Cross-Layer Rate Adaptation for Time-Varying Channels

Phil Schniter

July 2008

(Joint work with Dr. C. Emre Koksal and Mr. Rohit Aggarwal)

Basic Principles:

Communication in Gaussian noise:

$$\boxed{\epsilon \sim e^{-\gamma/2^r}} \text{ for } \begin{cases} \epsilon \ : \ \text{error rate} \\ \gamma \ : \ \text{SNR} \\ r \ : \ \text{transmission rate [bits/channel-use]} \end{cases}$$

Typical goals:

1. Low error rate ϵ,
2. High transmission rate r.

(We assume no control over SNR γ.)

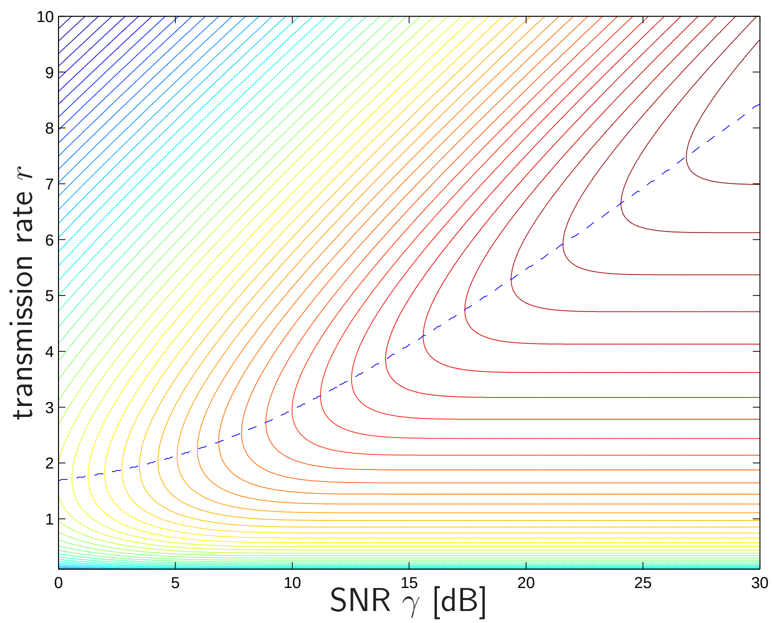
But these goals are in conflict! For example, we can decrease ϵ a little by decreasing r a little. . . Should we do it?

Goodput:

$\boxed{G = (1-\epsilon)r}$: the rate of *successful* communication.

- Useful because it unifies the conflicting goals of low error rate and high transmission rate.
- Plugging in the error-rate relationship $\epsilon \sim e^{-\gamma/2^r}$, we find

$$G \sim (1 - e^{-\gamma/2^r})r.$$

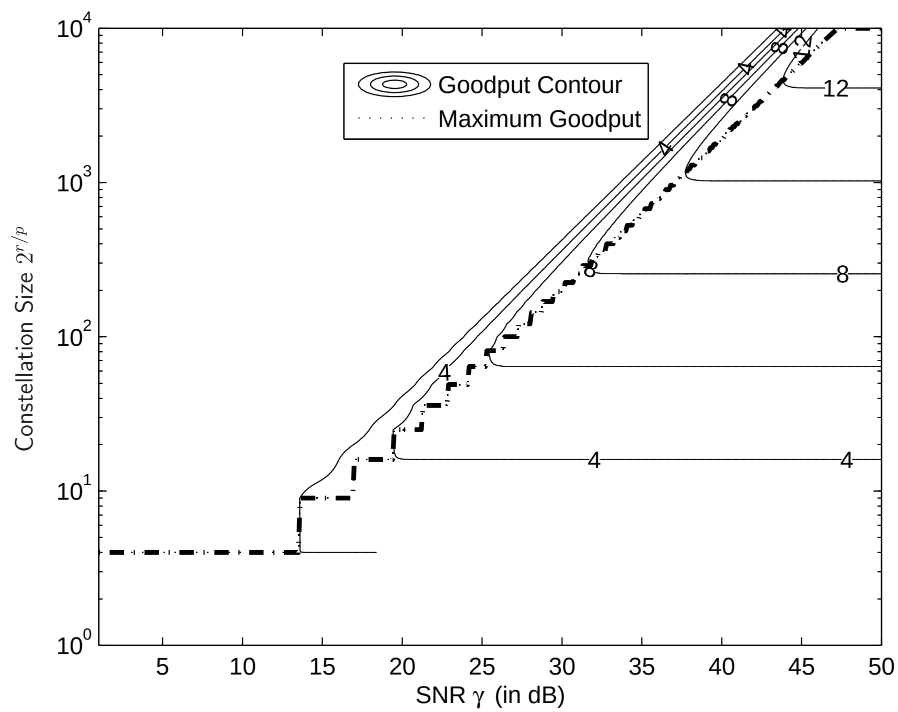


Notice that there is an optimal transmission rate r for every SNR γ.

Goodput (cont.):

For a more concrete example, consider the goodput per p-length packet of uncoded square-QAM symbols, where packet error rate equals

$$\epsilon(r,\gamma) \;=\; 1-\left(1-2\left(1-\frac{1}{\sqrt{2^{r/p}}}\right)Q\left(\sqrt{\frac{3\gamma}{2^{r/p}-1}}\right)\right)^{2p}:$$



Rate Adaptation:

- If we knew the SNR γ, we could easily select the goodput-maximizing transmission rate.
- But in practice the SNR varies as a result of
  - Multipath fading,
  - Interference.
- While tracking the SNR may be straightforward at the receiver, it is often non-trivial at the transmitter.

  For example, SNR estimation at the receiver followed by feedback to the transmitter consumes
  - Bandwidth (on both forward and backward channels),
  - Power (on both forward and backward channels),
  - Complexity.

A Cross-Layer Approach:

Automatic Repeat reQuest (ARQ):

- A link-layer technique used in many systems.
- For each packet, the receiver sends an ACK if it was received correctly or a NAK otherwise.
- Used to ensure strict end-to-end error rates (e.g., 10^{-10}) while keeping the physical layer practical (e.g., $\epsilon \sim 10^{-3}$).

The Main Idea:

- Use link-layer ARQ feedback for physical-layer rate adaptation.

$\rightsquigarrow$ *"Cross-layer rate adaptation"*

Problem Formulation:

Naive Goal: Choose rates to maximize total goodput over T packets:

$$[r_1^*, \ldots, r_T^*] = \underset{[r_1,\ldots,r_T]\in\mathcal{R}^T}{\arg\max} \sum_{t=1}^{T} G(\gamma_t, r_t).$$

where $\mathcal{R}$ is the set of allowed rates.

- If $\{\gamma_t\}_{t=1}^T$ was known perfectly, calculating $\{r_t^*\}_{t=1}^T$ would be easy.
- But, of course, SNR γ_t is not perfectly known!

Problem Formulation (cont.):

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

$$[r_1^*, \ldots, r_T^*] = \underset{[r_1,\ldots,r_T]\in\mathcal{R}^T}{\arg\max} \; \mathrm{E}\left\{ \sum_{t=1}^{T} G(\gamma_t, r_t) \right\}.$$

- If the statistics of γ_t were known and decoupled from the rates $\{r_k\}_{k\neq t}$, then it would still be relatively easy to calculate $\{r_t^*\}_{t=1}^{T}$:

$$r_t^* = \underset{r_t\in\mathcal{R}}{\arg\max}\, \mathrm{E}\{G(\gamma_t, r_t)\} = \underset{r_t\in\mathcal{R}}{\arg\max} \int G(\gamma_t, r_t) p(\gamma_t) dt.$$

- But our knowledge of γ_t will depend on the previously received ACK/NAKs, which depend on the previously chosen rates!

  The current rate choice affects not only the immediate goodput but also our future SNR knowledge (and hence future rate selection)!

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

Intuition:

- If rates are chosen purely based on "exploitation" (i.e., to maximize short-term goodput), it is not clear whether the feedback will be adequately informative about changes in SNR.
- Pilot-based transmission schemes could be interpreted as dedicating a certain percentage of packets to "exploration." But is the implied sacrifice in "exploitation" worthwhile?

What does the optimal rate-selecting scheme do?

Rate Adaptation via Degraded Error-Rate Feedback:

We have now seen that causal rate adaptation...

- via degraded SNR feedback is straightforward:

$$r_t^* = \arg\max_{r_t \in \mathcal{R}} \mathrm{E}\{G(\gamma_t, r_t) \mid \hat{\boldsymbol{\gamma}}_{t-d}\}$$

- via degraded error-rate feedback is more interesting:

$$r_t^* = \arg\max_{r_t \in \mathcal{R}} \mathrm{E}\left\{G(r_t, \gamma_t) + \sum_{k=t+1}^{T} G(r_k^*, \gamma_k) \,\middle|\, \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d}\right\}.$$

Notation: d : feedback delay ($d \geq 1$),

$\hat{\boldsymbol{\gamma}}_{t-d} = [\hat{\gamma}_1, \ldots, \hat{\gamma}_{t-d}]$: previous SNR estimates,

$\hat{\boldsymbol{\epsilon}}_{t-d} = [\hat{\epsilon}_1, \ldots, \hat{\epsilon}_{t-d}]$: previous error-rate estimates

$\boldsymbol{r}_{t-d} = [r_1, \ldots, r_{t-d}]$: previous transmission rates.

Notice: We now consider *general* forms of degraded error-rate feedback $\hat{\boldsymbol{\epsilon}}_{t-d}$.

The Optimal Rate Adaptation:

Optimal rate selection, i.e.,

$$r_t^* \;=\; \arg\max_{r_t \in \mathcal{R}} \; \mathrm{E}\left\{ G(r_t, \gamma_t) + \sum_{k=t+1}^{T} G(r_k^*, \gamma_k) \;\middle|\; \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d} \right\}$$

can be recognized as a *dynamic program*. Denoting the optimal expected goodput for current and future packets by

$$G_t^*(\hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d}) \;:=\; \mathrm{E}\left\{ \sum_{k=t}^{T} G(r_k^*, \gamma_k) \;\middle|\; \hat{\boldsymbol{\epsilon}}_{t-d}, \boldsymbol{r}_{t-d} \right\},$$

we can write the Bellman equation as

$$\begin{aligned} G_t^*(\hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}) \;=\; \max_{r_t \in \mathcal{R}} \Big\{ & \mathrm{E}\{G(r_t, \gamma_t) \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}\} \\ & + \mathrm{E}\{G_{t+1}^*([\hat{\boldsymbol{\epsilon}}_{t-1}, \hat{\epsilon}_t], [\boldsymbol{r}_{t-1}, r_t]) \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}\} \Big\} \end{aligned}$$

for the case $d = 1$. (For $d > 1$, the expression is more complicated.)

The Optimal Rate Adaptation (cont.):

The solution to this dynamic program is a *partially observable Markov decision process* (POMDP), which requires complexity and memory that increase exponentially in the horizon length T.

$\Rightarrow$ *Too difficult to implement!*

Greedy Rate Assignment:

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

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

which is much easier to implement.

So, how bad is this greedy scheme (relative to the optimal)?

Upper Bounding the Optimal Scheme:

- For the optimal POMDP-based scheme, implementation and (exact) analysis are both very difficult.
- But can we *upper bound* the performance of the optimal scheme?

If we can find an upper bound, and if the greedy scheme is found to perform close to this upper bound, then the greedy scheme must perform close to optimal.

The Causal Genie:

Consider optimal rate selection under *non-degraded* error-rate feedback:

$$r_t^{\text{cg}} := \underset{r_t \in \mathcal{R}}{\arg\max} \ \mathrm{E}\left\{ G(r_t, \gamma_t) + \sum_{k=t+1}^{T} G(r_k^{\text{cg}}, \gamma_k) \;\middle|\; \boldsymbol{\epsilon}_{t-d}, \boldsymbol{r}_{t-d} \right\}.$$

Because $\boldsymbol{\gamma}_{t-d}$ can be uniquely determined from $(\boldsymbol{\epsilon}_{t-d}, \boldsymbol{r}_{t-d})$, we have

$$r_t^{\text{cg}} := \underset{r_t \in \mathcal{R}}{\arg\max} \ \mathrm{E}\left\{ G(r_t, \gamma_t) + \sum_{k=t+1}^{T} G(r_k^{\text{cg}}, \gamma_k) \;\middle|\; \boldsymbol{\gamma}_{t-d} \right\}.$$

Since future causal-genie rates $\{r_k^{\text{cg}}\}_{k>t}$ will be chosen based on perfect knowledge of (d-delayed) SNRs, they will not depend on r_t. Thus,

$$r_t^{\text{cg}} := \underset{r_t \in \mathcal{R}}{\arg\max} \ \mathrm{E}\left\{ G(r_t, \gamma_t) \;\middle|\; \boldsymbol{\gamma}_{t-d} \right\}.$$

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

The Causal Genie is an Upper Bound:

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

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

What Now?:

So far we've shown that

1. Optimal rate adaptation is very difficult to implement/analyze.
2. There exists a suboptimal greedy scheme which may be relatively easy to implement.
3. If the feedback was non-degraded, the optimal scheme (i.e., the "causal genie") would itself be greedy.
4. The causal genie yields an upper bound on the optimal scheme under degraded feedback.

Thus, we'd like to know how the *greedy scheme* compares to the *best fixed-rate scheme* and to the *causal genie*.

But first: How do we actually implement the greedy scheme?

Implementing the Greedy Scheme:

(Let's assume delay $d = 1$ for simplicity.)

The greedy rate choice can be written as

$$\hat{r}_t = \arg\max_{r_t \in \mathcal{R}} \int G(r_t, \gamma_t)\, p(\gamma_t \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}) d\gamma_t$$

using the inferred SNR distribution

$$\begin{aligned} p(\gamma_t \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}) &= \int p(\gamma_t \mid \gamma_{t-1}, \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1})\; p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1})\; d\gamma_{t-1} \\ &= \int \underbrace{p(\gamma_t \mid \gamma_{t-1})}_{\text{SNR prediction}}\; \underbrace{p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1})}_{\text{estimation of delayed SNR}}\, d\gamma_{t-1}. \end{aligned}$$

For the last step, we assumed *Markov SNR variation*.

Next, we tackle estimation of the previous SNR. . .

Implementing the Greedy Scheme (cont.):

Estimation of the previous SNR:

$$
\begin{aligned}
& p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}) \\
&= p(\gamma_{t-1} \mid \hat{\epsilon}_{t-1}, \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-1}) \\
&= \frac{p(\hat{\epsilon}_{t-1} \mid \gamma_{t-1}, \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-1})\; p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-1})}{\int p(\hat{\epsilon}_{t-1} \mid \gamma'_{t-1}, \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-1})\; p(\gamma'_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-1})\; d\gamma'_{t-1}} \\
&= \frac{p(\hat{\epsilon}_{t-1} \mid \epsilon(r_{t-1}, \gamma_{t-1}), \hat{\boldsymbol{\epsilon}}_{t-2})\; p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-2})}{\int p(\hat{\epsilon}_{t-1} \mid \epsilon(r_{t-1}, \gamma'_{t-1}), \hat{\boldsymbol{\epsilon}}_{t-2})\; p(\gamma'_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-2})\; d\gamma'_{t-1}}.
\end{aligned}
$$

Assuming *memoryless error degradation*: $p(\hat{\epsilon}_t \mid \epsilon_t, \hat{\boldsymbol{\epsilon}}_{t-1}) = p(\hat{\epsilon}_t \mid \epsilon_t)$,

$$
\begin{aligned}
& p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}) \\
&= \frac{p(\hat{\epsilon}_{t-1} \mid \epsilon(r_{t-1}, \gamma_{t-1}))\; p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-2})}{\int p(\hat{\epsilon}_{t-1} \mid \epsilon(r_{t-1}, \gamma'_{t-1}))\; p(\gamma'_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-2})\; d\gamma'_{t-1}},
\end{aligned}
$$

where $p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-2})$ would have been previously calculated.

Summary of Greedy Implementation:

For $t = 1, \ldots, T$,

1. Measure $\hat{\epsilon}_{t-1}$ and compute the pdf (as a function of γ_{t-1})

$$p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}) \;=\; \frac{p(\hat{\epsilon}_{t-1} \mid \epsilon(r_{t-1}, \gamma_{t-1}))\; p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-2})}{\int p(\hat{\epsilon}_{t-1} \mid \epsilon(r_{t-1}, \gamma'_{t-1}))\; p(\gamma'_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-2}, \boldsymbol{r}_{t-2})\; d\gamma'_{t-1}}.$$

2. Compute the pdf

$$p(\gamma_t \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}) \;=\; \int p(\gamma_t \mid \gamma_{t-1})\; p(\gamma_{t-1} \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}) d\gamma_{t-1}.$$

3. Choose the greedy rate

$$\hat{r}_t \;=\; \arg\max_{r_t \in \mathcal{R}} \int G(r_t, \gamma_t)\, p(\gamma_t \mid \hat{\boldsymbol{\epsilon}}_{t-1}, \boldsymbol{r}_{t-1}) d\gamma_t.$$

end;

Note: When $t = 1$, set $p(\gamma_1 \mid \hat{\boldsymbol{\epsilon}}_0, \boldsymbol{r}_0) = p(\gamma_1)$ and skip steps 1 & 2.

Modifications for Block Update:

- So far, we have assumed that the transmission rate is adapted once per packet.
- To reduce complexity, we could instead adapt the rate once per block of n packets. For this, we...
  - Treat the SNR as if it were *fixed* across each block.
  - Treat the packet error rate as if it were *fixed* across each block.
- This block updating modification can be applied to both the greedy scheme and the causal genie scheme.
- We expect performance to decrease with block length n due to
  - The block-fading channel approximation.
  - The delay imposed by the block update.

Numerical Results:

Next, we present some numerical results that demonstrate (perhaps surprisingly) that, using packet-level ARQ, *greedy rate adaptation performs relatively close to the upper bound on the optimal rate allocation* (i.e., the causal genie).

In particular, we investigate steady-state goodput versus:

- feedback delay d,
- block size n,
- channel fading rate,
- average SNR $\mathrm{E}\{\gamma_t\}$,

and we also investigate

- packet drop rate due to a finite input buffer.

Numerical Results – Setup:

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

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

- Degraded error-rate feedback: one ACK/NAK per packet.
- Estimation of block error rate:
  - Note: # NAKs per n-block $\sim$ Binomial(n, ϵ_t).
  - Minimum-variance unbiased estimate of average error-rate over the $\lfloor \frac{t}{n} \rfloor$th block: $\hat{\epsilon}_t = \frac{\#\ \text{NAKs}}{n}$.
  - Degraded error-rate:

$$p(\hat{\epsilon}_t = \tfrac{k}{n} \mid \epsilon_t) \;=\; \begin{cases} \binom{n}{k} \epsilon_t^k (1 - \epsilon_t)^{n-k} & \text{for } k = 0, \ldots, n \\ 0 & \text{else.} \end{cases}$$

Numerical Results – Setup (cont.):

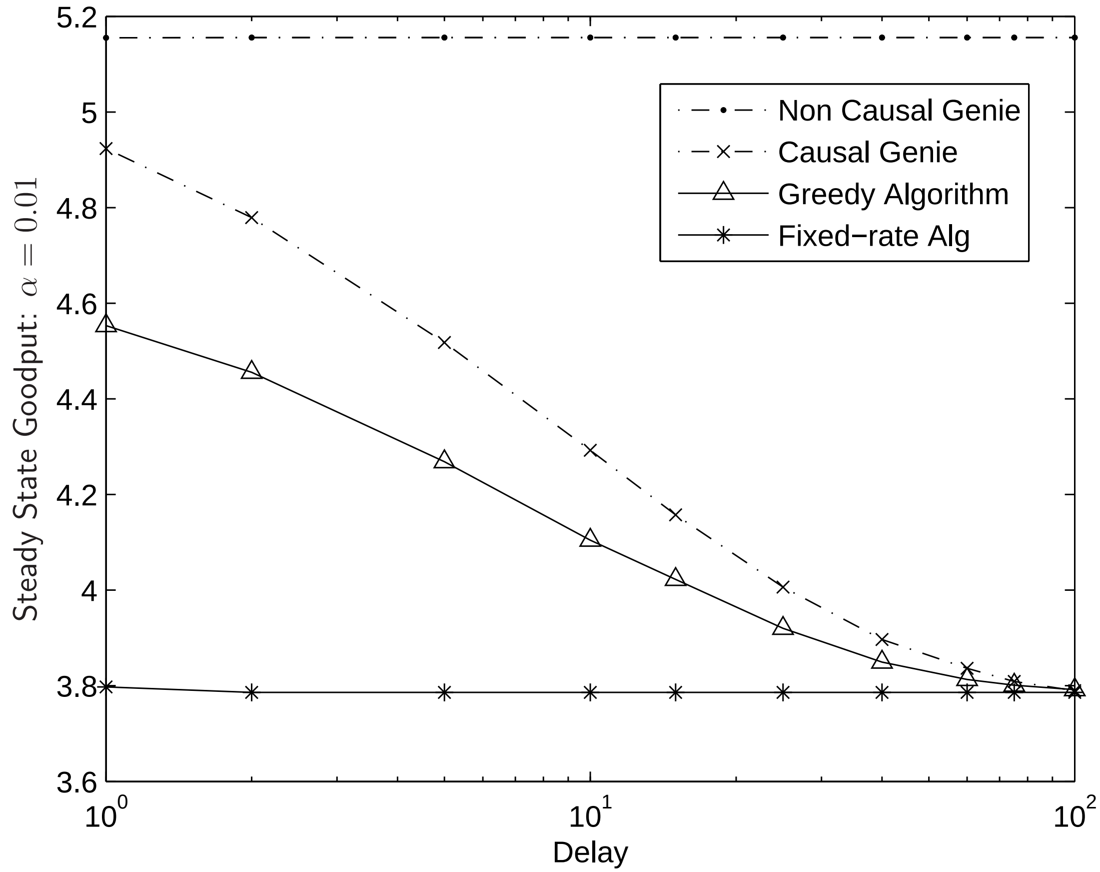
- SNR variation:
  - Rayleigh-fading channel gain via Gauss-Markov process:

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

  - SNR: $\gamma_t = K|g_t|^2.$
  - Parameters (α, K) control the SNR's mean and coherence time, where, nominally, we set $\mathrm{E}\{\gamma_t\} = 25$ dB and $\alpha = 0.01$.
- Schemes under comparison:

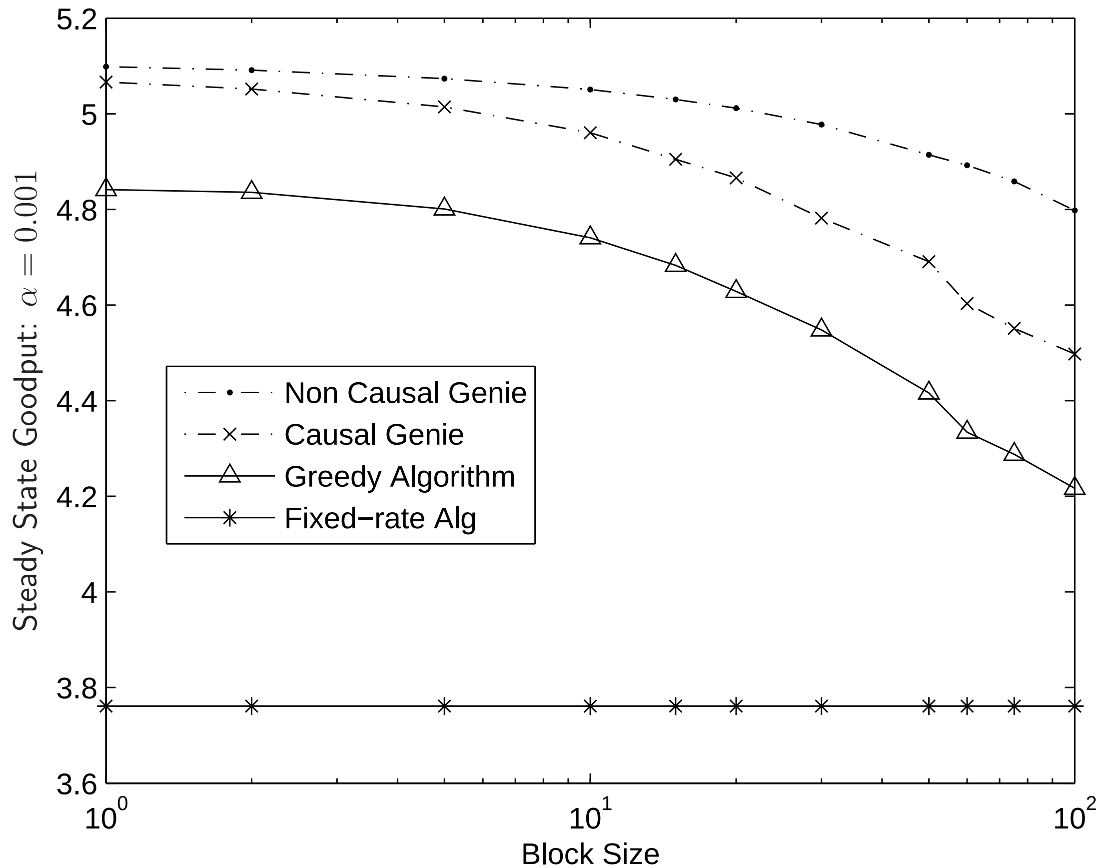
$$\begin{array}{rcl}
\text{Fixed-rate} & : & r_t^{\text{fr}} = \arg\max_{r_t} \int G(r_t,\gamma_t)p(\gamma_t)d\gamma_t \\
\text{Greedy} & : & \hat{r}_t = \arg\max_{r_t} \int G(r_t,\gamma_t)p(\gamma_t|\hat{\boldsymbol{\epsilon}}_{t-d},\boldsymbol{r}_{t-d})d\gamma_t \\
\text{Causal Genie} & : & r_t^{\text{cg}} = \arg\max_{r_t} \int G(r_t,\gamma_t)p(\gamma_t|\underbrace{\boldsymbol{\epsilon}_{t-d},\boldsymbol{r}_{t-d}}_{\boldsymbol{\gamma}_{t-d}})d\gamma_t \\
\text{Non-causal Genie} & : & r_t^{\text{ng}} = \arg\max_{r_t} G(r_t,\gamma_t)
\end{array}$$

Steady-state goodput versus feedback delay d:



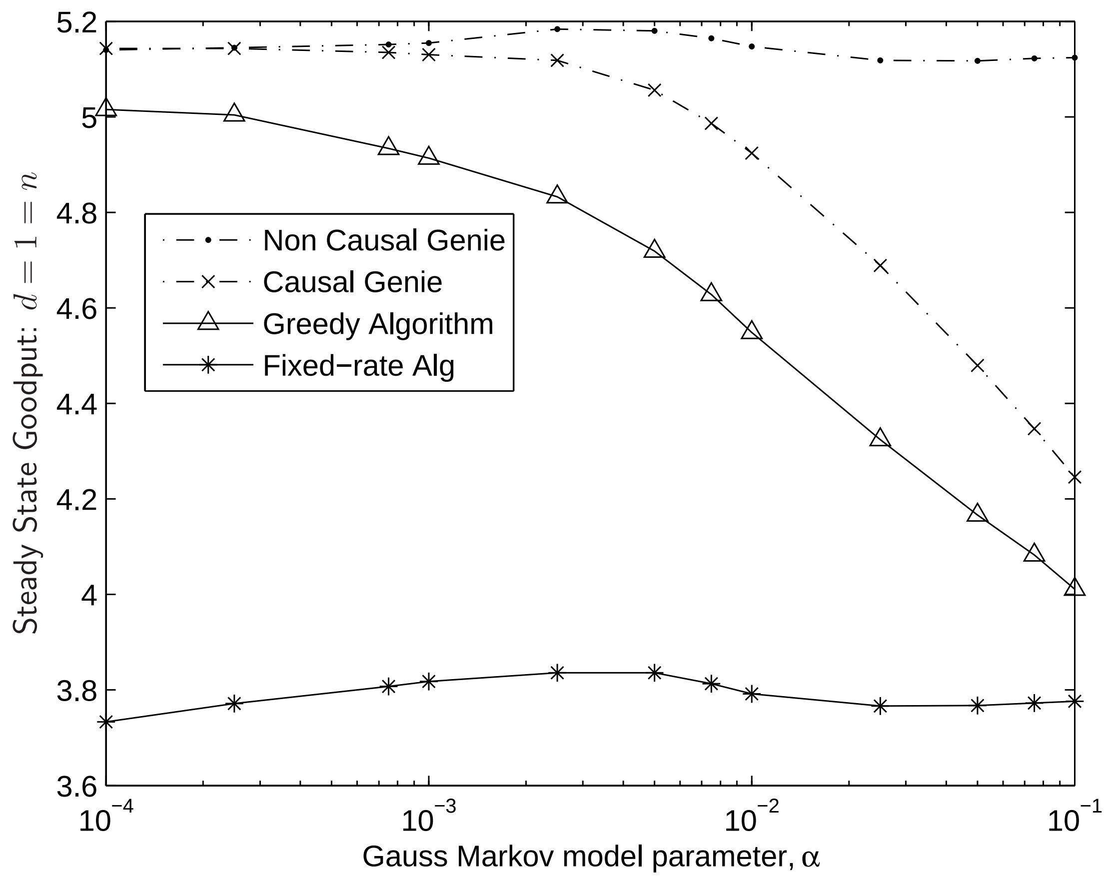
The greedy algorithm extracts most of the goodput gain achievable from the use of causal error-rate feedback, which diminishes with delay d.

Steady-state goodput versus block-size n:



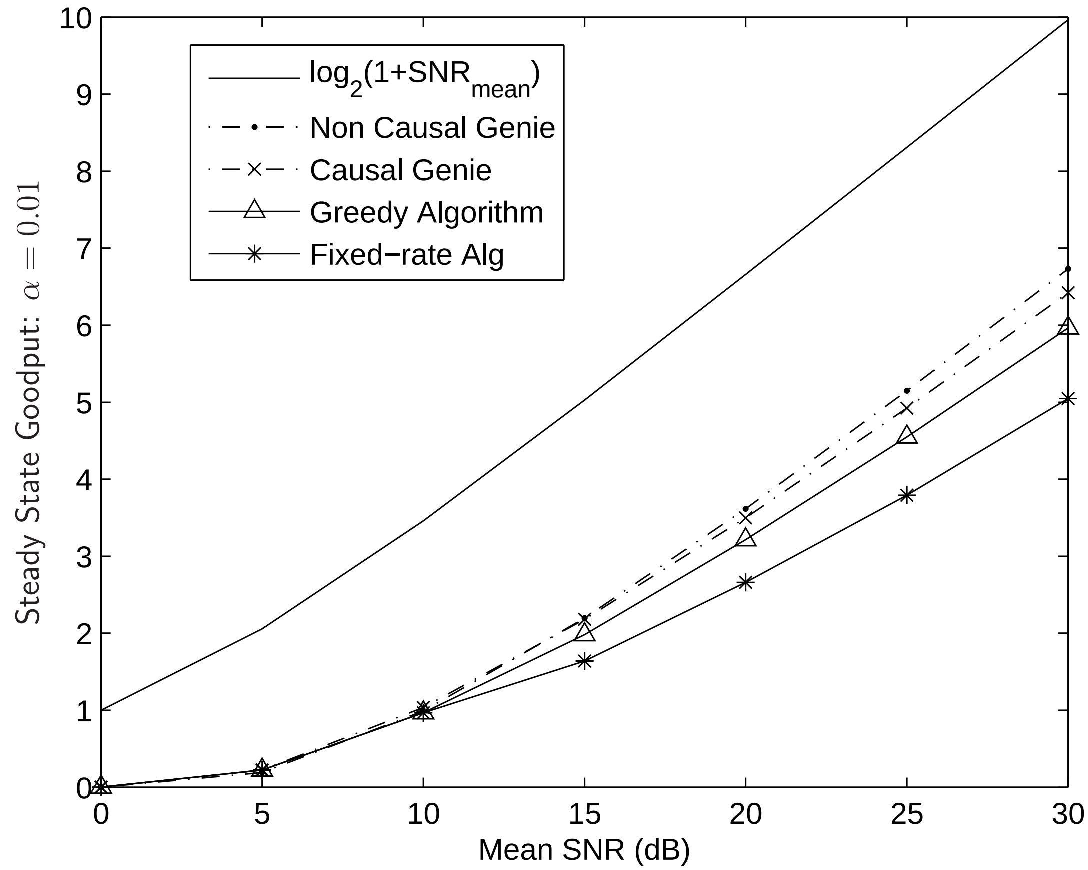
The greedy algorithm extracts most of the goodput gain achievable from the use of causal error-rate feedback, which diminishes with block-size n.

Steady-state goodput versus fading-rate parameter α:



The greedy algorithm extracts most of the goodput gain achievable from the use of causal error-rate feedback, which diminishes with fading rate.

Steady-state goodput versus mean SNR $\mathrm{E}\{\gamma_t\}$:



At SNR$= 25$ dB, the greedy algorithm performs about 1 dB worse than the causal genie; the fixed-rate scheme performs about 5 dB worse.

Finite Buffer Effects:

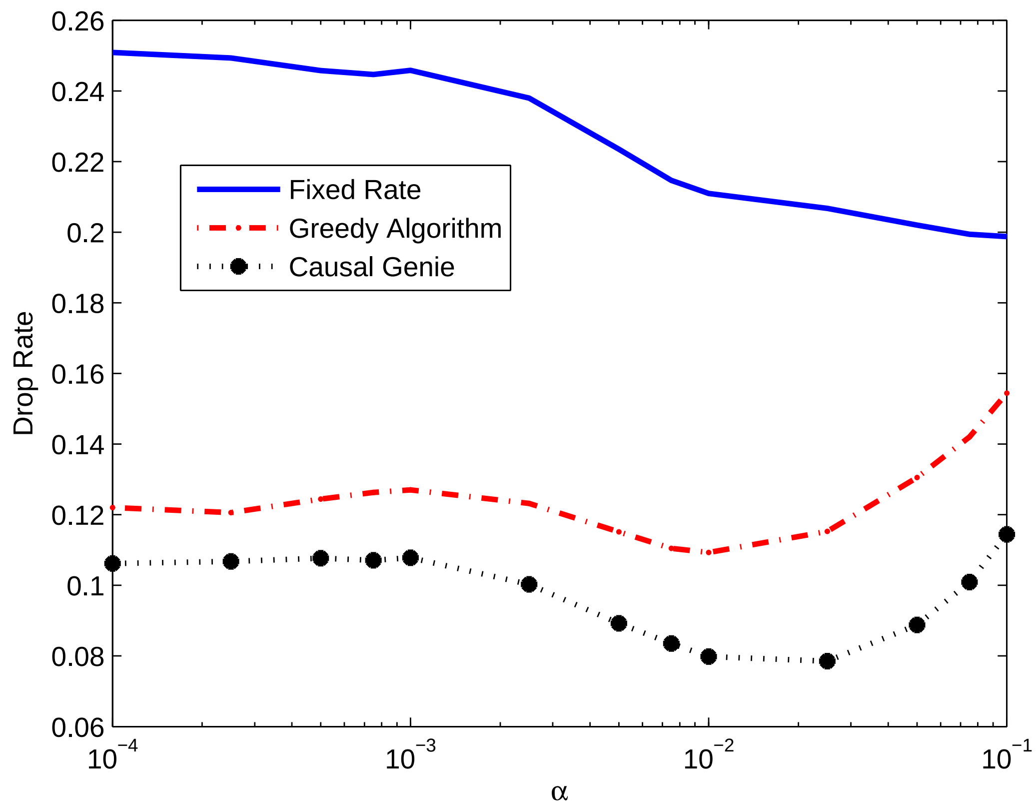
Buffer Description:

- Data arrives at a fixed rate of 516 bits per packet interval.
- Bits removed from buffer when transmitter receives an ACK.
- Overflowing bits are dropped from the buffer.

Simulation Setup:

- Same setup as before, including mean SNR $= 25$ dB.
  $\Rightarrow$ Fixed rate scheme uses $r_t^{\text{fr}} = 516$ bits/packet.
- Buffer size $= 30 \times 516$ bits.
- One trial $= 200$ packet intervals.
- Plots show an average of 500 trials.

Drop-rate versus fading-rate parameter α:



Slow fading (small α): SNR stays either "good" or "bad" over trial.
Fast fading (large α): SNR changes too quickly to track.
Moderate fading: adaptive algs track SNR and exploit its variation.

Conclusions:

- Motivation: Adapt physical-layer rate from link-layer ARQ.
- General problem: Rate adaptation via causal degraded error-rate feedback for maximization of total expected goodput.
- Optimal rate adaptation: POMDP... Too difficult!
- Practical rate adaptation: Greedy scheme.
- Upper bound on optimal: Causal genie. (Happens to be greedy!)
- Numerical experiments:
  - Steady-state goodput: Greedy scheme is much better than best fixed-rate scheme and not much worse than causal genie.
  - Achievable goodput diminishes as the feedback delay d, block-size n, or channel fading rate α increase.
  - Finite-buffer drop rate: greedy scheme much better than best fixed-rate scheme and not much worse than causal genie.

Thanks for listening!