STANFORD

Lecture 18

LIC and Future/ML Topics

June 7, 2023

JOHN M. CIOFFI

Hitachi Professor Emeritus of Engineering

Instructor EE392AA – Spring 2023

Announcements & Agenda

- Announcements
  - PS7 extended to June 12
  - Project due Tuesday, June 13 if you need to graduate on June 18
  - Will finish quarterly grades June 14 (if you want "I" let me know)

- Agenda
  - Recall CIC (OSB and minPIC)
  - The LIC and Iterative Water-Filling Methods
  - Research & Machine-Learning/AI Challenges

EE379 A , B (=C, 392AA)
2023-4 Winter, Spring

Student RA/TA interest?
(see JC)

Recall OSB & minPIC

OSB Refresher

- OSB
  - All xtalk is noise
  - NP-hard solution, but converges for min-energy-sum
  - Potential project
  - Probably not usable, but provides some guidance for LIC

$$\max_{\{R_{\boldsymbol{xx}}(u,n)\}} \quad \sum_{u=1}^{U} w_u \cdot \mathcal{E}_u$$
$$ST: \quad 0 \le \sum_n \text{trace}\left\{R_{\boldsymbol{xx}}(u,n)\right\} \le \mathcal{E}_{u,max} \;\; u = 1, ..., U$$

- Relate $\boldsymbol{b}$ to $R_{\boldsymbol{xx}}(u,n)$.
  - No crosstalk cancellation
  - Non convexity enters here

$$b_u = \sum_n \log_2 \frac{\left| H_{uu,n} \cdot R_{\boldsymbol{xx}}(u,n) \cdot H^*_{uu,n} + \mathcal{R}_{noise}(u,n) \right|}{\left| \mathcal{R}_{noise}(u,n) \right|}$$

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

Quantize to M energy levels
M^U cacluations (per tone)

minPIC = more "optimum"

- minPIC concept allows for each receiver u to cancel $i \in \mathcal{D}_u(\boldsymbol{\Pi}, p_{\boldsymbol{xy}}, \boldsymbol{b})$; the decodable set
- Order has been restored 😀
- The optimization is

$$\min_{\{R_{\boldsymbol{xx}}(u)\}} \quad \sum_{u=1}^{U} w_u \cdot \underbrace{\text{trace}\left\{R_{\boldsymbol{xx}}(u)\right\}}_{\mathcal{E}_u}$$

$$ST: \quad b_{i,u} \geq \left\{ \begin{array}{ll} b_{min,i} & \pi_u(i) \leq \pi_u(u) \\ 0 & \pi_u(i) > \pi_u(u) \end{array} \right\} \triangleq \boxed{b_{min(\pi_u),u,i}}$$

$$R_{\boldsymbol{xx}}(u) \succeq \mathbf{0} \quad .$$

- $\boldsymbol{\theta} \rightarrow \boldsymbol{\Theta}$, which now has U^2 terms, U for each receiver → determines the $\boldsymbol{\Pi}$
- This is actually convex already (like minPMAC)
- Implement GDFE at each receiver for final order ($\boldsymbol{\Theta}$ → $\boldsymbol{\Pi}$)

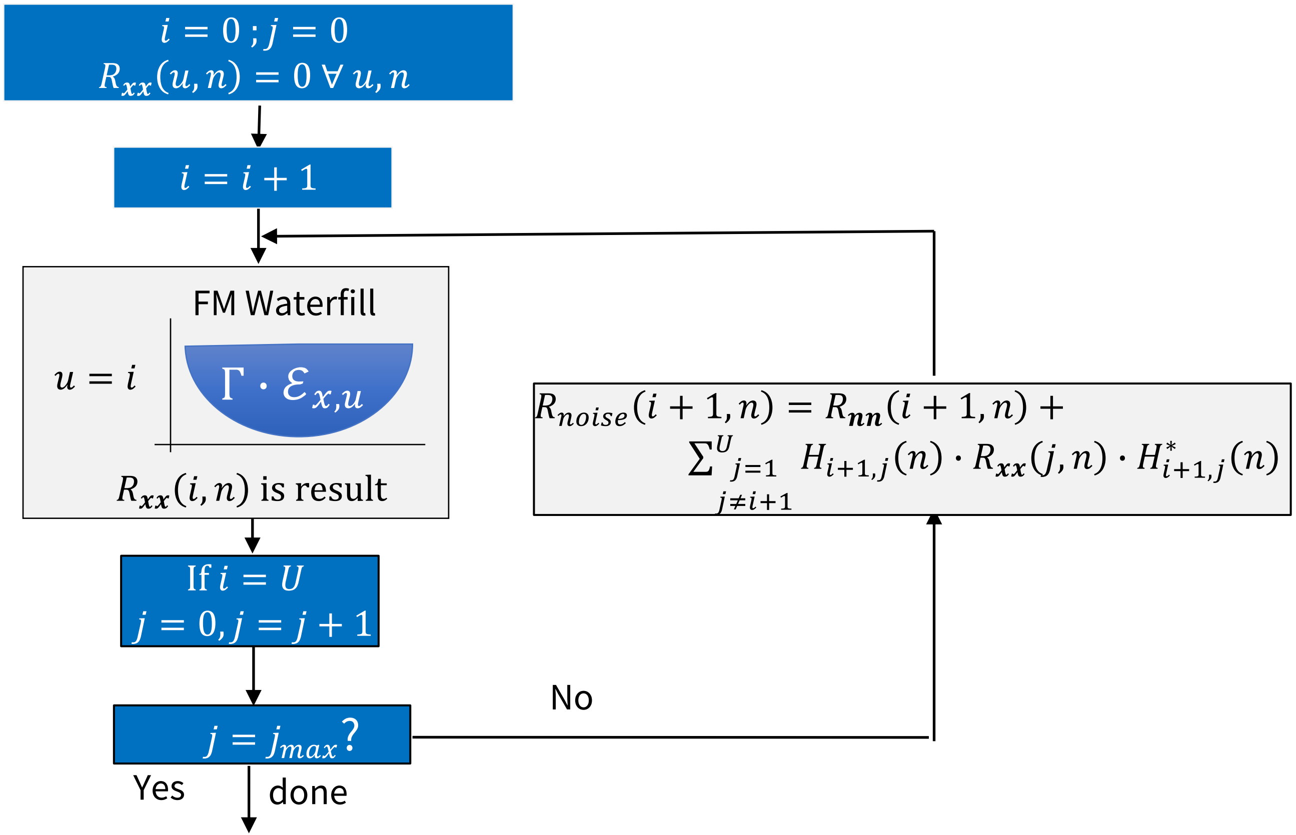
LIC Design

Iterative WF borrows MAC's SWF for the IC

- Rate-sum partial derivatives

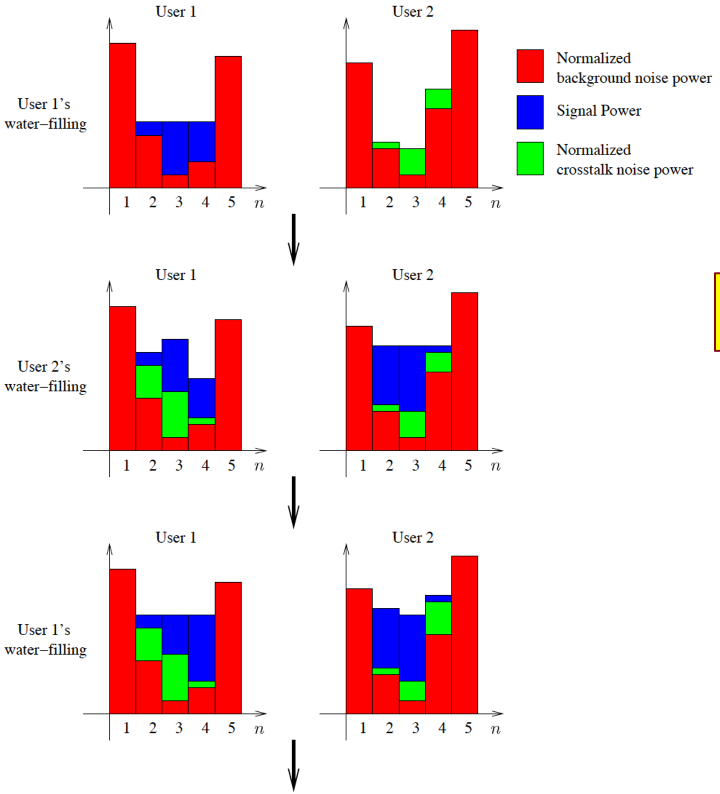
$$\lambda_u = \mathcal{E}_{u,n} + \frac{\Gamma\sigma_n^2 + \sum_{i\neq u} |H_{ui}|^2 \cdot \mathcal{E}_{i,n}}{|H_{uu,n}|^2}$$

- Converges in practice
  - Mild channel conditions (Leshem)



IW Illustrated for the IC

- Each user "reacts" to others



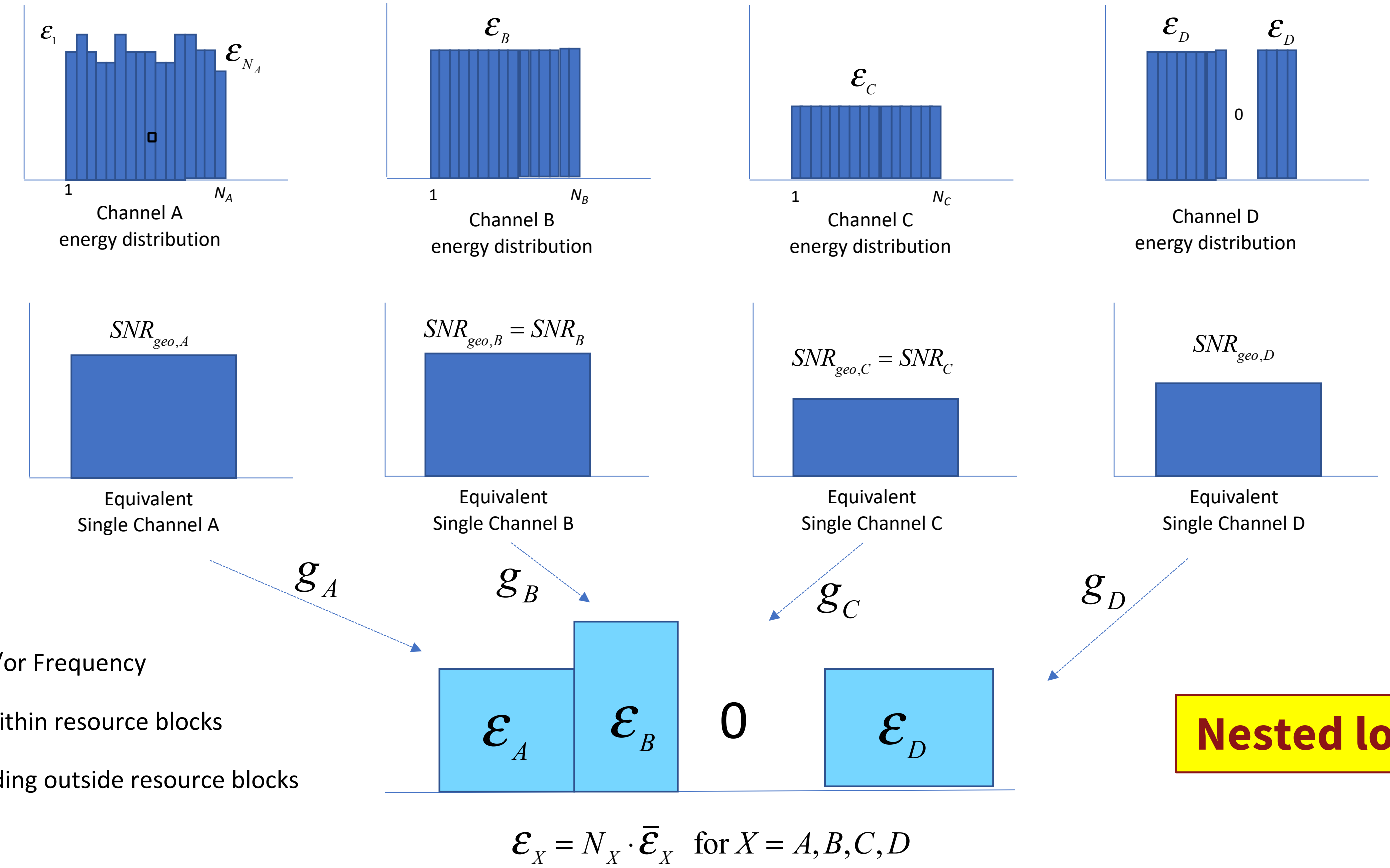
- Others sense the new xtalk

- Eventually converges

"Nash Equilibrium"

Repeat until converged

Wireless Potential Use (Resource Blocks)

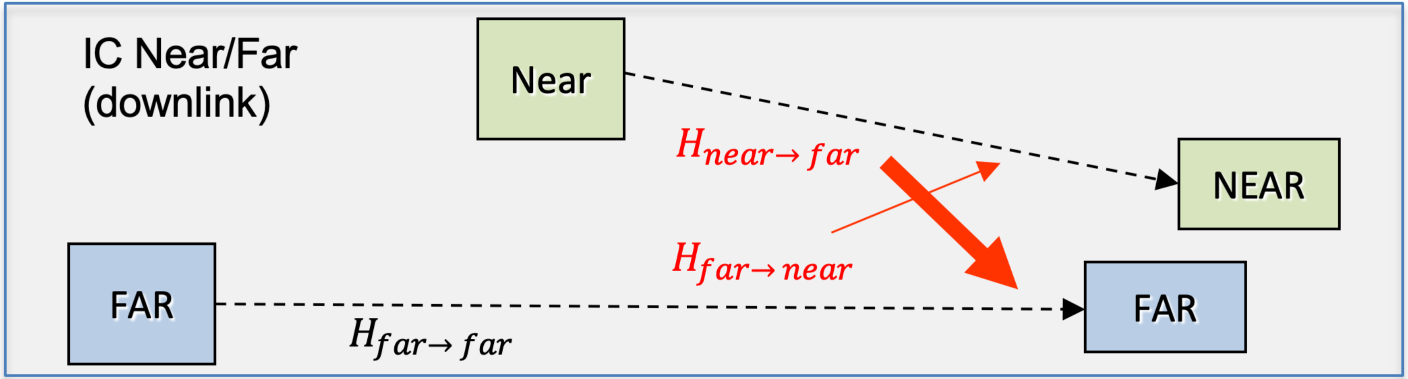


- Space and/or Frequency
- C-OFDM within resource blocks
- Vector-Coding outside resource blocks

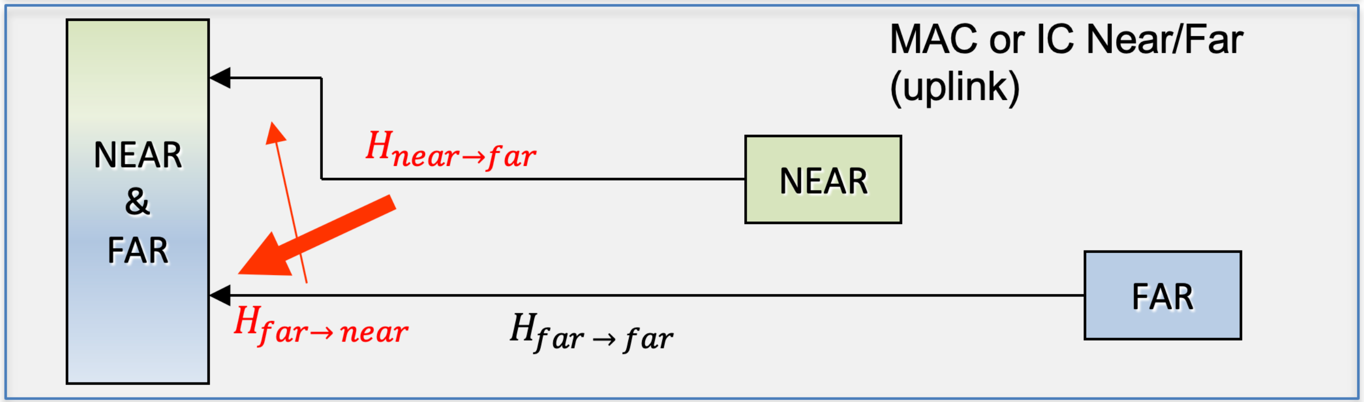
Nested loading

Near-Far Example

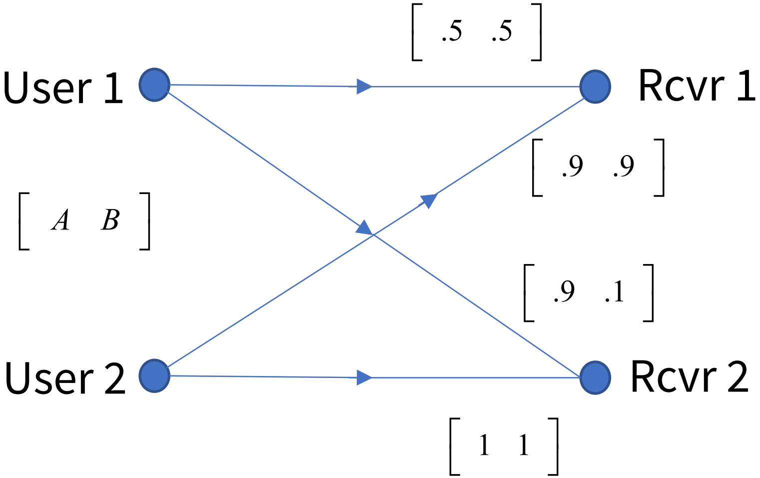
- Downlink has another transmitter for another IC user closer (the "near" user)



- Uplink has another transmitter for another IC user closer (the "near" user)



Example IC with IWF



$$\begin{bmatrix} y_{1A} \\ y_{1B} \end{bmatrix} = .5 \cdot \begin{bmatrix} x_{1A} \\ x_{1B} \end{bmatrix} + .9 \cdot \begin{bmatrix} x_{2A} \\ x_{2B} \end{bmatrix} + \begin{bmatrix} n_{1A} \\ n_{1B} \end{bmatrix}$$

$$\begin{bmatrix} y_{2A} \\ y_{2B} \end{bmatrix} = \begin{bmatrix} x_{2A} \\ x_{2B} \end{bmatrix} + [.9 \quad .1] \cdot \begin{bmatrix} x_{1A} \\ x_{1B} \end{bmatrix} + \begin{bmatrix} n_{1A} \\ n_{1B} \end{bmatrix}$$

$\boxed{\sigma_1^2 = \sigma_2^2 = 0.1 \text{ (noises independent)}}$

$\Rightarrow$ Energy 1A = 2.0 | Energy 2B = 2.0

- Which receiver has near-far issue?
  - RCVR 1 in both bands
- Who is near user?
  - User 2

Tabular Tracking of IW

- User 2 reacts to user 1 crosstalk
- User 1 then counter acts
- Further reduction of energy on user 2 band B
- Converges in 2 cycles
  - Solution looks FDM
- This is better than equal energy on both users in both bands, try yourself.

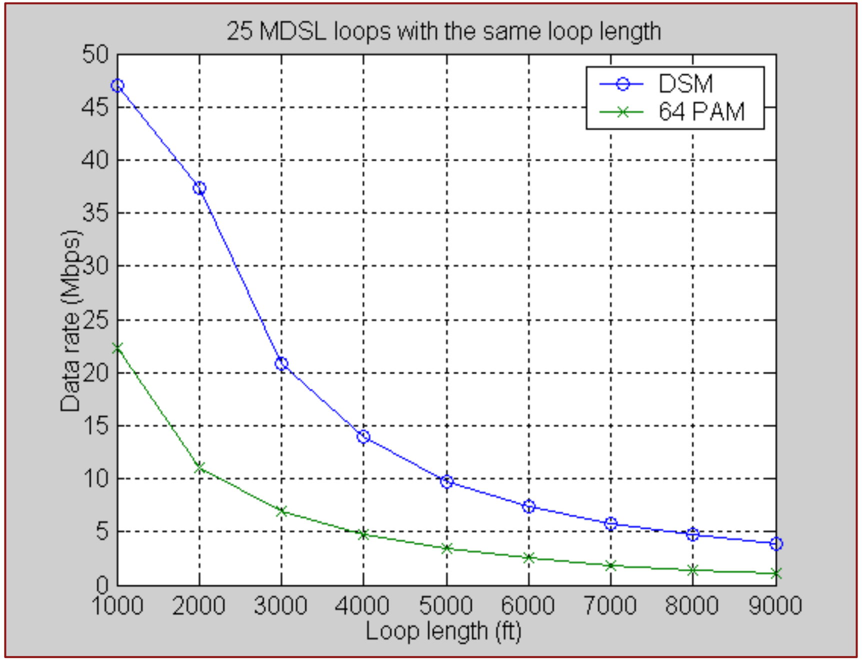
SIMPLE IW EXAMPLE

	Band A	Band B
User 1	$\varepsilon_{1A}=1$	$\varepsilon_{1B}=1$
User 2	$\frac{1}{g_{2A}}=.1+(.9)^2=.91$	$\frac{1}{g_{2B}}=.1+(.1)^2=.11$
	$\varepsilon_{2A}+.91=\varepsilon_{2B}+.11$ $\varepsilon_{2A}+\varepsilon_{2B}=2$ $\varepsilon_{2A}=.6 \quad \varepsilon_{2B}=1.4$	
User 1	$\frac{1}{g_{1A}}=\frac{.1+.6\cdot(.9)^2}{(.5)^2}=2.344$	$\frac{1}{g_{1B}}=\frac{.1+1.4\cdot(.9)^2}{(.5)^2}=4.936$
	$\varepsilon_{1A}+2.344=\varepsilon_{1B}+4.936$ $\varepsilon_{1A}+\varepsilon_{1B}=2$ $\varepsilon_{1A}=2 \quad \varepsilon_{2B}=0$	
User 2	$\frac{1}{g_{2A}}=.1+2\cdot(.9)^2=1.72$	$\frac{1}{g_{2B}}=.1+0\cdot(.1)^2=.1$
	$\varepsilon_{2A}+1.72=\varepsilon_{2B}+.1$ $\varepsilon_{2A}+\varepsilon_{2B}=2$ $\varepsilon_{2A}=.19 \quad \varepsilon_{2B}=1.81$	
User 1	Remains $\varepsilon_{1A}=2 \quad \varepsilon_{2B}=0$ → IW has converged	
Data rates User 1	$\log_2(1+2/2.344)=.89$	0
Total User 1	.89 bits	
Data rates User 2	$\log_2(1+.19/1.72)=.15$	$\log_2(1+1.81/.1)=4.26$
Total User 2	4.4	
Rate Sum	5.29 bits	

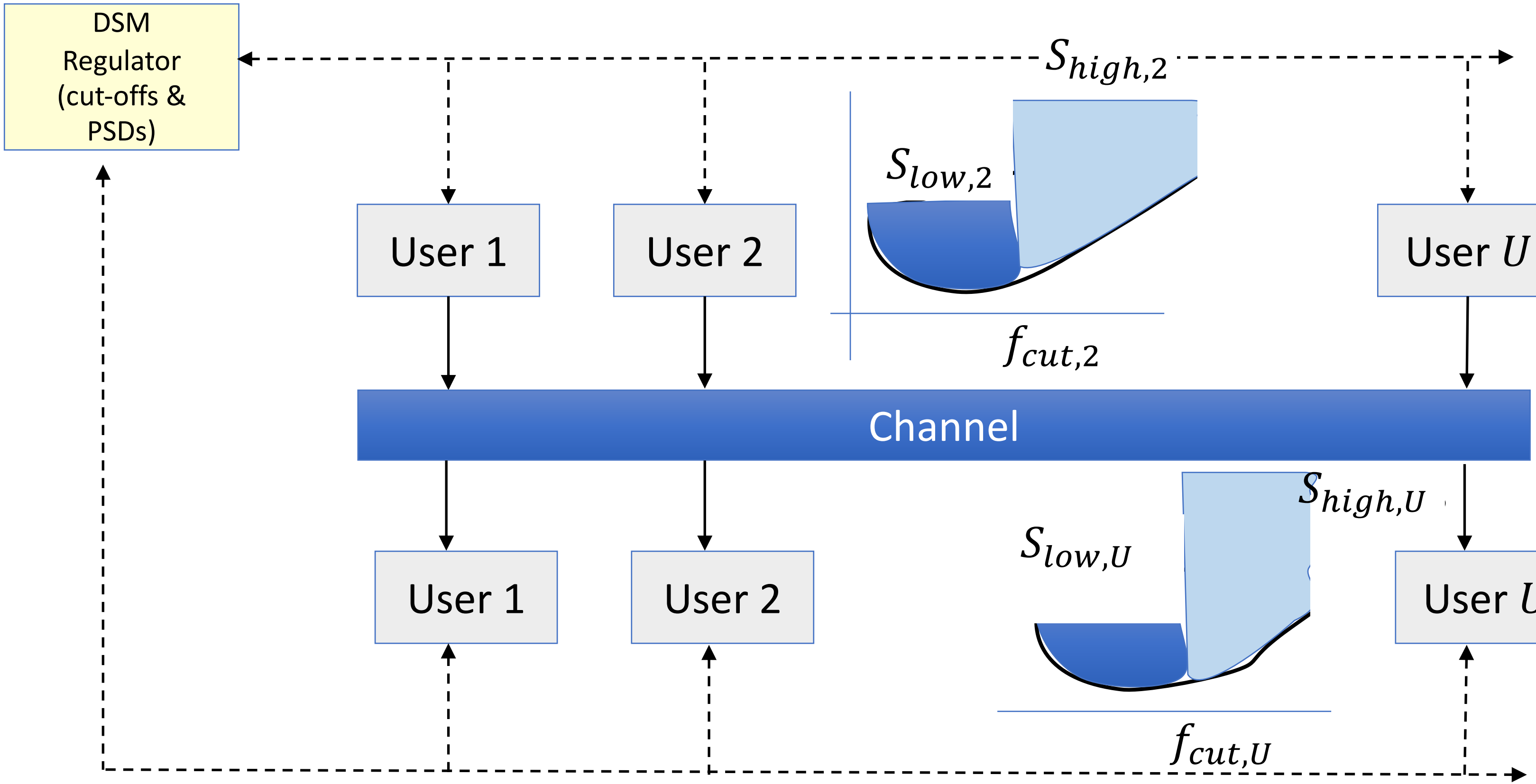
More sophisticated situation

- 25 bi-directional users (so really 50 users if they all share same band)

- Turn on MA WF for them all and let them run versus fixed spectrum with b=6 bits/Hz
  - Which was state of art prior to IW



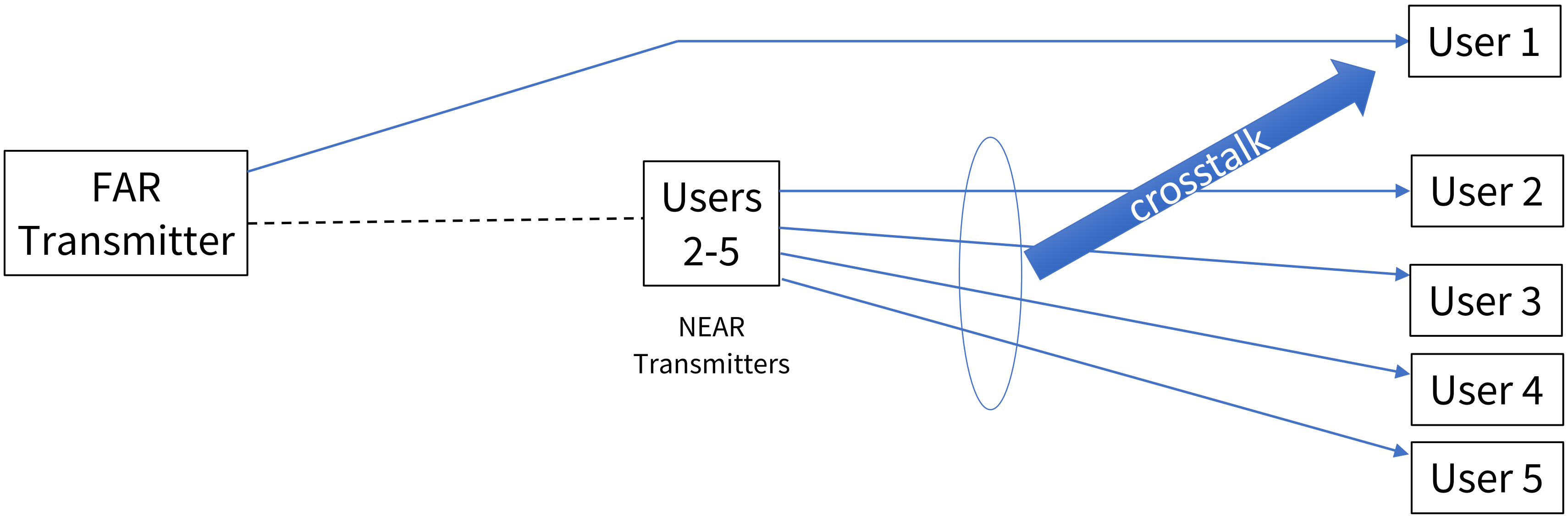
Multi-Level Water-fill



- Runs IW, but with different water-levels (a bit like SWF) – must find cut-off frequency(ies)
- Very low complexity (same as IW), but central control distributes (learns) cut-off frequency

Near-Far Example

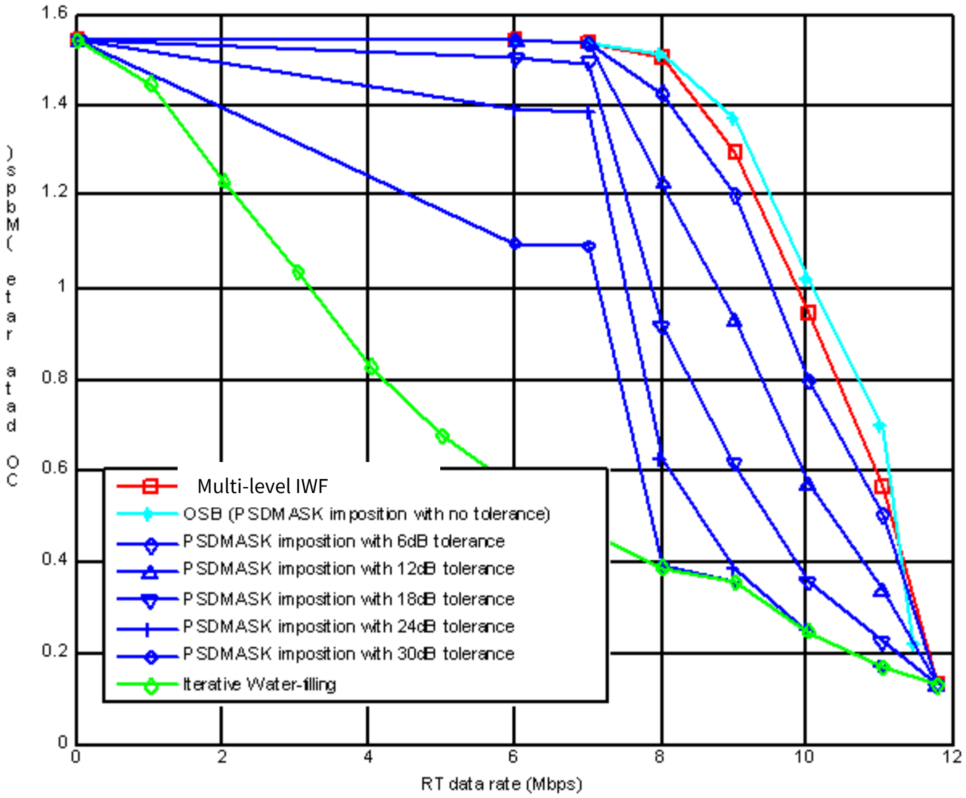
Wi-Fi's "pods" – if on same/overlapping channels



- Problem with RIS (reflective intelligent surfaces) for adjacent bands of RIS
- Can occur in wireline also – "remote terminals" or "distribution units"

Achievable Region Comparison

- IW better than fixed, but not so good
- 25 users with ML IW and with OSB
  - See upper right
  - Ignore the other curves
- ML IW is pretty close to optimum
- Open question – could we use it for the Rxx step in minPIC, with only uncancelled users instead of Hessians or CVX?

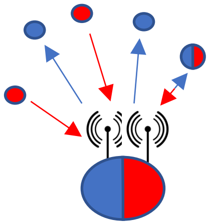


Projects: Need MLIW.m and IW.m

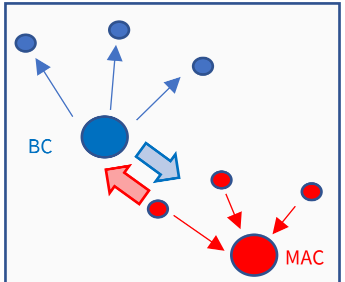
ML & Challenges

Nesting

- Use minPmac/bc on node channels



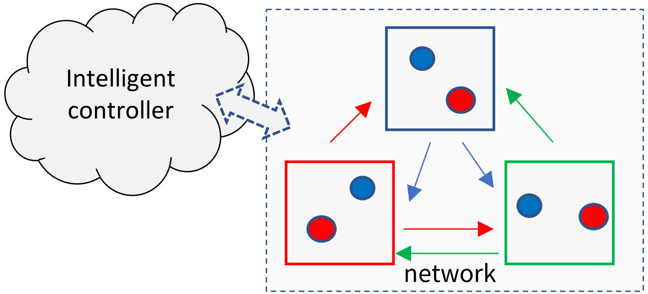
a) radio node edge
(base station or
Access point)



b) two subnetworks
Down (BC) and up (MAC)
IC network between them

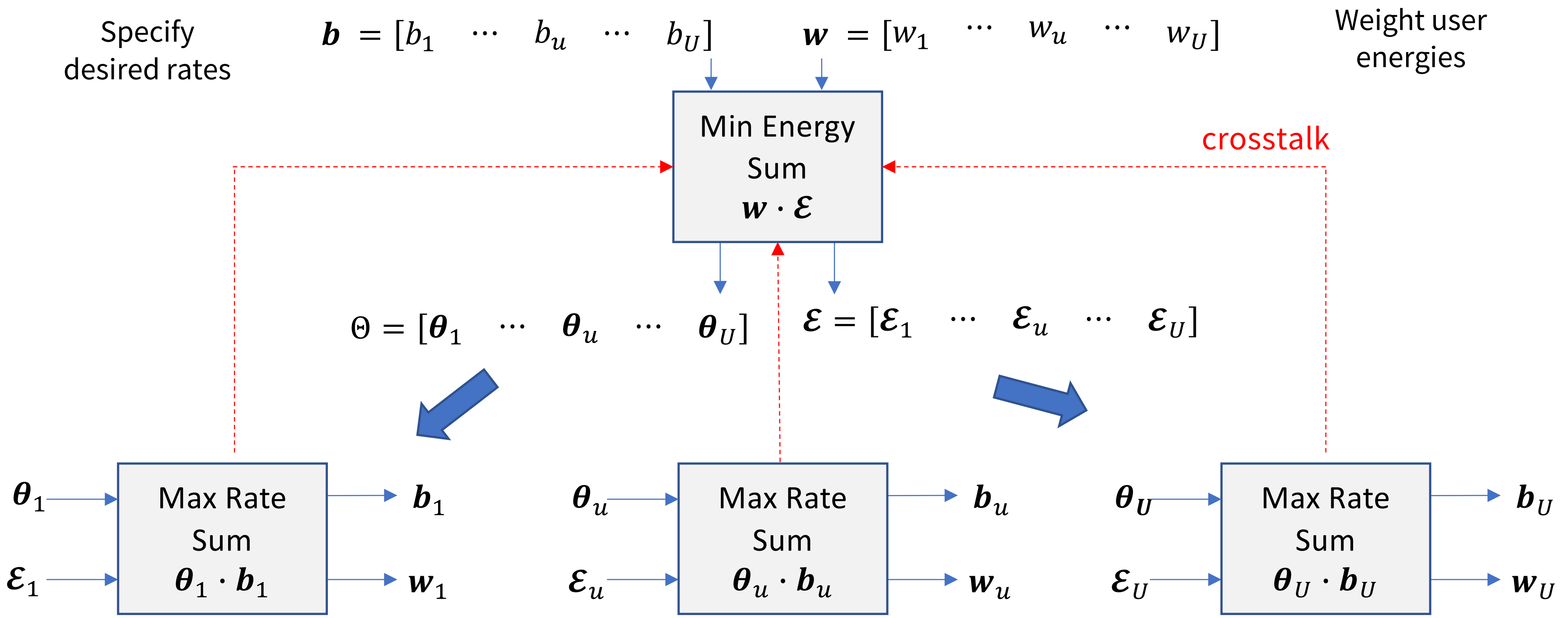
- Use ML Water-Fill between nodes

- Efficient Algorithms?
  - How/where to update?



c) three nested IC: (BC, MAC)
Like those in b), nested into
3x3 IC network

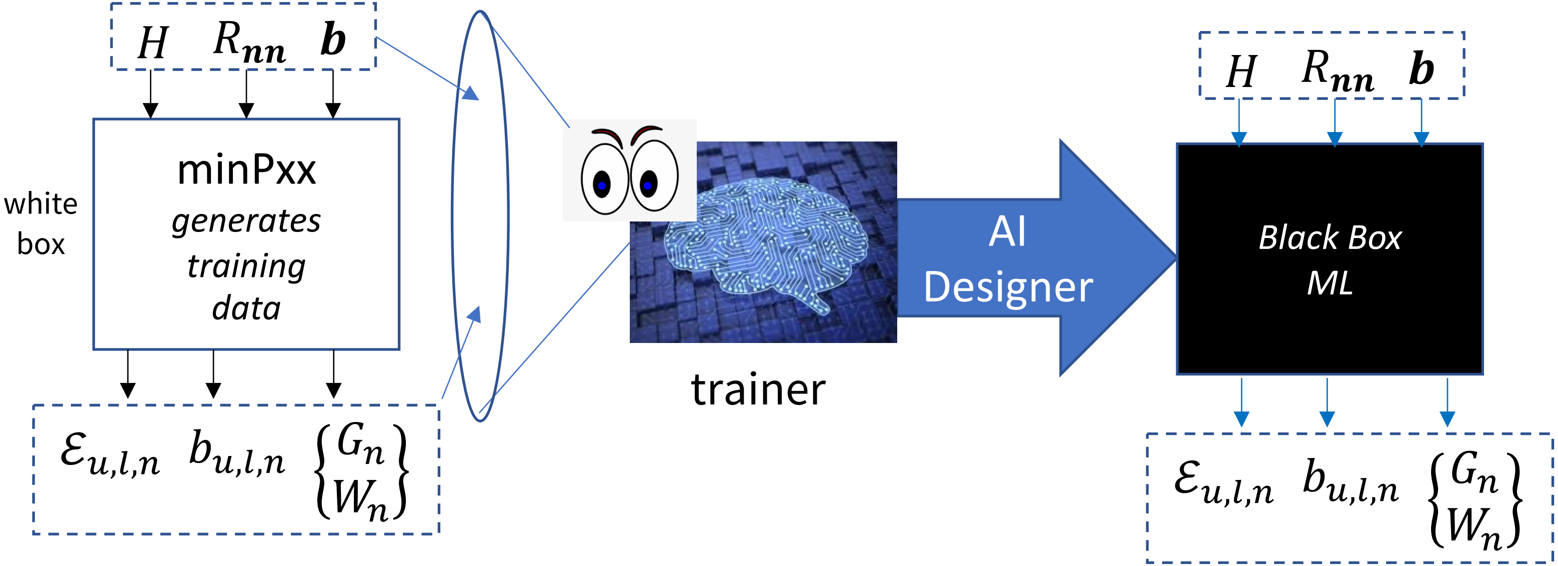
AI ("machine learned") Approximations?



- Each of these "boxes" (subnetworks) can be intense calculation
- The overall recursive cycling is actually then more intense

Machine learned "minPxx"

- minPMAC (and minPIC) optimize, but may have long run-times and numerical issues
  - They accept channel+noise and data rates (and maybe energy in admxx)

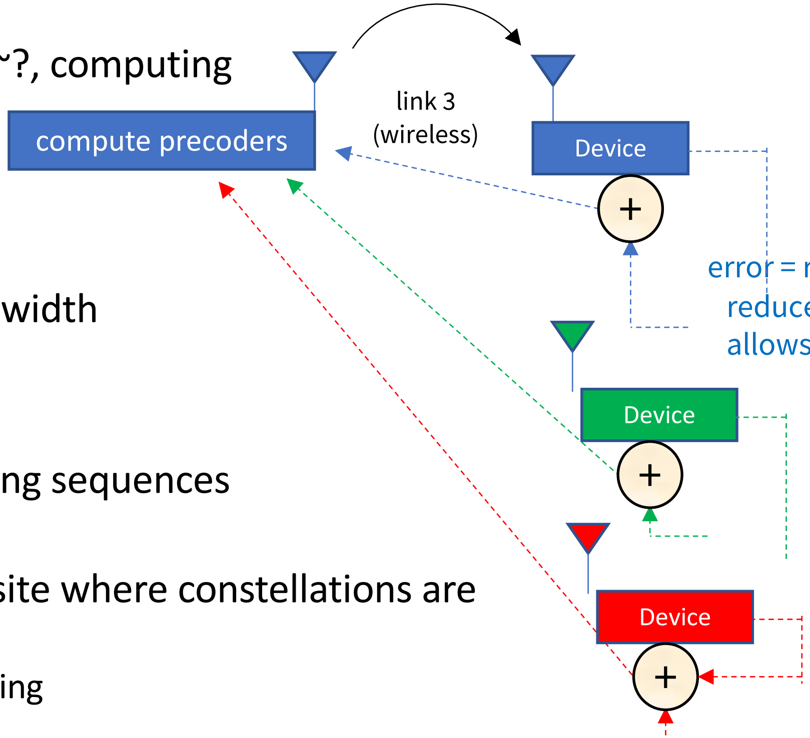


probably reccurrent and/or convolutional neural net with a few layers
With also Reinforcement-learning
Over time

- Extended to nesting, complex networks

Where, and from what, to compute precoders?

- Receivers estimate channels today ~?, computing
  - filters/matrices
  - bit distributions
  - energy distributions
- This generates large overhead bandwidth
  - (even with ”indexing” schemes)
- Return only errors for pilots/sounding sequences
- Compute instead at edge or at the site where constellations are generated
  - Need error signals from pilots/sounding

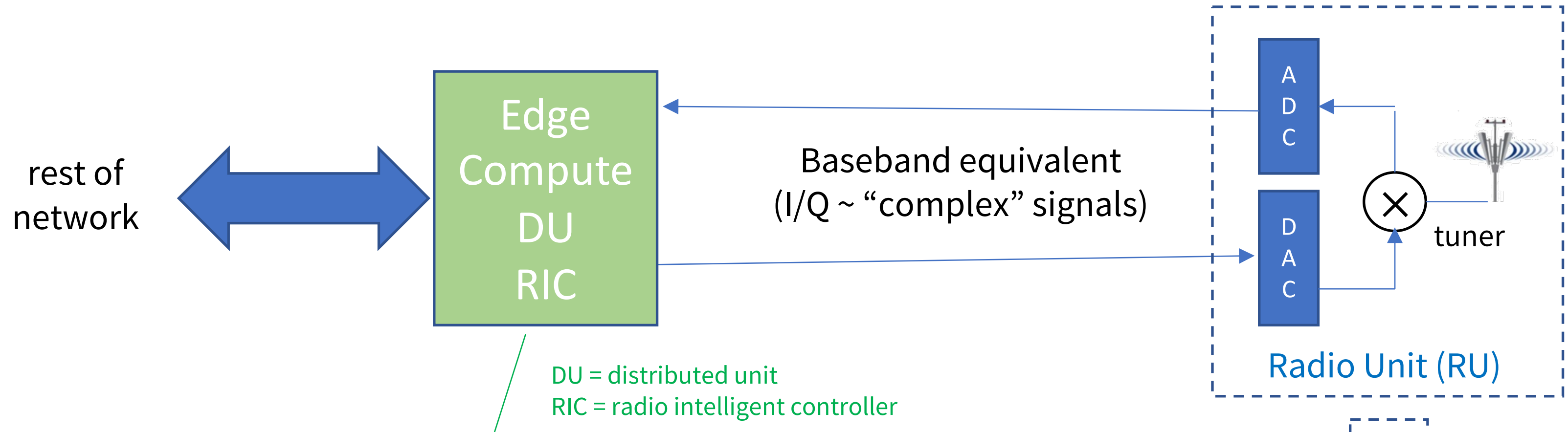


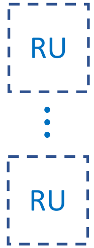
compute only errors here

error = received pilot - known pilot
reduces feedback bandwith
allows precoder calculation at edge

Algorithms (ML/AI) based on digital twin of this to update precoders ?

O-RAN/Xhaul split 7.2 (over) simplified



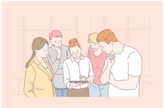


- Share servers for calculations when needed
- Coordinate the messages/signals , space, time, frequency
- This where the AI will become really helpful

Correlate Design choices with User Reaction

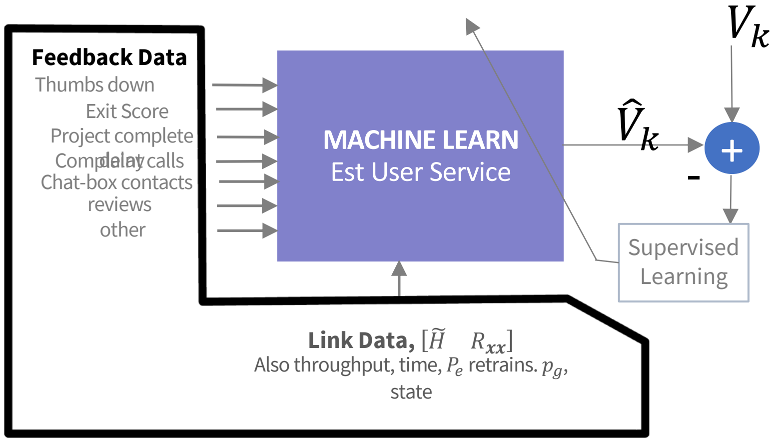
- Thumbs down 
- Exit score

- Calls to IT/ISP 
- Group success rate



- Repair/Intervention counts 

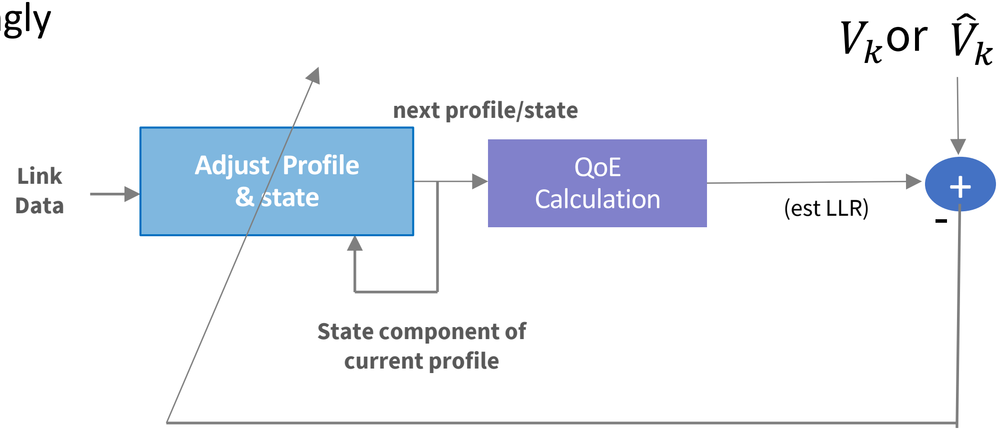
Diagnostics & Analytics
Learn the reward function V
employee feedback and link data



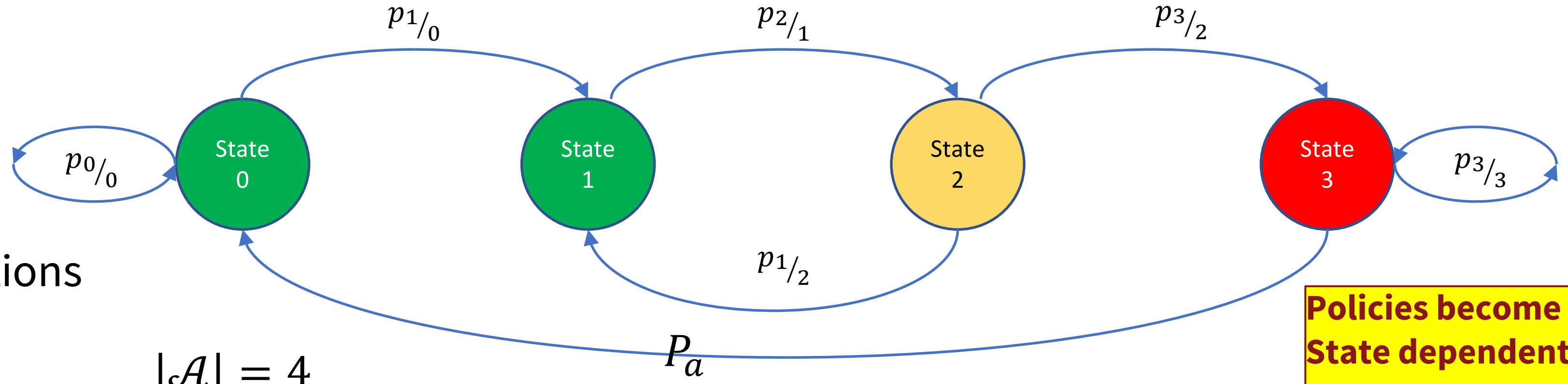
Optimize QoE (value)

- Use analytics to derive
  - Priorities (orders, weights)
- Optimize accordingly

Optimization (management)
Objective is to improve service to “green”



A Network State Machine: Reinforcement Learning



Variations
time
load

$|\mathcal{A}| = 4$

Policies become State dependent

Learn the states and P_a

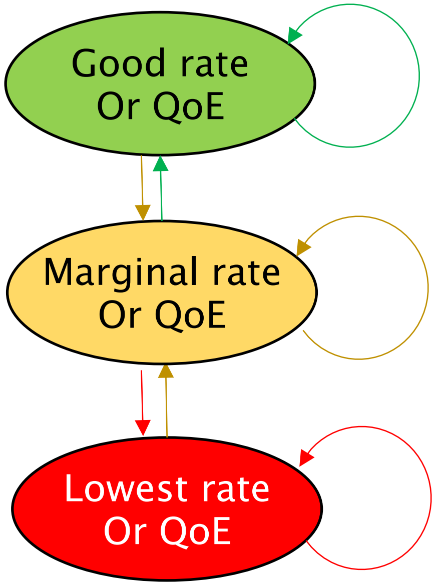
- Network user/link may be in a state or profile
  - Some are ok (user happy or green) ; amber on the edge ;
  - Red – very likely unhappy
- Markov (state-machine) models
- Learn the profile, apply appropriate design for each state
  - Objective is move to green state with profile change

$$P_a = \begin{bmatrix} p_{3/3} & p_{3/2} & 0 & 0 \\ 0 & 0 & p_{2/1} & 0 \\ 0 & p_{1/2} & 0 & p_{1/0} \\ p_{0/3} & 0 & 0 & p_{0/0} \end{bmatrix}$$

$\pi = P_a \cdot \pi$ Markov (stationary) distribution

A Network State Machine: Reinforcement Learning

- Determines Next Action (State)
  - Profile $\{R_{\boldsymbol{xx}}(u), b_u\,, [G_u \quad W_u]\}\,, u = 1, \ldots, U'$
- GYR
  - Try to get to better states
  - But this depends on cost of doing so
- Markov (state-machine) models



Can include, MCS, number of spatial streams, channel, spectrum, priority (weights $[\boldsymbol{w} \quad \boldsymbol{\theta}]$), etc

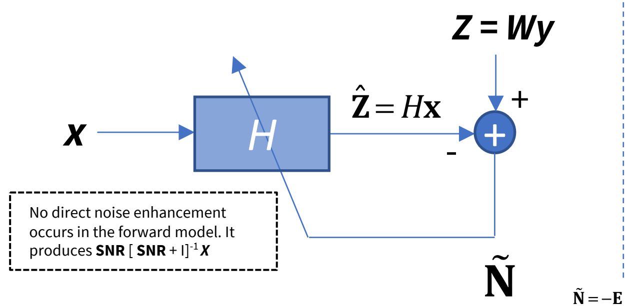
Estimating the probabilities and States

- Better on-line/real-time "fading" distributions
- While all the "Raleigh, Ricean, log-normal, angle-spread, delay-spread ……… " models create simulation environments that range through many situations, they're not specific to situation
- Each channel/user may need to estimate probability distributions for "fading/xtalk"
  - How do do this well
  - Ergodic state machines (Markov models) or slowing varying
  - Digital Twins?
  - Know the settings for each in advance?
- Then identify which state and associated pre-computed design?
  - Would this save a lot of computing energy?
- Are Pe and data-rates the right measures? → Quality of Experience (QoE)
  - Learned from user "feedback"👍 or 👎
- Reinforcement Learning? (Recurrent Neural Net as base?)

MMSE for SVD, QR, Cholesky

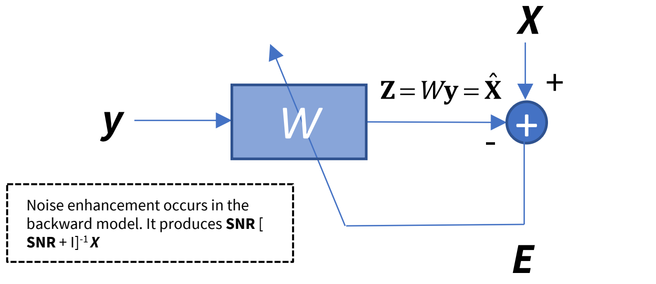
Forward Channel Model for SVD Calculation

M

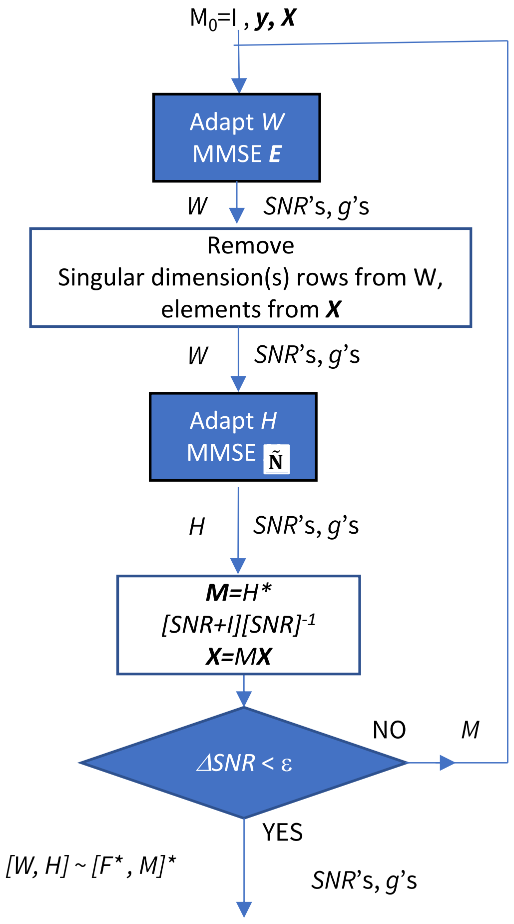


Backward Channel Model for SVD Calculation

F



$\tilde{\mathbf{N}} = -\mathbf{E}$ (when both converged)

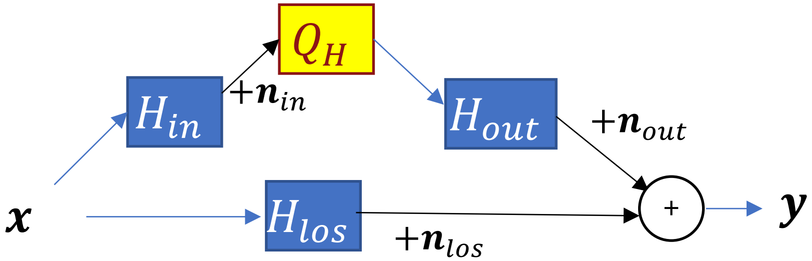


- The matrix factorizations our designs use are SVD, QR, and Cholesky
  - Each can be modeled by a MMSE program (or collections thereof)
- Better ways to compute them online efficiently – see (Section 7.2 at right) - not tested yet.
- Similar MMSE problems set up with G and W for MMSE-based QR

Reflective Intelligent Surfaces (RIS)

Posed Project/Research
"maxRIS" or "minRIS"

$$\boldsymbol{y} = \underbrace{\begin{bmatrix} H_{los} \\ H_{out} \cdot Q_H \cdot H_{in} \end{bmatrix}}_{H_{RIS}} \cdot \boldsymbol{x} + \underbrace{\begin{bmatrix} \boldsymbol{n}_{los} \\ \boldsymbol{n}_{out} + Q_H \cdot \boldsymbol{n}_{in} \end{bmatrix}}_{\boldsymbol{n}_{RIS}}$$



- The RIS matrix Q_H satisfies $||Q_H||_F^2 \leq G_H$, the RIS gain – it may also satisfy
  - Q_H is unitary matrix (preserves energy)
  - Q_H is diagonal, and usually also unitary, to be phase/gain-only adjustment on each antenna port (in-to-out)
  - Q_H has individual elements restricted

- For a given $R_{\boldsymbol{xx}}$, maximize over Q_H

$$\mathcal{I}(\boldsymbol{y};\boldsymbol{x}) = \log_2 |R_{n,RIS} + H_{RIS} \cdot R_{\boldsymbol{xx}} \cdot H_{RIS}^*|$$

- For a given Q_H , maximize the same over $R_{\boldsymbol{xx}}$

$$R_{\boldsymbol{nn},RIS} = \begin{bmatrix} R_{\boldsymbol{nn}} & 0 \\ 0 & R_{\boldsymbol{nn},out} + Q_H \cdot R_{\boldsymbol{nn}in} \cdot Q_H^* \end{bmatrix}$$

- Iterate

STANFORD

End Lecture 18
&
EE 392 AA - 2023

THANK YOU !!