OFDMA Downlink Resource Allocation using Limited Cross-Layer Feedback

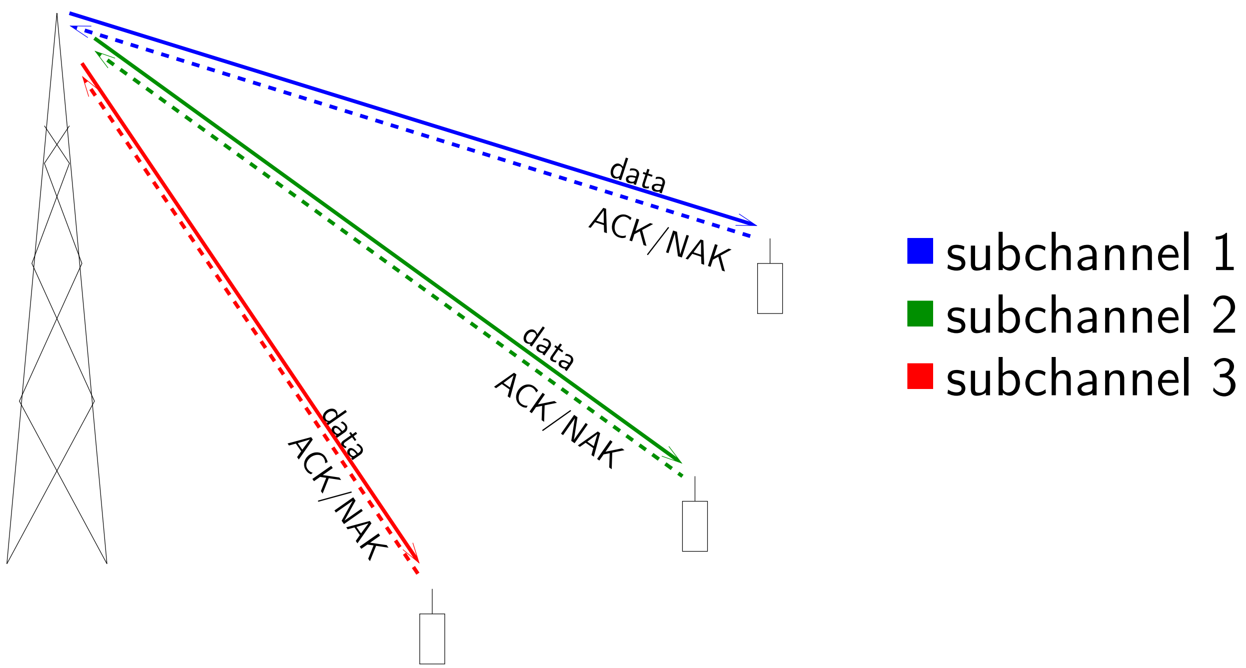
Prof. Phil Schniter

Joint work with
Mr. Rohit Aggarwal, Dr. Mohamad Assaad, Dr. C. Emre Koksal

September 20, 2010

Setup:

- Single-antenna downlink with K mobile users
- OFDMA with N subchannels
- Channels are Markov time-varying with impulse responses of length L
- ACK/NAK feedback on each transmitted data packet (i.e., ARQ)



The OFDMA Resource Allocation Problem:

- At each time t, we want to assign the "best" users (*multiuser diversity*) to their "best" subchannels (*frequency diversity*).
- We also want to optimize the powers and data-rates of assigned users.
- For this, we need channel state information (CSI) at the transmitter.
- Feedback of receiver CSI can be very costly! (Many users, subchannels.)
- Feedback of data-packet ACK/NAKs is free! (Provided by link layer.)

Is it possible to do near-optimal resource allocation using only ACK/NAK feedback from previously transmitted data packets?

Can we get **enough** *Tx-CSI from such limited cross-layer feedback?*

A Much Simpler Problem:

- Simplified scenario: $K = 1$ user, $N = 1$ subchannel, $L = 1$ channel tap.

  $\rightsquigarrow$ *Point-to-point communication over a time-varying channel.*

- The only parameter to optimize is the transmission rate r_t (bits/packet).

  $\rightsquigarrow$ *Rate adaptation via ARQ feedback.*

- Though it may sound simple, this is still an interesting problem!
  - Notice that r_t effects **short-term** throughput:

    r_t too high $\Rightarrow$ the packet will not be correctly decoded.

    r_t too low $\Rightarrow$ the channel will be underutilized.
  - But r_t also effects **long-term** throughput:
    Recalling that more accurate CSI leads to higher throughput. . .

    r_t very high $\Rightarrow$ likely NAK $\Rightarrow$ nothing learned about CSI.

    r_t very low $\Rightarrow$ likely ACK $\Rightarrow$ nothing learned about CSI.

    r_t such that $\Pr\{\text{ACK}\} = \frac{1}{2}$ is most informative about CSI.

Rate Adaptation based on ARQ Feedback:

Goal: Maximize total *expected* goodput over T packets:

$$[r_1^*,\ldots,r_T^*] = \underset{[r_1,\ldots,r_T]}{\operatorname{argmax}} \mathrm{E}\left\{\sum_{t=1}^{T} G(\gamma_t,r_t)\right\}, \quad G(\gamma,r) = \big(1-\epsilon(\gamma,r)\big)r.$$

- If our knowledge of the SNR γ_t was decoupled from the rates $\{r_\tau\}_{\tau\neq t}$, then it would be relatively easy to calculate the optimal $\{r_t^*\}_{t=1}^T$:

$$r_t^* = \underset{r_t}{\operatorname{argmax}} \mathrm{E}\{G(\gamma_t,r_t)\} = \underset{r_t}{\operatorname{argmax}} \int G(\gamma_t,r_t)p(\gamma_t)dt.$$

- But really, our knowledge of SNR γ_t will depend on the ARQ feedbacks received up to time t, which are strongly influenced by rates $\{r_\tau\}_{\tau=1}^{t-1}$.

  The current rate allocation affects not only immediate goodput, but also future SNR knowledge (and hence future goodput)!

  $\Rightarrow$ *A classical tradeoff between* **exploitation** *and* **exploration**.

Optimal Rate Adaptation:

Optimal rate adaptation, i.e.,

$$r^*_{t+1} = \underset{r_{t+1}}{\operatorname{argmax}} \; \mathrm{E}\left\{ G(r_{t+1}, \gamma_{t+1}) + \sum_{\tau=t+2}^{T} G(r^*_\tau, \gamma_\tau) \;\middle|\; \hat{\boldsymbol{\epsilon}}_t, \boldsymbol{r}_t \right\}$$

can be recognized as a *dynamic program*.

Denoting the optimal expected future-goodput by

$$\bar{G}^*_{t+1:T}(\hat{\boldsymbol{\epsilon}}_t, \boldsymbol{r}_t) \triangleq \mathrm{E}\left\{ \sum_{\tau=t+1}^{T} G(r^*_\tau, \gamma_\tau) \;\middle|\; \hat{\boldsymbol{\epsilon}}_t, \boldsymbol{r}_t \right\},$$

we can write the Bellman equation as

$$\bar{G}^*_{t+1:T}(\hat{\boldsymbol{\epsilon}}_t, \boldsymbol{r}_t) = \max_{r_{t+1}} \Big\{ \mathrm{E}\{G(r_{t+1}, \gamma_{t+1}) \mid \hat{\boldsymbol{\epsilon}}_t, \boldsymbol{r}_t\} + \mathrm{E}\{\bar{G}^*_{t+2:T}([\hat{\boldsymbol{\epsilon}}_t, \hat{\epsilon}_{t+1}], [\boldsymbol{r}_t, r_{t+1}]) \mid \hat{\boldsymbol{\epsilon}}_t, \boldsymbol{r}_t\} \Big\}$$

with the first expectation over γ_{t+1} and the second expectation over $\hat{\epsilon}_{t+1}$.

Optimal Rate Adaptation (cont.):

The solution to this dynamic program is a *partially observable Markov decision process* (POMDP), which

- is *intractable* when the SNRs $\{\gamma_t\}$ are *continuous* random variables,
- can be solved numerically when the SNRs $\{\gamma_t\}$ are *discrete*, but at *great cost*: both complexity and memory increase exponentially with the horizon length T and the number of states used for γ_t. ("PSPACE-complete")

$\Rightarrow$ *Not practical to implement!*

Greedy Adaptation:

- What if we maximize only the *short-term* reward?
- This corresponds to the *greedy* scheme

$$r_{t+1}^{\mathrm{g}} \triangleq \underset{r_{t+1}}{\operatorname{argmax}} \ \mathrm{E}\left\{G(r_{t+1}, \gamma_{t+1}) \,\middle|\, \hat{\boldsymbol{\epsilon}}_t, \boldsymbol{r}_t\right\},$$

which should be much easier to implement.

How good is this greedy scheme (w.r.t the optimal POMDP)?

The Causal Genie:

Consider the optimal rate under *non-degraded* causal error-rate feedback:

$$r_{t+1}^{\text{cg}} \triangleq \underset{r_{t+1}}{\operatorname{argmax}} \ \mathrm{E}\left\{ G(r_{t+1}, \gamma_{t+1}) + \sum_{\tau=t+2}^{T} G(r_\tau^{\text{cg}}, \gamma_\tau) \;\middle|\; \boldsymbol{\epsilon}_t, \boldsymbol{r}_t \right\}.$$

For a given rate r_t, the error ϵ_t uniquely determines the SNR γ_t, so that

$$r_{t+1}^{\text{cg}} = \underset{r_{t+1}}{\operatorname{argmax}} \ \mathrm{E}\left\{ G(r_{t+1}, \gamma_{t+1}) + \sum_{\tau=t+2}^{T} G(r_\tau^{\text{cg}}, \gamma_\tau) \;\middle|\; \boldsymbol{\gamma}_t \right\}.$$

Since future causal-genie rates $\{r_\tau^{\text{cg}}\}_{\tau \geq t+2}$ will be chosen based on perfect SNR knowledge, they will not depend on r_{t+1}. Thus,

$$r_{t+1}^{\text{cg}} = \underset{r_{t+1}}{\operatorname{argmax}} \ \mathrm{E}\left\{ G(r_{t+1}, \gamma_{t+1}) \;\middle|\; \boldsymbol{\gamma}_t \right\}.$$

Optimal adaptation under non-degraded error-rate feedback is greedy!

The Causal Genie is an Upper Bound on the POMDP:

Since $\mathrm{E}\{G(r_t^*, \gamma_t) \mid \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d}\}$

$$\leqslant \max_{r_t \in \mathcal{R}} \mathrm{E}\{G(r_t, \gamma_t) \mid \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d}\}$$

...since r_t^* is not necessarily short-term optimal

$$= \max_{r_t \in \mathcal{R}} \mathrm{E}\Big\{ \mathrm{E}\{G(r_t, \gamma_t) \mid \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d}, \boldsymbol{\epsilon}_{t-d}\} \Bigm| \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d}\Big\}$$

$$\leqslant \mathrm{E}\Big\{ \max_{r_t \in \mathcal{R}} \mathrm{E}\{G(r_t, \gamma_t) \mid \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d}, \boldsymbol{\epsilon}_{t-d}\} \Bigm| \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d}\Big\}$$

...since $\max_{r_t} \mathrm{E}\{f(r_t)\} \leqslant \mathrm{E}\{\max_{r_t} f(r_t)\}$ for any $f(\cdot)$

$$= \mathrm{E}\Big\{ \max_{r_t \in \mathcal{R}} \mathrm{E}\{G(r_t, \gamma_t) \mid \boldsymbol{\gamma}_{t-d}\} \Bigm| \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d}\Big\}$$

$$= \mathrm{E}\{G(r_t^{\mathsf{cg}}, \gamma_t) \mid \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d}\},$$

summing and averaging both sides gives

$$\mathrm{E}\left\{ \sum_{t=1}^{T} G(r_t^*, \gamma_t) \right\} \leqslant \mathrm{E}\left\{ \sum_{t=1}^{T} G(r_t^{\mathsf{cg}}, \gamma_t) \right\}.$$

Recap:

So far we've shown that, in the point-to-point problem,

1. Optimal rate adaptation is very difficult to implement/analyze.
2. There exists a suboptimal greedy scheme which is easy to implement.
3. The causal-genie scheme (i.e., the optimal scheme based on non-degraded feedback) is easy to implement, since it's greedy.
4. The causal genie upper bounds the performance of the optimal scheme.

So, if we can show that the *greedy scheme* is close to the *causal genie*, then greedy must be close to optimal.

Numerical Examples – Setup:

- Modulation: Uncoded square-QAM, $p = 100$, packet error rate

$$\epsilon(r_t, \gamma_t) \;=\; 1 - \left(1 - 2\left(1 - \frac{1}{\sqrt{2^{r_t/p}}}\right) Q\left(\sqrt{\frac{3\gamma_t}{2^{r_t/p} - 1}}\right)\right)^{2p}.$$

- ARQ feedback: $\hat{\epsilon}_t \in \{0, 1\}$

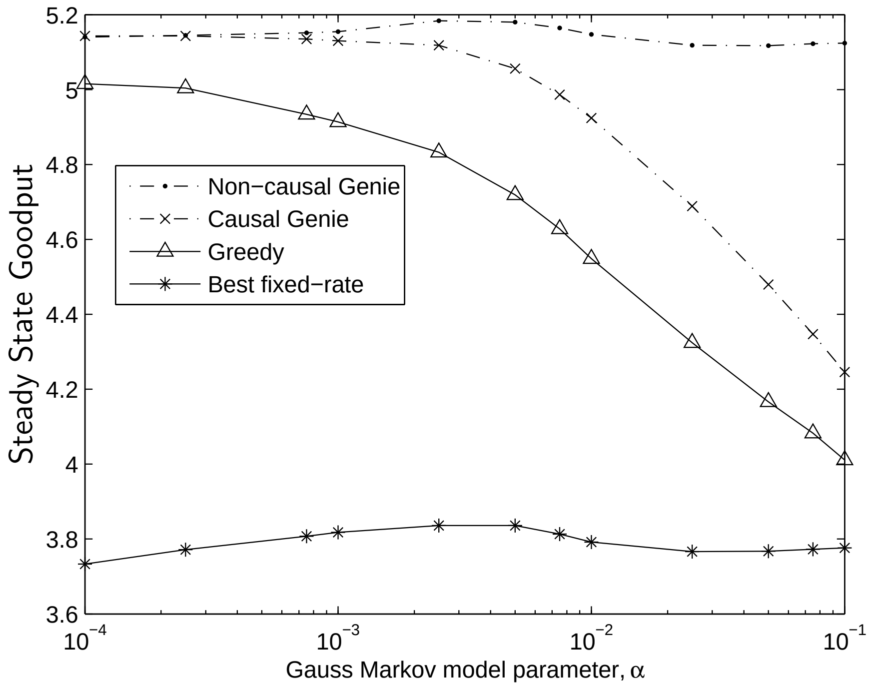
$$\Pr(\hat{\epsilon}_t = f \mid \epsilon_t) \;=\; \begin{cases} \epsilon_t & \text{for } f = 0 \quad (\text{“NAK”}) \\ 1 - \epsilon_t & \text{for } f = 1 \quad (\text{“ACK”}) \end{cases}$$

- SNR variation:
  - Rayleigh-fading channel gain (via Gauss-Markov process):

$$g_{t+1} \;=\; (1 - \alpha) g_t + \alpha w_t, \quad \{w_t\} \sim \text{i.i.d } \mathcal{CN}(0, 1)$$

  - SNR: $\gamma_t = C|g_t|^2$.
  - Parameters (α, C) control the mean and coherence time of γ_t. For a given α, we assume C is chosen so that $\mathrm{E}\{\gamma_t\} = 25$ dB.

Steady-state goodput versus fading-rate parameter α:



Greedy rate adaptation extracts most of the goodput gain available from the use of causal ARQ feedback!

$$\text{Best fixed-rate}: r_{t+1}^{\mathsf{fr}} = \operatorname{argmax}_{r_{t+1}} \int G(r_{t+1}, \gamma_{t+1}) p(\gamma_{t+1}) d\gamma_{t+1}$$

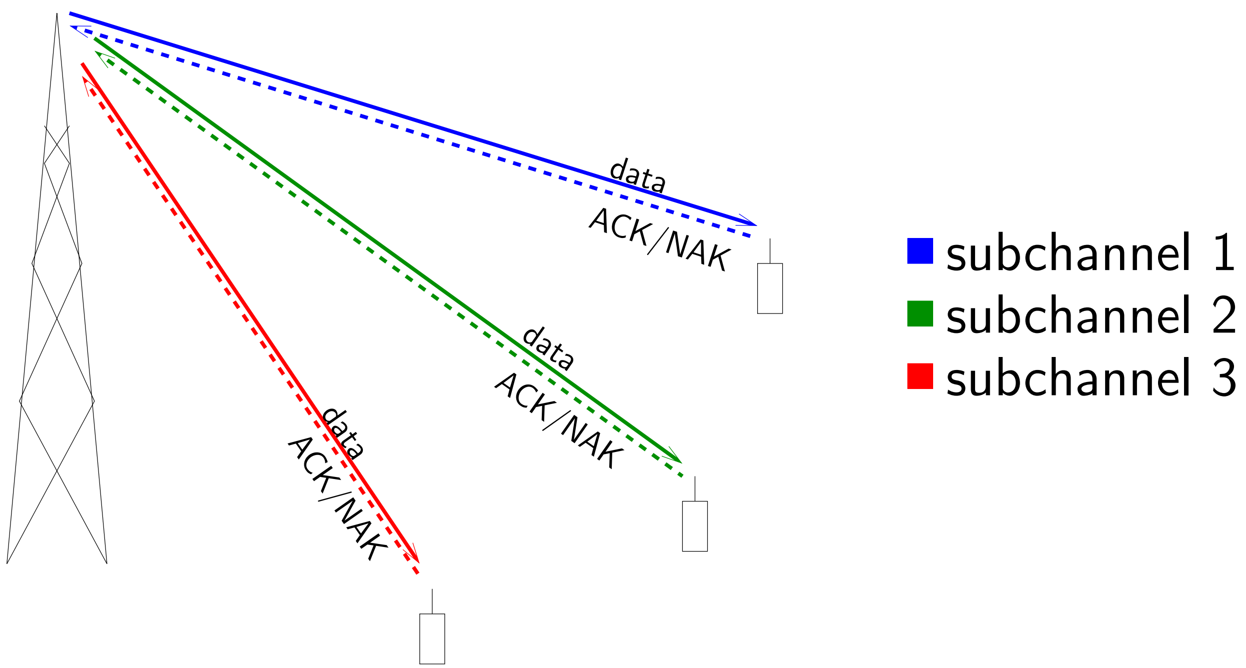
$$\text{Greedy}: r_{t+1}^{\mathsf{g}} = \operatorname{argmax}_{r_{t+1}} \int G(r_{t+1}, \gamma_{t+1}) p(\gamma_{t+1} | \hat{\boldsymbol{\epsilon}}_t, \boldsymbol{r}_t) d\gamma_{t+1}$$

$$\text{Causal Genie}: r_{t+1}^{\mathsf{cg}} = \operatorname{argmax}_{r_{t+1}} \int G(r_{t+1}, \gamma_{t+1}) p(\gamma_{t+1} | \underbrace{\boldsymbol{\epsilon}_t, \boldsymbol{r}_t}_{\boldsymbol{\gamma}_t}) d\gamma_{t+1}$$

$$\text{Non-causal Genie}: r_{t+1}^{\mathsf{ng}} = \operatorname{argmax}_{r_{t+1}} G(r_{t+1}, \gamma_{t+1})$$

Back to the OFDMA Problem:

- Single-antenna downlink with K mobile users
- OFDMA with N subchannels
- Channels are Markov time-varying with impulse responses of length L
- ACK/NAK feedback on each transmitted data packet (i.e., ARQ)



Formal Objective:

At each time t and subchannel n, choose each user k's next...

- rate $r_{n,k,t+1} \in \mathcal{R}$, for some M-ary rate alphabet $\mathcal{R}$,
- power $p_{n,k,t+1} \geq 0$,

based on ARQ feedbacks $\boldsymbol{F}_{1:t}$, to maximize the total future utility

$$\bar{G}_{t+1:T} = \sum_{\tau=t+1}^{T} \sum_{k=1}^{K} \mathrm{E}\left\{ \sum_{n=1}^{N} U\Big(\underbrace{\big(1 - \epsilon(r_{n,k,\tau}, \gamma_{n,k,\tau}, p_{n,k,\tau})\big) r_{n,k,\tau}}_{\text{goodput from } k \text{ on } n \text{ at } \tau} \Big) \Big| \boldsymbol{F}_{1:t} \right\}$$

subject to the power constraint $\sum_{n,k} p_{n,k,\tau} \leq P_{\text{con}}, \quad \forall \tau$,

and subject to ≤ 1 (user,rate) per (time,subchannel).

Here, $\epsilon(r, \gamma, p)$ is packet error rate, $U(\cdot)$ is a concave utility function, and $\boldsymbol{F}_{1:t} \in \{0, 1, \emptyset\}^{tNK}$ collects all past ARQ feedbacks.

Greedy Resource Allocation:

Using the indicator $I_{n,m,k,t} \in \{0,1\}$ to denote time-t assignment of subchannel n to user k at MCS index m, the time-t greedy allocation is

$$\max_{\substack{I_{n,k,m,t+1}\in\{0,1\} \\ p_{n,k,m,t+1}\geq 0}} \sum_k \mathrm{E}\left\{ \sum_{n,m} U\Big(I_{n,k,m,t+1}(1-a_m e^{-b_m p_{n,k,m,t+1}\gamma_{n,k,t+1}})r_m\Big) \Big| \boldsymbol{F}_{1:t}\right\}$$

$$\text{subject to } \sum_{n,k,m} I_{n,k,m,t}\, p_{n,k,m,t} \leq P_{\text{con}}, \quad \forall t,$$

$$\text{and } \sum_{k,m} I_{n,k,m,t} \leq 1, \quad \forall n, \ \forall t,$$

where

- $\gamma_{n,k,t}$ is SNR of user k on subchannel n at time t,
- (a_m, b_m, r_m) determines data rate and error rate for MCS index m,
- $\boldsymbol{F}_{1:t}$ collects all ARQ feedbacks collected from times 1 to t.

A mixed-integer programming problem!

Greedy Resource Allocation:

Exact solution:

- Brute Force: Compute water-filling solution for every hypothesized (subcarrier,user,rate) combination. Costs $\mathcal{O}\big((KM)^N\big)$.

Proposed approximate solution:

- Relax problem by allowing each (subcarrier,time) slot to be shared among multiple (user,rate) pairs.
  - Can reformulate as a convex optimization problem.
  - Can solve dual problem using an iterative algorithm with exponential convergence rate at $\mathcal{O}(NKM)$ per iteration.
- Byproducts of above algorithm yield a near-exact solution to the original mixed-integer problem.

R. Aggarwal, M. Assaad, C.E. Koksal, P. Schniter, "Optimal Joint Scheduling and Resource Allocation in OFDMA Downlink Systems with Imperfect Channel-State Information," *manuscript in preparation*.

Approximate Greedy Algorithm — Sketch:

Say that we relax the binary indicators to $\tilde{I}_{n,m,k,t} \in [0,1]$.
Then the KKT conditions become (suppressing the time-t notation):

$$\forall n,k,m, \quad \mu = a_m b_m r_m \,\mathrm{E}\{\gamma_{n,k} e^{-b_m p_{n,k,m}\gamma_{n,k}} \mid \boldsymbol{F}\} \qquad (1)$$

$$\forall n,k,m, \quad \lambda_n = r_m \,\mathrm{E}\{1 - a_m e^{-b_m p_{n,k,m}\gamma_{n,k}} \mid \boldsymbol{F}\} - \mu p_{n,k,m} \qquad (2)$$

where $\{\lambda_n\}_{n=1}^N$ and μ are Lagrange multipliers. A practical alg is then:

1. Initialize search region $[\underline{\mu}, \overline{\mu}]$ with midpoint $\mu = (\underline{\mu} + \overline{\mu})/2$.
2. For each subchannel n,
   - For each $(k,m)\ldots$
     - calculate $p_{n,k,m}$ by solving (1) and forcing a non-negative result.
     - plug $p_{n,k,m}$ into (2) and calculate the corresponding $\lambda_n(k,m)$.
   - Find $(k^*, m^*) = \arg\max_{(k,m)} \lambda_n(k,m)$.
   - Set $I_{n,k^*,m^*} = 1$ and $I_{n,k,m}|_{(k,m)\neq(k^*,m^*)} = 0$.
3. If $\sum_n p_{n,k^*,m^*} > P_{\textsf{con}}$, set $[\underline{\mu}, \overline{\mu}] \leftarrow [\mu, \overline{\mu}]$, else set $[\underline{\mu}, \overline{\mu}] \leftarrow [\underline{\mu}, \mu]$. Goto 2.

Approximate Greedy Algorithm — Example Performance:

N	K	M	greedy goodput (brute force)	approximation
1	3	9	5.9884	5.988
1	5	9	6.3501	6.3499
2	3	9	10.3251	10.3249
2	5	9	10.9778	10.9774
3	3	9	14.0573	14.0571
3	5	9	14.9653	14.9651

The practical approximation typically yields 99.99% of the goodput attained by the exact greedy scheme!

We have also derived performance bounds on our approximation.

Tracking the SNR distribution:

The greedy allocator tracks the SNR by updating the SNR distributions

$$\left\{p(\gamma_{n,k,t+1} \mid \boldsymbol{F}_{1:t}), \quad \forall \text{ users } k \text{ and subchannels } n\right\}$$

The SNR evolves as follows:

- Markov evolution of time-domain channel taps:

$$h_{l,k,t+1} \;=\; (1-\alpha)h_{l,k,t} + \alpha w_{l,k,t}, \quad w_{l,k,t} \sim \mathcal{CN}(0,1),$$

- subchannel gains as a function of time-domain channel taps:

$$H_{n,k,t} \;=\; \sum_{l=0}^{L-1} h_{l,k,t} e^{-j\frac{2\pi}{N}nk},$$

- subchannel SNRs as a function of subchannel gains:

$$\gamma_{n,k,t} \;=\; C|H_{n,k,t}|^2.$$

Tracking the SNR distribution (cont.):

SNR tracking can be done as follows:

$$p(\gamma_{n,k,t+1} \mid \boldsymbol{F}_{1:t}) = \int_{\boldsymbol{h}_{k,t+1}} \underbrace{p(\gamma_{n,k,t+1} \mid \boldsymbol{h}_{k,t+1})}_{\text{(approx of) Dirac delta}} p(\boldsymbol{h}_{k,t+1} \mid \boldsymbol{F}_{1:t}) \quad (3)$$

$$p(\boldsymbol{h}_{k,t+1} \mid \boldsymbol{F}_{1:t}) = \int_{\boldsymbol{h}_{k,t}} \underbrace{p(\boldsymbol{h}_{k,t+1} \mid \boldsymbol{h}_{k,t})}_{\text{Markov prediction}} p(\boldsymbol{h}_{k,t} \mid \boldsymbol{F}_{1:t}) \quad (4)$$

$$p(\boldsymbol{h}_{k,t} \mid \boldsymbol{F}_{1:t}) = \frac{p(\boldsymbol{f}_{k,t} \mid \boldsymbol{h}_{k,t}) p(\boldsymbol{h}_{k,t} \mid \boldsymbol{F}_{1:t-1})}{\int_{\boldsymbol{h}'_{k,t}} p(\boldsymbol{f}_{k,t} \mid \boldsymbol{h}'_{k,t}) p(\boldsymbol{h}'_{k,t} \mid \boldsymbol{F}_{1:t-1})} \quad \text{(Bayes rule)} \quad (5)$$

$$p(\boldsymbol{f}_{k,t} \mid \boldsymbol{h}_{k,t}) = \prod_{n=1}^{N} p(f_{n,k,t} \mid \gamma_{n,k,t}(\boldsymbol{h}_{k,t})) \quad (6)$$

$$p(f_{n,k,t} = f \mid \gamma_{n,k,t}) = \begin{cases} \sum_m I_{n,k,m,t} a_m e^{-b_m p_{n,k,m,t} \gamma_{n,k,t}} & f = 0 \\ \sum_m I_{n,k,m,t} (1 - a_m e^{-b_m p_{n,k,m,t} \gamma_{n,k,t}}) & f = 1 \\ 1 - \sum_m I_{n,k,m,t} & f = \emptyset \end{cases} \quad (7)$$

Tracking the SNR distribution (cont.):

Thus, for each user k,

1. measure feedbacks $\boldsymbol{f}_{k,t}$ across all subchannels,
2. compute $p(\boldsymbol{f}_{n,k} \mid \gamma_{n,k,t}(\boldsymbol{h}_{k,t}))$ on $\boldsymbol{h}$-lattice using error-rate rules (6)-(7),
3. compute $p(\boldsymbol{h}_{k,t} \mid \boldsymbol{F}_{1:t})$ on $\boldsymbol{h}$-lattice by updating previous posterior via (5),
4. compute $p(\boldsymbol{h}_{k,t+1} \mid \boldsymbol{F}_{1:t})$ on $\boldsymbol{h}$-lattice via Markov-prediction step (4),
5. compute $p(\boldsymbol{\gamma}_{k,t+1} \mid \boldsymbol{F}_{1:t})$ on $\boldsymbol{\gamma}$-lattice via $\boldsymbol{h}$-to-γ conversion step (3).

This costs $\mathcal{O}\big(KNQ_h^L + KLQ_h^{L+1} + KNQ_\gamma Q_h^L\big)$, where

$Q_h =$ number of grid points used per dimension of $\boldsymbol{h}$-lattice,

$Q_\gamma =$ number of grid points used per dimension of $\boldsymbol{\gamma}$-lattice.

Numerical Experiments:

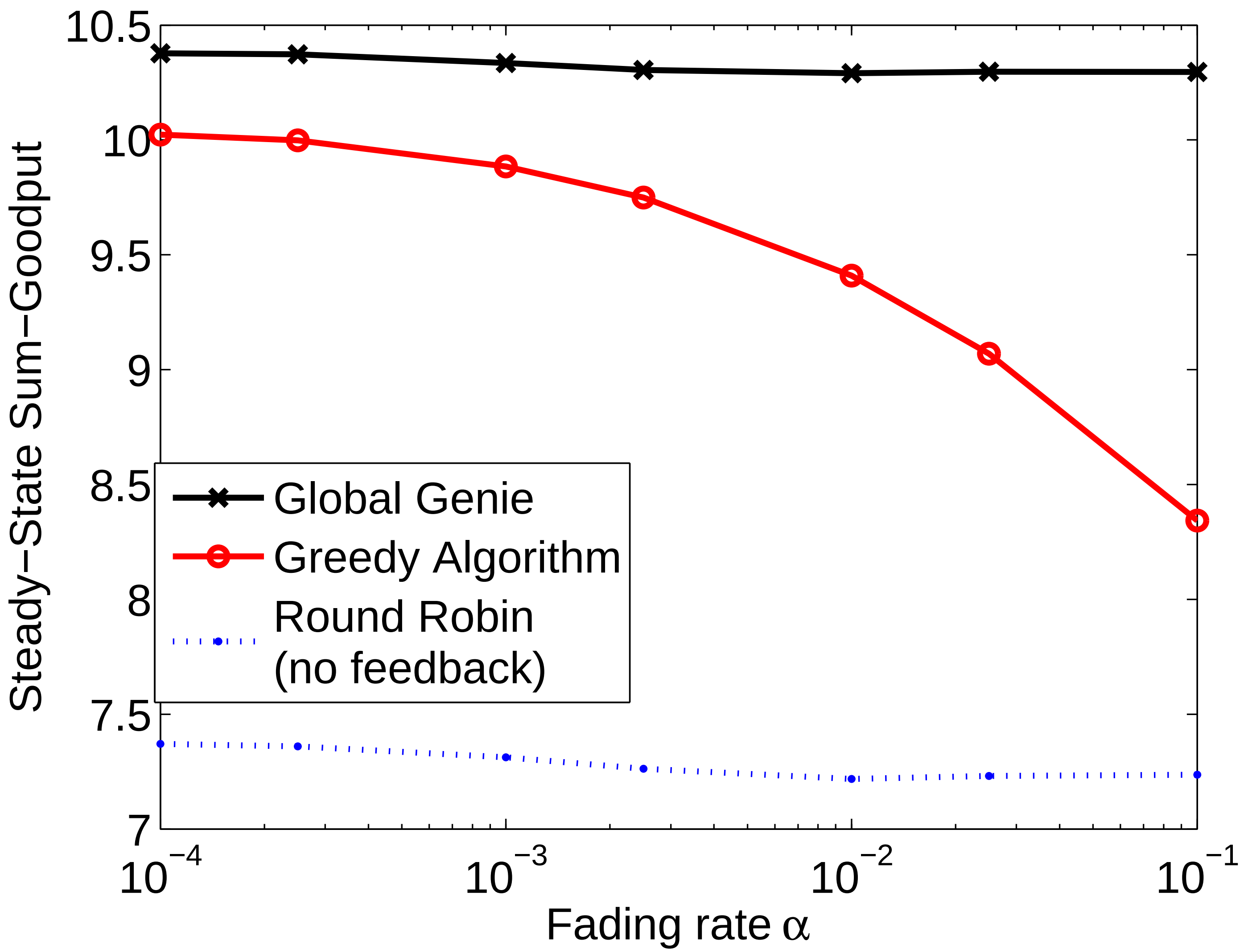
Setup:

$K = 2$	users
$N = 2$	subchannels
$L = 2$	time-domain channel taps
$\mathrm{E}\{\gamma_{n,k,t}\} = 25\text{dB}$	mean subchannel SNR
$\rho = 0.33$	subchannel correlation

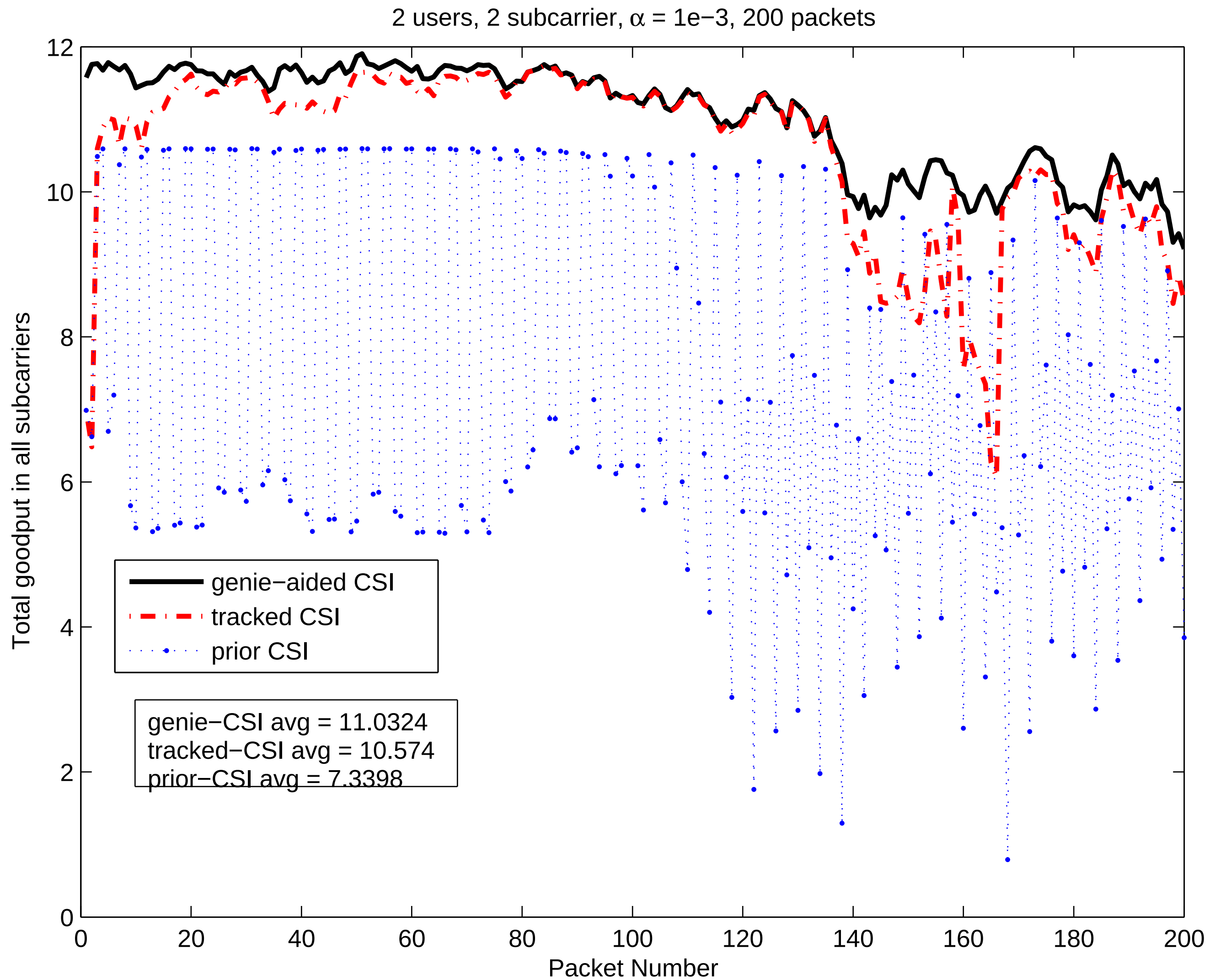
Plots show

- goodput versus fading rate α
- goodput versus time t
- power/rate/user versus time t

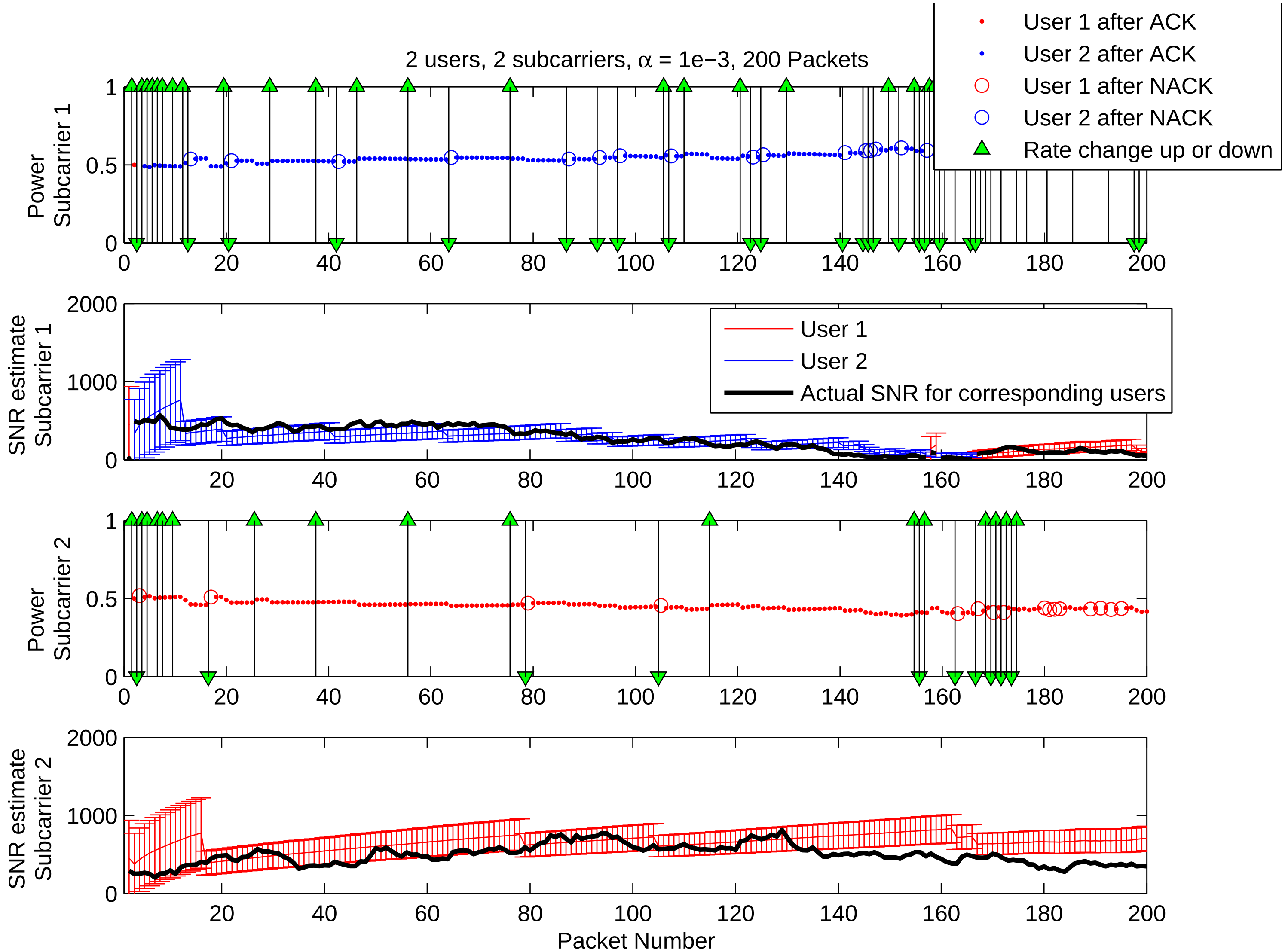
Steady-state goodput versus α:



Goodput for $\alpha = 0.001$:



Allocations for $\alpha = 0.001$:



Summary:

- Goal: From only ARQ feedback, optimize OFDMA users, powers, and rates to maximize finite-horizon expected goodput under an instantaneous total-power constraint.
- The optimal resource allocator is a POMDP, which is computationally impractical.
- We settle for greedy resource allocation, found to be near-optimal for practical fading rates.
- Greedy allocation is a mixed-integer programming problem, but we can solve it almost exactly with $\mathcal{O}(NKM)$ complexity.
- To maintain CSI, we track the SNR *distribution* (conditioned on past ACK/NAK feedback) of each user at each subcarrier.
- Preliminary experiments for 2 users and 2 subchannels indicates that our practical algorithm performs relatively close to a genie-aided upper bound on the optimal POMDP.

Thanks for listening!