed spectra (Wi-Fi, Bluetooth).
  - Everyone else is noise (try to detect and remove them at your own risk).



Another Interference Channel Form

- The BUS model;



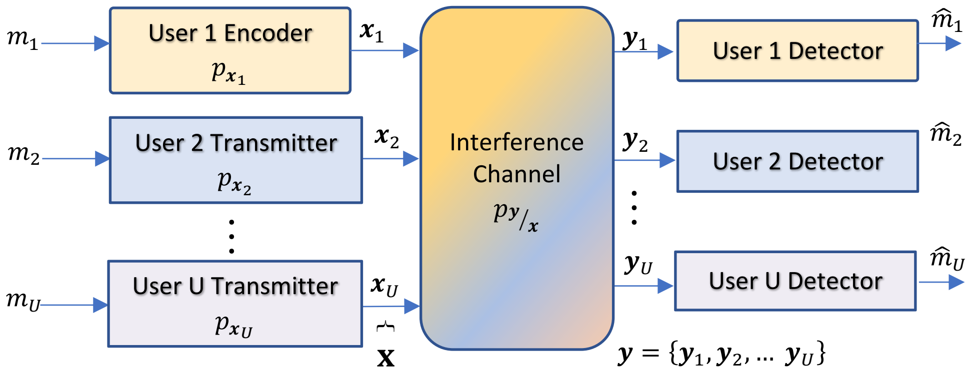
- All connections are bi-directional and the transmitter/receiver are independent; $U' = U \cdot (U-1)$.
  - This is actually $U \cdot (U-1)$-user MACs with $U \cdot (U-1)$-user BCs. And then, we can consider subusers ….

- Restrictions to IC might include:
  - Only U messages are permitted (e.g., a "switch"), and there is a transmit/receive pair for each.
  - In any given symbol period → time-varying IC.

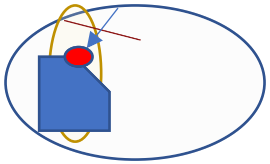
- The "yellow" might be a common wire(s) or air (common shared spectrum).

- Designs, so far, are "ODM" (**orthogonal division multiplex** – all users occupy mutually separate dimensions.)

The CIC



- Studied earlier and found an order-indexed set of vertices $\mathcal{I}_{min}(\boldsymbol{\Pi}, p_{xy})$.
- Then take convex combinations over the possible $(U^2!)^U$ orders of the $\mathcal{I}_{min}(\boldsymbol{\Pi}, p_{xy})$.



The achievable-region remains convex, and the Lagrangian $\boldsymbol{\theta}$ is equivalent to order.

minPIC

minPIC = more "optimum"

- minPIC receiver u cancels all subusers $i \in \mathcal{D}_u(\boldsymbol{\Pi}, p_{\boldsymbol{xy}}, \boldsymbol{b})$; the decodable set.
- Order has been restored 😀.
- The optimization is
  - $(i, u) = (RCVR, USER)$

$$\min_{\{R_{\boldsymbol{xx}}(i,u,n)\}} \sum_{n=0}^{\overline{N}-1} \sum_{u=1}^{U} w_u \cdot \underbrace{\text{trace}\left\{\sum_{i=1}^{U} R_{\boldsymbol{xx}}(i,u,n)\right\}}_{\mathcal{E}_{u,n}}$$

$$ST: \quad b_u \geq b_{min,u}$$
$$R_{\boldsymbol{xx}}(i,u,n) \succeq \mathbf{0} \ .$$

- $\boldsymbol{\theta}$ again has U terms that determine the "**feasible**" U^2-dimensional orders $\boldsymbol{\Pi}$.
- Feasible orders retain a convex achievable-region constraint (like minPMAC).
- Implement GDFE at each receiver (no transmitter nonlinear precoders).

Review MU Capacity step: Prior-User Set

- Order vector and inverse, or
  - Permutation (permutation matrix J), etc

$$\boldsymbol{\pi}_u = \begin{bmatrix} \pi(U') \\ \vdots \\ \pi(1) \end{bmatrix} \quad \boldsymbol{\pi}_u^{-1} = \begin{bmatrix} U' \\ \vdots \\ 1 \end{bmatrix} \qquad j = \pi(i) \;\rightarrow i = \pi^{-1}(j).$$

- Prior-User Set is $\mathbb{P}_u(\boldsymbol{\pi}) = \{j \mid \boldsymbol{\pi}_u^{-1}(j) < \boldsymbol{\pi}_u^{-1}(u)\}$.
  - That is "all the subusers before the desired user u" in the given order $\boldsymbol{\pi}_u$ at receiver u.
  - Receiver u best decodes these "prior" users and removes them, while "post" users are noise.
  - $\boldsymbol{\pi}_u$ can be any order in $\mathbb{P}_u(\boldsymbol{\pi}_u)$ that is chosen in design.

rcvr/ User i	$\boldsymbol{\pi}_4(i)$	$\boldsymbol{\pi}_3(i)$	$\boldsymbol{\pi}_2(i)$	$\boldsymbol{\pi}_1(i)$
$i = 4$	3	3	4	3
$i = 3$	4	2	3	2
$i = 2$	1	4	2	1
$i = 1$	2	1	1	4
$\mathbb{P}_u(\boldsymbol{\pi}_u)$	{1,2}	{2,4,1}	{1}	{4}

$$\boldsymbol{\Pi} = \begin{bmatrix} 3 & 3 & 4 & 3 \\ 4 & 2 & 3 & 2 \\ 1 & 4 & 2 & 1 \\ 2 & 1 & 1 & 4 \end{bmatrix}$$

Assumes $U' = 4$; so, no subusers

Generally could have $(16)^4$ Orders.

- **Data rates** (mutual information bounds) depend on those user rates that are decoded/cancelled; these energies are equivalent to a good-code choice and consequent rate(s).

$\mathfrak{I}$	$\mathfrak{I}_4$	$\mathfrak{I}_3$	$\mathfrak{I}_2$	$\mathfrak{I}_1$
top	∞	$\mathcal{I}_3(3/1,2,4)$ 20	∞	∞
	$\mathcal{I}_4(4/1,2)$ 10	$\mathcal{I}_3(2/1,4)$ 9	∞	∞
	$\mathcal{I}_4(1/2)$ 5	$\mathcal{I}_3(4/1)$ 4	$\mathcal{I}_2(2/1)$ 4	$\mathcal{I}_1(1/4)$ 2
bottom	$\mathcal{I}_4(2)$ 1	$\mathcal{I}(1)$ 2	$\mathcal{I}_2(1)$ 2	$\mathcal{I}_1(4)$ 5

$$\mathcal{I}_{min}(\boldsymbol{\Pi}, p_{\boldsymbol{xy}}) = \begin{bmatrix} 4 \\ 20 \\ 1 \\ 2 \end{bmatrix}$$

3-User Order example

- Given a $\boldsymbol{\theta}$, say for example with $\theta_3 > \theta_1 > \theta_2$, they determine all receivers' order:

$$FOR: \quad \theta_3 \; > \; \theta_1 \; > \; \theta_2 \; > 0$$

(RCVR, USER)	Receiver 3	Receiver 1	Receiver 2
	(1,1),(2,1),(1,2),(2,2)	(2,2),(3,2),(2,3),(3,3)	(1,1),(3,1),(1,3),(3,3)
	(3,3)	(1,3)	(2,3)
	(1,3)	**(3,1)**	(2,1)
	(2,3)	**(1,1)**	**(3,2)**
	(3,1)	**(2,1)**	**(1,2)**
	(3,2)	(1,2)	**(2,2)**

THESE ARE constant XTALK, AND CAN BE 0

Π

$$\left\{\sum_{u=1}^{3} \theta_u \cdot b_u\right\}_{RCVR3opt} = (\theta_3 - \theta_1) \cdot \log_2(B) + (\theta_1 - \theta_2) \cdot \log_2(C) + \theta_2 \cdot \log_2(D)$$

- Any other order is inconsistent with the Lagrangian multipliers' interpretation and capacity/achievable region's convexity

$$\begin{aligned} A &\triangleq |H_{3,1}|^2 \cdot (\mathcal{E}_{1,1} + \mathcal{E}_{2,1}) + |H_{3,2}|^2 \cdot (\mathcal{E}_{1,2} + \mathcal{E}_{2,2}) + I \\ B &\triangleq |H_{3,3}|^2 \cdot \mathcal{E}_3 + A \\ C &\triangleq |H_{3,1}|^2 \cdot \mathcal{E}_{3,1} + B \\ D &\triangleq |H_{3,2}|^2 \cdot \mathcal{E}_{3,2} + C \end{aligned}$$

RCVR 3

$$\begin{aligned} b_3 &= \log_2(B) - \log_2(A) \\ b_{3,1} &= \log_2(C) - \log_2(B) \\ b_{3,2} &= \log_2(D) - \log_2(C) \end{aligned}$$

Do same for other 2 receivers

- RCVR 1 optimization of rate sum

$$\begin{aligned} A &\triangleq |H_{1,3}|^2 \cdot (\mathcal{E}_{2,3} + \mathcal{E}_{3,3}) + |H_{1,2}|^2 \cdot (\mathcal{E}_{1,2} + \mathcal{E}_{2,2}) + I \\ B &\triangleq |H_{1,3}|^2 \cdot \mathcal{E}_{1,3} + A \\ C &\triangleq |H_{1,1}|^2 \cdot \mathcal{E}_1 + B \\ D &\triangleq |H_{1,2}|^2 \cdot \mathcal{E}_{1,2} + C \end{aligned}$$

RCVR 1

$$\begin{aligned} b_{1,3} &= \log_2(B) - \log_2(A) \\ b_1 &= \log_2(C) - \log_2(B) \\ b_{1,2} &= \log_2(D) - \log_2(C) \quad . \end{aligned}$$

$$\left\{\sum_{u=1}^{3} \theta_u \cdot b_u \right\}_{RCVR1opt} = (\theta_3 - \theta_1) \cdot \log_2(B) + (\theta_1 - \theta_2) \cdot \log_2(C) + \theta_2 \cdot \log_2(D)$$

- RCVR 2 optimization of rate sum

$$\begin{aligned} A &\triangleq |H_{2,3}|^2 \cdot (\mathcal{E}_{1,3} + \mathcal{E}_{3,3}) + |H_{2,1}|^2 \cdot (\mathcal{E}_{1,1} + \mathcal{E}_{3,1}) + I \\ B &\triangleq |H_{2,3}|^2 \cdot \mathcal{E}_{2,3} + A \\ C &\triangleq |H_{2,1}|^2 \cdot \mathcal{E}_{2,1} + B \\ D &\triangleq |H_{1,1}|^2 \cdot \mathcal{E}_2 + C \end{aligned}$$

RCVR 2

$$\begin{aligned} b_{2,3} &= \log_2(B) - \log_2(A) \\ b_{2,1} &= \log_2(C) - \log_2(B) \\ b_2 &= \log_2(D) - \log_2(C) \quad . \end{aligned}$$

$$\left\{\sum_{u=1}^{3} \theta_u \cdot b_u \right\}_{RCVR2opt} = (\theta_3 - \theta_1) \cdot \log_2(B) + (\theta_1 - \theta_2) \cdot \log_2(C) + \theta_2 \cdot \log_2(D)$$

- Six energies repeat – select the smallest that has corresponding lowest rate for its transmitter.
- Outer $\boldsymbol{\theta}$ loop (e.g., Ellipsoid) remains the same as minPMAC.

Generalize – first order them to simplify

- Create order of users for each of (reordered) users

θ_U $U^2 \setminus \{(1:U,U),(U,1:U-1)\}$	… …	θ_u $U^2 \setminus \{(1:U,u),(u,1:U-1)\}$	… …	θ_1 $U^2 \setminus \{(U,1:U),(1,1:U-1)\}$
(U,U)	…	(u,U)	…	1,U-1
$\vdots$		$\vdots$		$\vdots$
	…	$(u,U-u+1)$	…	$(1,1)$
$(1,U)$	…	(U,u)	…	$(U,1)$
$(U,U-1)$	…	$\vdots$	…	$\vdots$
$\vdots$	…	$(1,u)$	…	$\vdots$
		$(u,U-u-1)$		
$\vdots$	…	$\vdots$	…	$\vdots$
$(U,1)$	…	$(u,1)$	…	(U,U)

Table 5.2: **Generalized of overall decoding order pairs given descending-order θ.**

Generalize A,B,C, D

$$
\begin{array}{lll}
K_{1,u} & \triangleq \sum_{i\neq u} H_{u,i} \cdot \left(\sum_{j\neq i} R_{\boldsymbol{xx}}(i,j)\right) \cdot H^*_{u,i} + I & \text{for } \; b_{u,U} \\
K_{2,u} & \triangleq H_u(2) \cdot R_{\boldsymbol{xx}}(u, 2U-3, 2) \cdot H^*_u(2^{nd}) + K_{1,u} & \text{for } \; b_{u,U-} \\
\vdots & \vdots & \\
K_{u,u} & \triangleq H_{u,2U-u+1}(2) \cdot R_{\boldsymbol{xx}}(u, 2U-u+1(2^{nd})) \cdot H^*_{u,2U-u+1}(2) + K_{u-1,u} & \text{for } \; b_u \\
\vdots & \vdots & \\
K_{2U-2,u} & \triangleq H_{u,1}(2) \cdot R_{\boldsymbol{xx}}(u, 1(2^{nd})) \cdot H^*_{u,1}(2) + K_{2U-3,u} & \text{for } \; b_{u,1}
\end{array}
$$

$$
\left\{\sum_{u=1}^{U} \theta_u \cdot b_u\right\}_{RCV\,Ruopt} = (\theta_U - \theta_{U-1}) \cdot \log_2(K_{1,u}) + \ldots + (\theta_2 - \theta_1) \cdot \log_2(K_{2U-3,u}) + \theta_2 \cdot \log_2(K_{2U-2,u})
$$

- This is convex in those quantities optimized
- Need the outer subgradient loop on theta to drive IC rate vector to bmin.

All done tacitly in minPIC

minPIC Examples

Spectrum Balancing and Iterative Waterfilling

CIC's "Optimum" Spectrum Balancing (no GDFEs)

- **OSB** minimizes weighted sums for given $\boldsymbol{b}_{min}$ or $\boldsymbol{\mathcal{E}}_{max}$.
  - OSB uses no crosstalk cancellation.
  - It is not "optimum" (minPIC is optimum).

$$\min_{\{\mathcal{E}_{u,n}\}} \sum_{u=1}^{U} w_u \cdot \mathcal{E}_u \qquad \max_{\{\mathcal{E}_{u,n}\}} \sum_{u=1}^{U} \theta_u \cdot b_u$$

- OSB relates $\boldsymbol{b}$ *to* $\boldsymbol{\mathcal{E}}$ *, with all others as noise:*
  - $\mathcal{R}_{noise}(u,n) = I + \sum_{i \neq u} \tilde{H}_{u,i,n} \cdot R_{\boldsymbol{xx}}(i,n) \cdot \tilde{H}^*_{u,i,n}$.
  - There is no decoding order.

$$ST: 0 \leq \sum_n \mathcal{E}_{u,n} \leq \mathcal{E}_{u,max} \;\; or \;\; 0 \leq \sum_n b_{u,n} \geq b_{u,min}$$

$$b_u = \sum_n \log_2 \left\{ \frac{\left|H_{uu,n}\right|^2 \cdot \mathcal{E}_{u,n} + \mathcal{R}_{noise}(u,n)}{\mathcal{R}_{noise}(u,n)} \right\}$$

- Tonal Lagrangian OSB – "dual-decomposition:"
  - minimizes each tone's $L_n(R_{xx}, \boldsymbol{b}_n)$ individually, and then sums,
  - is not convex (no sequential-differences transformation), &
  - tests all energies/bits in quantized steps (exponentially complex).

$$L_n(R_{\boldsymbol{xx}}(u,n), \boldsymbol{b}_n, \boldsymbol{w}, \boldsymbol{\theta}) = \sum_{u=1}^{U} w_u \cdot \mathcal{E}_{u,n} - \theta_u \cdot b_{u,n}$$

RA OSB, energy minimization similar

Definitions (drop n's below)

$\tilde{H}_{u,u,n} \triangleq \frac{H_{u,u,n}}{S_{u,n}^{1/2}}\ \forall\ u \in [1:U]$ and $i \in [1:U]$

$\widetilde{SNR}_{u,n} \triangleq \Gamma \cdot \left(2^{2\cdot \bar{b}_{u,n}} - 1\right) = \frac{\bar{\mathcal{E}}_{u,n} \cdot \left|\tilde{H}_{u,u,n}\right|^2}{1 + \sum_{i \neq u} \bar{\mathcal{E}}_{i,n} \cdot \left|\tilde{H}_{u,i,n}\right|^2}$

- **Energy step:** Compute $(\bar{b}_{max} + 1)^U$ possible $\widetilde{SNR}_{u,n}$ values (the following drops the n subscript).
  - These may also have a power-spectrum "mask" that limits tonal energy.
  - OSB then checks energy, summing over all tones and taking this smallest weighted energy sum as the present solution.

- **EM OSB solves** this linear equation $(b_{max} + 1)^U$ times (for each tone) to find possible U-tuples for $\{\bar{\mathcal{E}}_{u,n}\}$:

$$\begin{bmatrix} |\tilde{H}_{11}|^2 & -\widetilde{SNR}_1 \cdot |\tilde{H}_{12}|^2 & \ldots & -\widetilde{SNR}_1 \cdot |\tilde{H}_{1U}|^2 \\ -\widetilde{SNR}_2 \cdot |\tilde{H}_{21}|^2 & |\tilde{H}_{22}|^2 & \ldots & -\widetilde{SNR}_2 \cdot |\tilde{H}_{2U}|^2 \\ \vdots & \vdots & \ddots & \vdots \\ -\widetilde{SNR}_U \cdot |\tilde{H}_{U1}|^2 & -\widetilde{SNR}_U \cdot |\tilde{H}_{U(U-1)}|^2 & \ldots & |\tilde{H}_{UU}|^2 \end{bmatrix} \cdot \begin{bmatrix} \bar{\mathcal{E}}_1 \\ \bar{\mathcal{E}}_2 \\ \vdots \\ \bar{\mathcal{E}}_U \end{bmatrix} = \begin{bmatrix} \widetilde{SNR}_1 \\ \widetilde{SNR}_2 \\ \vdots \\ \widetilde{SNR}_U \end{bmatrix}$$

OSB computes weighted sum over all tones and keeps the smallest.

- **Constraint:** Use sub-gradient descent method to update the w Lagrange multiplier for rate constraints $\mathcal{E}_{max}$.

$$\Delta\mathcal{E} = \mathcal{E}_{max} - \sum \mathcal{E}_n \qquad \boldsymbol{w} \leftarrow \boldsymbol{w} + \epsilon' \cdot \Delta\mathcal{E}$$

- **For MA:** Change to $M = \frac{\max_{u,n} \mathcal{E}_{u,n}}{\Delta\mathcal{E}}$ and evaluate and solve linear equations for $\widetilde{SNR}_{u,n}$ with given $\bar{\mathcal{E}}_{u,n}$.

$$\Delta b = b_{min} - \sum_n b_n \qquad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \epsilon \cdot \Delta \boldsymbol{b}$$

OSB.m (rate adaptive)

```
function [S1, S2, b1, b2] = osb(Hmag_sq, No, E, theta, mask, ...
 gap, bitcap, cb)

 osb and also finds w1 energy weight for USER 1
 A. Chowdhery ~2010 ; Updated by J. Cioffi in 2024.  It presently
handles only 2 users, so U=2.

 Inputs

 Hmag_sq is a N x 2 x 2 where N is FFT size. N inferred from this.
 No      is a 1 x U white-noise power spectra density matrix.
    If Hmag_sq is complex BB, then No should be the one-sided PSD.
 E       is a 1 x U energy vector.
 theta   is a 1 x U user-rate weighting vector.
 mask    is an N x U PSD maximum allowed.
 gap     is the (non-dB) linear gap (so 1 if 0 dB gap).
 bitcap  is a 1 x U maximum number of bits allowed per tone.
 cb      is 2 for real baseband and 1 for cplex bband

 Outputs

 S1      is user 1's Nx1 PSD
 S2      is user 2's Nx1 PSD
 b1      is user 1's Nx1 bit distribution
 b2      is user 2's Nx1 bit distribution

calls optimize_l2.m, which calls optimize_s.m
User order is reversed with respect to class convention.
```

Only tests integer bits/tone up to bitcap
Discards solutions that exceed user energy

```
>> H2
H2(:,:,1) =   0.6400    0.2500   % note this is squared mag each term
H2(:,:,2) =   0.4900    0.3600
>> Noise =  1.0e-04 * [   1.0000    1.0000];
>> Ex =[    1     1];
>> mask =[    1     1];
>> gap =     1;
>> bitcap =[   15    15];
>> [S1, S2, b1, b2] = osb(H2, Noise, Ex, [5 .5], mask, gap, bitcap,2)

S1 =   0.6398
S2 =    0
b1 =    6  % note < 6.3 for the GDFE based IC's maximum L11:16
b2 =    0
>> [S1, S2, b1, b2] = osb(H2, Noise, Ex, [0.01 .99], mask, gap, bitcap,2)
S1 =    0
S2 =   0.1419
b1 =    0
b2 =   4.5000  <5.9 for L11:16
```

- The OSB search can be very complex for $U > 3$, so this program only works for $U = 2$.
  - Extra credit to someone who wants to generalize it.

Multitone OSB

```
h = cat(3, [1  .8 ;  -1   1], [-.9  -.7 ;  0  1]  )*10;
He = fft(h, 8, 3);
>> H3=zeros(8,2,2);
>> H3(:,1,1)=He(1,1,:);
>> H3(:,2,1)=He(2,1,:);
>> H3(:,2,2)=He(2,2,:);
>> H3(:,1,2)=He(1,2,:);
>> Noise=ones(8,2);
>> mask=ones(8,2);
>> Ex=8*Ex;
>> [S1, S2, b1, b2] = osb(H3.*conj(H3), Noise, Ex, [0.5 .5], mask, gap,
bitcap,1);
>> S1' =    0         0        0.7017   0.8272   0   0.8272   0.7017      0
>> S2' =   0.6375   0.7469       0        0         0       0         0          0.7469

>> b1' =    0    0    7    8    0    8    7    0
>> b2' =    8    8    0    0    0    0    0    8
>> sum(b1) =   30
>> sum(b2) =   24
sum(b1+b2) =   54 % < ~116 that MAC, BC, single had for this channel
```

Same 2x2 channel As in L16.

OSB solution is often FDM

```
>> [S1, S2, b1, b2] = osb(H3.*conj(H3), Noise, Ex, [0.5 .5], 1000*mask,
gap, bitcap,1);
>> S1'  =    0       0       0   1.6576      0   1.6576       0       0
>> S2'=   1.2775   1.4967   1.2750      0       0       0   1.2750   1.4967

>> b1' =    0    0    0    9    0    9    0    0
>> b2' =    9    9    8    0    0    0    8    9
>> sum(b1) =   18
>> sum(b2) =   43
61 is bit sum << 116
>> sum(S1) =   3.3152
>> sum(S2) =   6.8209
```

Energies < 8; it is an **interference limited channel.** (Sum of interference >> noise and scales with increasing energy → constant SNRs.)

- GDFE's cancellation of crosstalk makes a large difference for this channel.

```
>> sum(S1)  =   3.0577
>> sum(S2) =   2.1313
```

IW_polite.m (RA, integer bits like osb.m)

```
% function [b, E] = iw_polite(N, df, U, Hmag, No, Ex, mask, gap, mode,
b_target, bitcap,cb)
%
% Calculates data rates of M users and corresponding bit distributions and
Energy distributions
% using iterative waterfilling
%
% Inputs
% ------
% N: number of sub-channels
% M: number of users
% Hmag: squared channel transfer and crosstalk matrix (N x U x U matrix)
% Hmag(n,i,j) is the crosstalk transfer function from loop i to j at the
nth bin.
% No: noise energy/sample
% Ex: signal energy/SYMBOL
% mask: PSD mask - largest value N x U
% gap: gap (not in dB)
% mode: (U x 1 vector) each value is one of the followings
% 0 - rate adaptive
% 1 - fixed margin (power minimization)
% 2 - margin adaptive
% b_target: target bits on 1 DMT symbol for modes 1 and 2
% bitcap: maximum possible number of bits at each frequency bin
% cb =1 for cplx BB and =2 for real BB
%
% Outputs
% -------
% b: bit distribution (N x U matrix)
% E: energy distribution (N x U matrix)
%
% Remarks
% Iterate waterfiling for each user 10 times
% Youngjae(Sean) Kim - modified J. Cioffi, April 2024
```

```
>> gap=1;
>> [b, E] = iw_polite(8, 2, H3.*conj(H3), Noise, [8 8], 1000*mask,
gap, [0 0], [20  60], bitcap,1)
b =    0   10
       0    9
       2    1
       9    0
       9    0
       9    0
       2    1
       0    9
E =  0        2.5575
     0        1.4967
  1.2836   0.7303
  1.6576        0
  1.4155        0
  1.6576        0
  1.2836   0.7303
     0        1.4967
>> sum(b) =   31   30
>> sum(E) =   7.2980   7.0114
OSB?
>> sum(b1)=18;
>> sum (b2) = 43
```

rate adaptive

```
>> [b1 b2] %(osb)
   0   9
   0   9
   0   8
   9   0
   0   0
   9   0
   0   8
   0   9
```

```
>> [S1 S2] = %(osb)
     0   1.2775
     0   1.4967
     0   1.2750
1.6576      0
     0      0
1.6576      0
     0   1.2750
     0   1.4967
```

Total is **61** also, but **OSB uses less energy**.

sum(S1)
= 3.3152
sum(S2)
= 6.8209

sum(E)
= 7.2980
7.0114

- Sum is same, user 1 is better in osb.
- With more granularity, osb would be slightly better.

Energies < 8; again it's **interference Limited.**

iw.m (RA, non-integer)

```
function function [b, E] = iw(N, U, Hmag, No, Ex, gap, mode, b_target,cb)

  Calculates data rates of M users and corresponding bit distributions and
Energy distributions
     using iterative waterfilling.

  Inputs
  ------
  N: number of sub-channels
  U: number of users
  Hmag: squared channel transfer and crosstalk matrix (N x U x U matrix)
         Hmag(n,i,j) is the crosstalk transfer function from loop i to j
at the nth bin.
  No: noise power spectrum per tone (N x U)
  Ex: signal energy/SYMBOL
  mask: PSD mask - largest value N x U
  gap: gap in dB
  mode: (U x 1 vector) each value is one of the followings
        0 - rate adaptive
        1 - fixed margin (power minimization)
        2 - margin adaptive
  b_target: target bits on 1 DMT symbol for modes 1 and 2
  bitcap: maximum possible number of bits at each frequency bin
  cb =1 for cplx BB and =2 for real BB

  Outputs
  -------
  b: bit distribution (N x U matrix)
  E: energy distribution (N x U matrix)

  Remarks
  Iterate waterfiling for each user 10 times
  Youngjae(Sean) Kim - modified J. Cioffi, April 2024
```

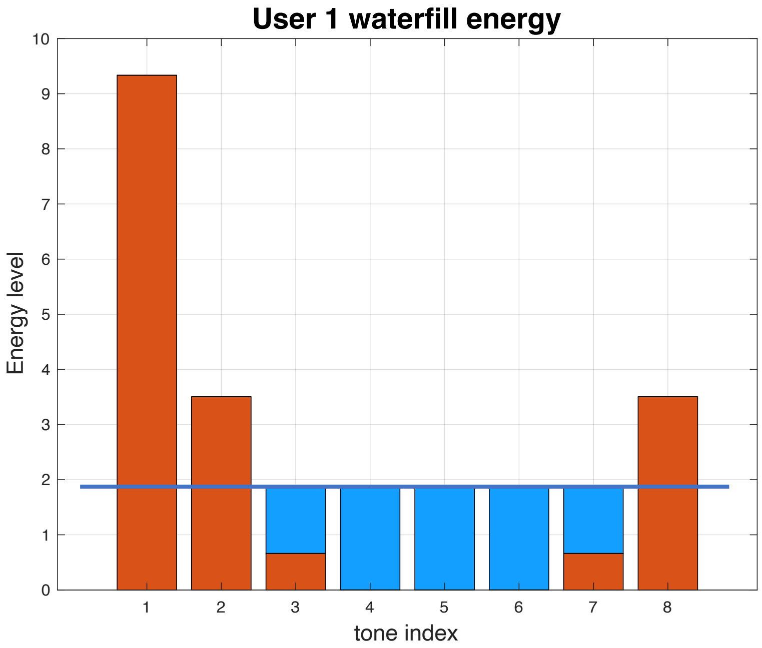
```
>> gap = 0 % dB for this program
>> [b, E,K,NF] = iw(8, 2, H3.*conj(H3), Noise, [1 1], gap, [0 ; 0], [30 24],1)
i =   9 (9 iterations)
b =  0       9.5513
     0       9.3229
  1.4931   1.4529
  9.1689       0
  9.3967       0
  9.1689       0
  1.4931   1.4529
     0       9.3229
E =  0       1.8732
     0       1.8728
  1.2039   1.1906
  1.8639       0
  1.8644       0
  1.8639       0
  1.2039   1.1906
     0       1.8728
K =   1.8672   1.8757
NF =
 188.3248   0.0025
  3.5048   0.0029
  0.6633   0.6852
  0.0032   6.1326
  0.0028     Inf
  0.0032   6.1326
  0.6633   0.6852
  3.5048   0.0029
>> sum(b) =  30.7209  31.1029
>> sum(E) =   8.0000   8.0000
>> sum(b,'all') =  61.8237
>> sum([S1  S2],1) =   3.0577   2.1313
```

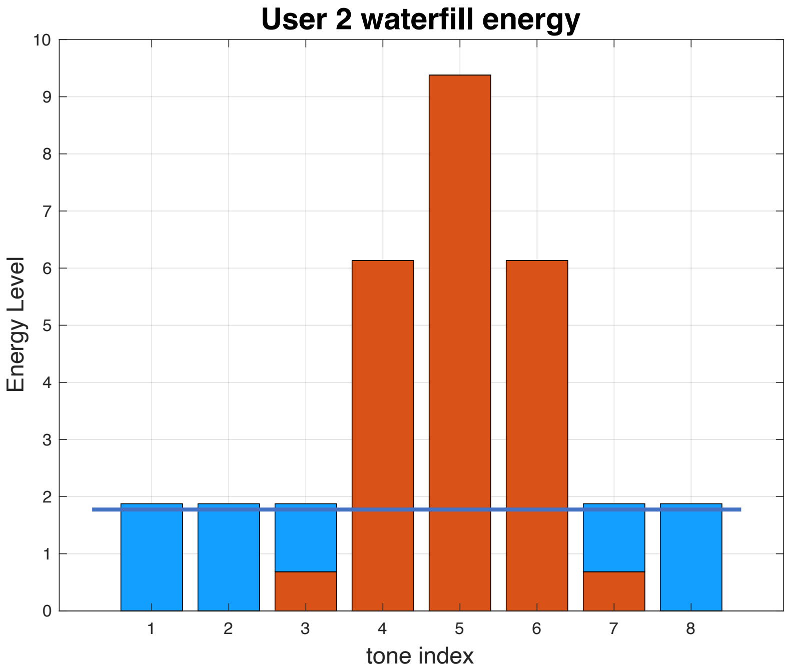
```
>> [b1 b2] %(osb)
   0   9
   0   9
   0   8
   9   0
   0   0
   9   0
   0   8
   0   9
```

```
>> [S1 S2] = %(osb)
      0   1.2775
      0   1.4967
      0   1.2750
 1.6576      0
      0      0
 1.6576      0
      0   1.2750
      0   1.4967
```

Matches bit rate now with fractional bits, and full energy

IW Energies (water)





- Water levels are not exactly the same (just close this time)

Command for this plot? bar([1:8],min([NF(:,1)' ; E(:,1)'],K(1)*5),'stacked')

MIMO case – IWF.m

```
% function [ Rxx, b, bsum ] = IWF(Eu, H, Lxu, Lyu, Rnn, cb)
%
% Iterative water-filling for the full MIMO case, specifically rate
maximization.
%
% Inputs:
% Eu U x 1 energy/space-subsymbol vector. U = # of users.
% any (N/N+nu) scaling should occur BEFORE input to this program.
% H The FREQUENCY-DOMAIN Ly x sum(Lx(u)) x N MIMO channel for all users.
% N is determined from size(H) where N = # used tones H(i,j,n) is from
% loop j to loop i on tone n.
% Lxu 1xU vector of each transmitter's number of antennas
% Lyu 1xU vector of each receiver's nunmber of antennas.
% Rnn The Ly x Ly x N noise-autocorrelation tensor (last index is per tone)
% non-xtalk background noise, presumed to be block diagonal. Ly=sum(Lyu)
% cb cb = 1 for complex, cb=2 for real baseband
% cb=2 corresponds to a frequency range at an sampling rate 1/T' of
% [0, 1/2T'] while with cb=1, it is [0, 1/T']. The Rnn entered for
% these two situations may differ, depending on how H is computed.
%
% Outputs:
% Rxx An per-tone Lx x Lx x n matrix diagonal psd (per tone)matrix Lx=sum(Lxu)
% with IWF input autocorrelation user's transmitter.
% sum trace(Rxx (u)) over tones and spatial dimensions equal the Eu(u)
% b is the bit distribution
% bsum the rate sum.
```

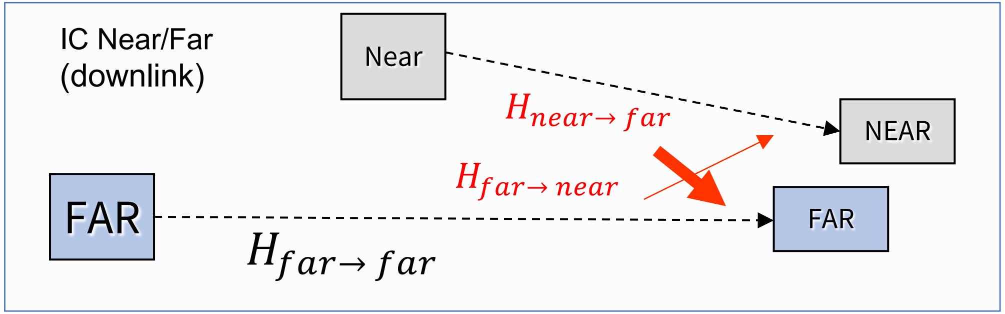
- Accepts arbitrary number of transmit and receive antennas.
  - IWF has per-user vector coding with all other users as interference when Lx > 1.
- For the MAC, only 1 GDFE, allowing coordination
  - Noticeably higher data rates
- For the IC with only scalar receivers for each IWF system, the data rates are lower than MAC or single user.

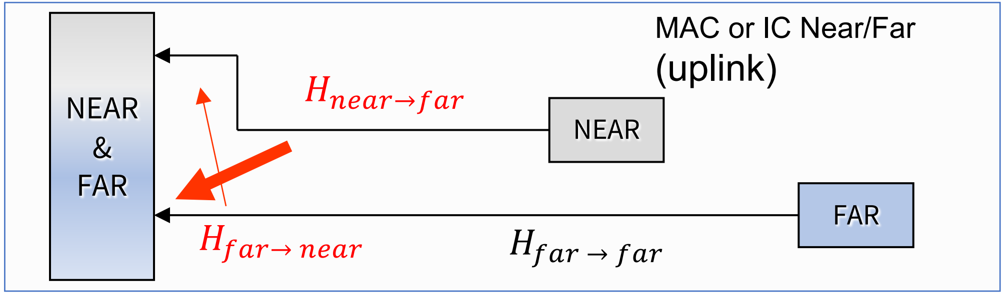
```
H4=permute(H3, [3 2 1]);
Rnn=zeros(2,2,8);
for n=1:8
Rnn(:,:,n)=eye(2);
end
 Lxu= [ 1 1];
Lyu = [1 1];
[Rxx, b, bsum] = IWF([8 8], H4, Lxu , Lyu, Rnn, 1)
b' =  30.7209  31.1029
bsum =  61.8237  (checks)
>> sum(Rxx,'all') =   16

>> H2 = [   4    3    2    1
            5    6    7    8];
>> Lxu =[   2    2];
>> Lyu =[   1    1];
% MAC Call:
[Rxxs, Eun, w, bun] = maxRMACMIMO(H2, [2 2], [5 6], [1 1] , 1);
Eun' =    5.0000   6.0000
w ' =      0.3881   0.3275
bun =  7.6484   5.8337
Rxx, b, bsum] = IWF([5 6], H2, Lxu , Lyu, Rnn, 1)
j =    3
Rxx =
   3.2000   2.4000        0        0
   2.4000   1.8000        0        0
        0        0   2.6018   2.9735
        0        0   2.9735   3.3982
b =  2.5063   1.7397
bsum =   4.2460
```

Near Far Situations

- The other(s') transmitter(s) is (are) too close.

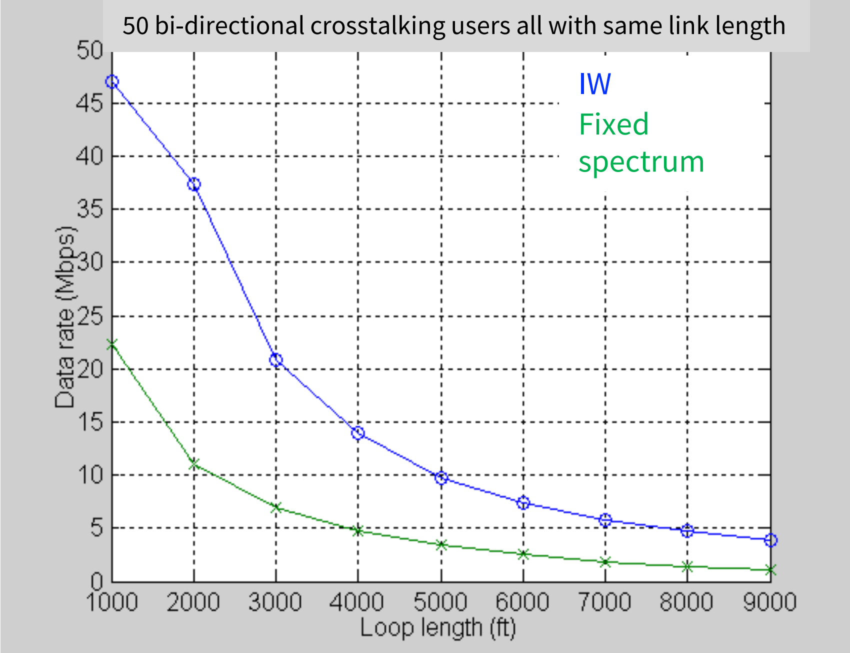




Dynamic Spectrum Sharing, 25 bi-directional users

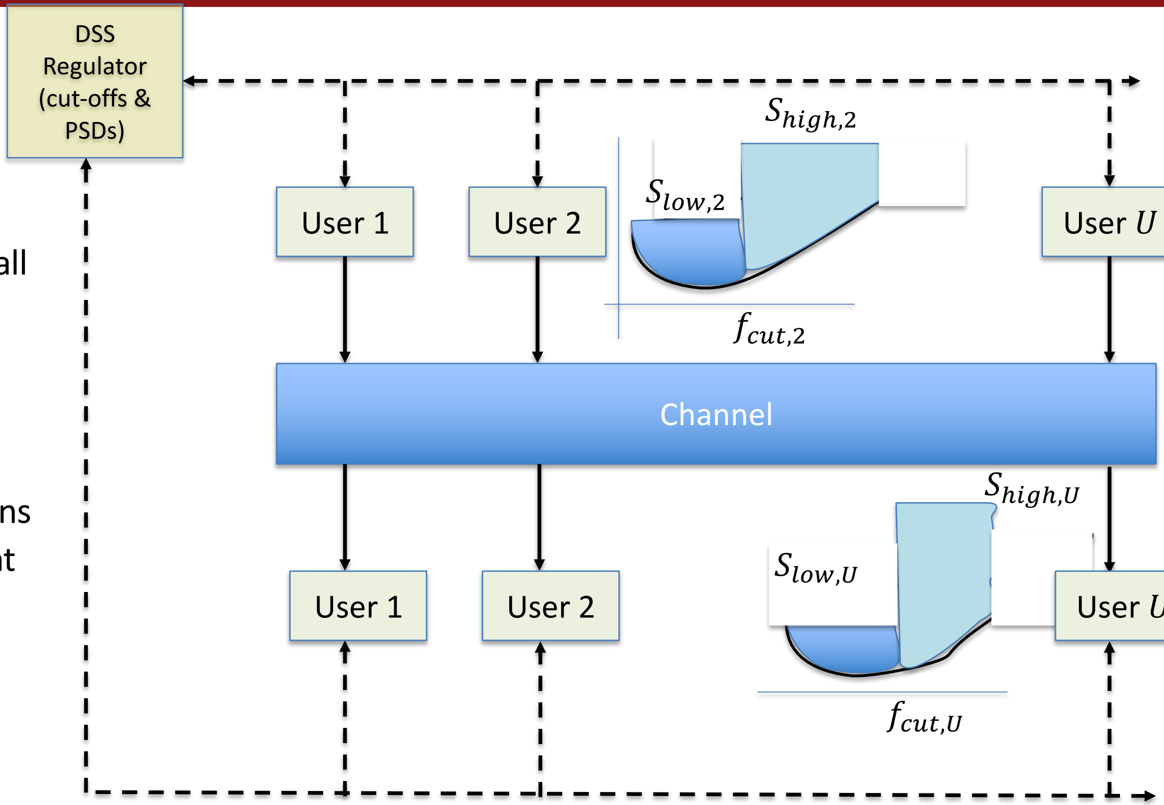
- All links the same length, 25 users.
- Both directions allowed to use same spectrum.
- There is large near-far from “echoed crosstalk.”
- No cancellation of crosstalk occurs.
- Previous state of art had been a fixed transmitter spectrum on all links.

- IW instead used to decide all users
- Roughly double the range or rate possible



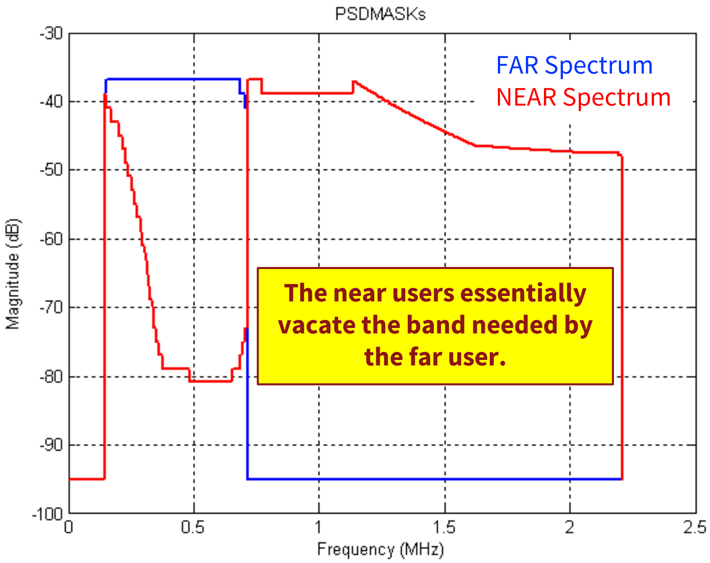
Multilevel Waterfilling

- Adapt the duplexing dynamically.
- DSS (dynamic spectrum sharing)
- Regulator has digital twin
  - Only knows each users SNRs with all others as interference
- Regulator distributes $f_{cut,u}$.
- Each user does IWF, but also:
  - Moves bits above $f_{cut,u}$
  - Lowest upper-band energy positions
  - Until power or PSD mask constraint met.

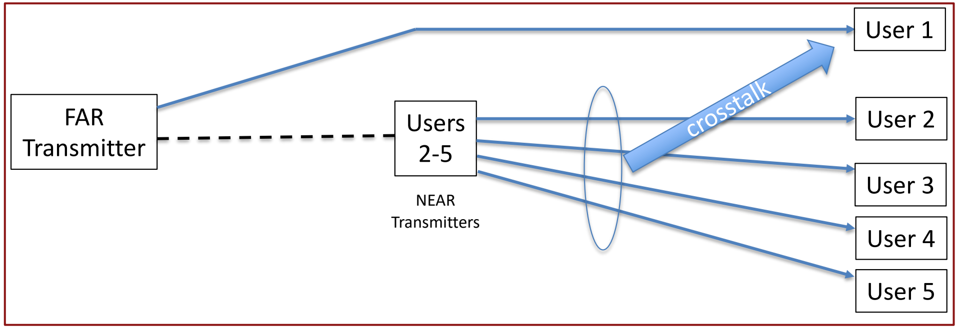


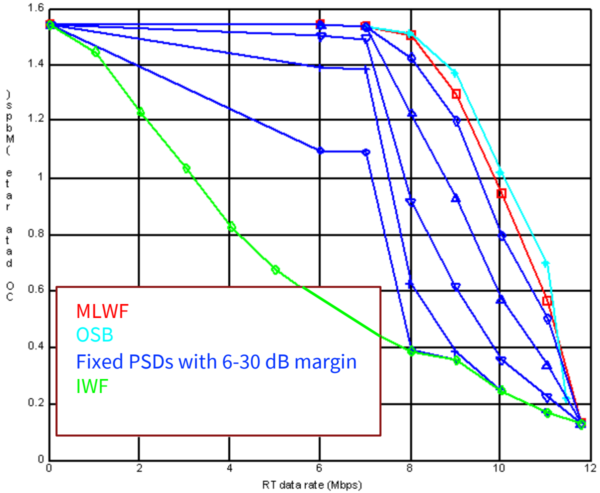
Near-Far Example with MLWF

- Essentially two users are here (Near and Far).



- Results in 3 bands, (the lowest frequencies are just not used, so there is no sharp transition, just they are not part of the exercise.





MLWF Matches OSB

STANFORD

End Lecture 16