Delay, feedback, and the price of ignorance

Anant Sahai
based in part on joint work with students:
Tunc Simsek   Cheng Chang

Wireless Foundations
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley

Major Support from NSF ITR

EPFL Summer Research Institute: July 18th, 2006

Shannon tells us

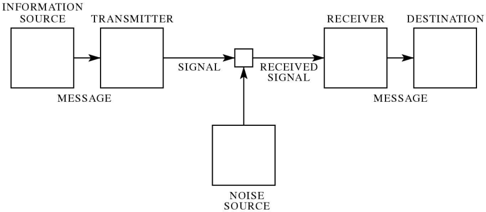


Fig. 1—Schematic diagram of a general communication system.

- Architectural implication: separate source and channel coding
- Delay is the most basic price of reliability

Shannon tells us

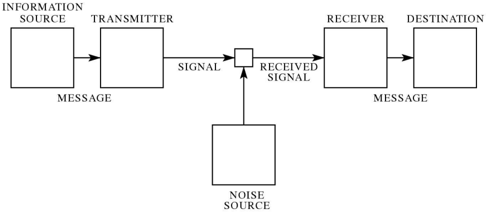


Fig. 1—Schematic diagram of a general communication system.

- Architectural implication: separate source and channel coding
- Delay is the most basic price of reliability

"[The duality between source and channel coding] can be pursued further and is related to a duality between past and future and the notions of control and knowledge. Thus we may have knowledge of the past and cannot control it; we may control the future but have no knowledge of it." — *Claude Shannon 1959*

Shannon tells us

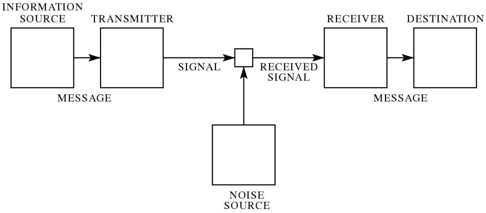


Fig. 1—Schematic diagram of a general communication system.

- Architectural implication: separate source and channel coding
- Delay is the most basic price of reliability

"[The duality between source and channel coding] can be pursued further and is related to a duality between past and future and the notions of control and knowledge. Thus we may have knowledge of the past and cannot control it; we may control the future but have no knowledge of it." — *Claude Shannon 1959*

- What did he mean?

Review of block coding

- Long block codes are the traditional info theory approach
  - Source: $X_1^n \longrightarrow B_1^{Rn} \longrightarrow \widehat{X}_1^n$
  - Channel: $B_1^{Rn} \longrightarrow Y_1^n \longrightarrow Z_1^n \longrightarrow \widehat{B}_1^{Rn}$

Review of block coding

- Long block codes are the traditional info theory approach
  - Source: $X_1^n \longrightarrow B_1^{Rn} \longrightarrow \widehat{X}_1^n$
  - Channel: $B_1^{Rn} \longrightarrow Y_1^n \longrightarrow Z_1^n \longrightarrow \widehat{B}_1^{Rn}$
- No real sense of time, except trivial interpretation
  - Source-coding: randomness is before encoding
  - Channel-coding: randomness is after encoding

Review of block coding

- Long block codes are the traditional info theory approach
  - Source: $X_1^n \longrightarrow B_1^{Rn} \longrightarrow \widehat{X}_1^n$
  - Channel: $B_1^{Rn} \longrightarrow Y_1^n \longrightarrow Z_1^n \longrightarrow \widehat{B}_1^{Rn}$
- No real sense of time, except trivial interpretation
  - Source-coding: randomness is before encoding
  - Channel-coding: randomness is after encoding
- Block error exponents: $P_e \propto \exp(-nE(R))$

Review of block coding

- Long block codes are the traditional info theory approach
  - Source: $X_1^n \rightarrow B_1^{Rn} \rightarrow \widehat{X}_1^n$
  - Channel: $B_1^{Rn} \rightarrow Y_1^n \rightarrow Z_1^n \rightarrow \widehat{B}_1^{Rn}$
- No real sense of time, except trivial interpretation
  - Source-coding: randomness is before encoding
  - Channel-coding: randomness is after encoding
- Block error exponents: $P_e \propto \exp(-nE(R))$
- Source coding:

$$
\begin{aligned}
E_b(R) &= \min_{Q:H(Q)\geq R} D(Q||P) \\
&= \sup_{\rho\geq 0} \rho R - E_0(\rho) \\
E_0(\rho) &= \ln[\sum_x P(x)^{\frac{1}{1+\rho}}]^{(1+\rho)}
\end{aligned}
$$

Review of block coding

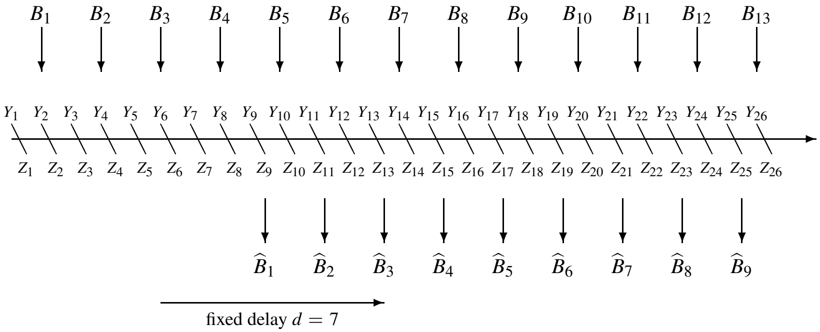
- Long block codes are the traditional info theory approach
  - Source: $X_1^n \longrightarrow B_1^{Rn} \longrightarrow \widehat{X}_1^n$
  - Channel: $B_1^{Rn} \longrightarrow Y_1^n \longrightarrow Z_1^n \longrightarrow \widehat{B}_1^{Rn}$
- No real sense of time, except trivial interpretation
  - Source-coding: randomness is before encoding
  - Channel-coding: randomness is after encoding
- Block error exponents: $P_e \propto \exp(-nE(R))$
- Channel "sphere-packing" bound:

$$
\begin{aligned}
E_{sp}(R) &= \max_{\vec{q}} \min_{G:I(\vec{q},G)\leq R} D\left(G||P|\vec{q}\right) \\
&= \sup_{\rho\geq 0} E_0(\rho) - \rho R \\
E_0(\rho) &= \max_{\vec{q}} - \ln \sum_z \left[\sum_y q_y p_{y,z}^{\frac{1}{1+\rho}} \right]^{(1+\rho)}
\end{aligned}
$$

Outline

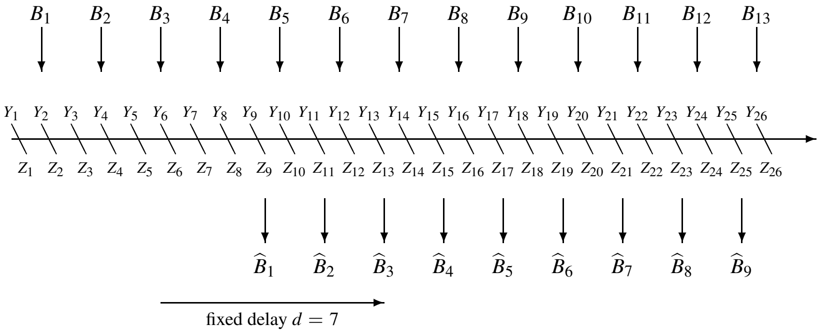
1. Motivation and introduction
2. Fixed-delay channel coding
   - Without feedback
   - The BEC example
   - The focusing bound
   - Approaching the focusing bound with feedback
3. The source-coding analog
   - Without side-information
   - With side-information
4. Conclusions

What about fixed delay?



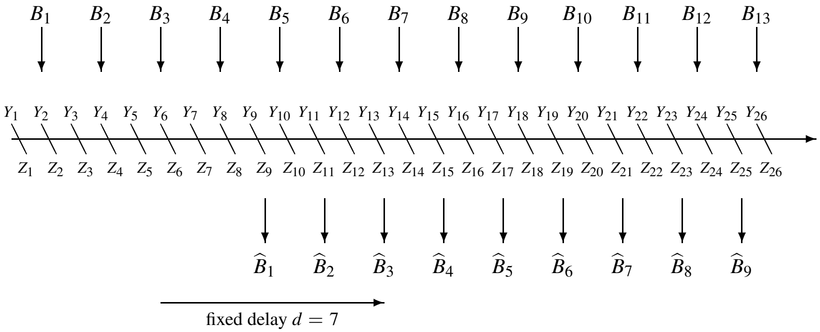
- Consider “hard” deadlines today. (“Soft” deadlines allow “erasures”)

What about fixed delay?



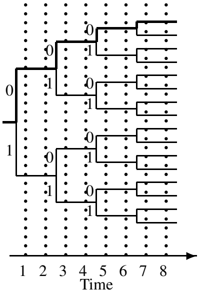
- Consider "hard" deadlines today. ("Soft" deadlines allow "erasures")
- Can achieve $E_r(R)$ with delay using convolutional codes.

What about fixed delay?



- Consider "hard" deadlines today. ("Soft" deadlines allow "erasures")
- Can achieve $E_r(R)$ with delay using convolutional codes.
- Pinsker (1967: PPI 3.4.44-55) claimed that the block-exponents continued to govern the non-block case *with and without feedback.*

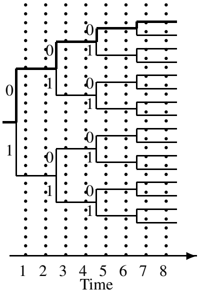
Nonblock codes without feedback



Infinite binary tree, with iid random labels:

- Choose a path through the tree based on data bits
- Transmit the path labels through the channel

Nonblock codes without feedback

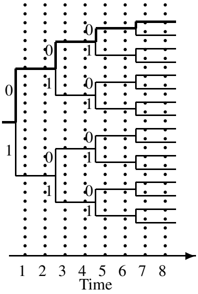


Infinite binary tree, with iid random labels:

- Choose a path through the tree based on data bits
- Transmit the path labels through the channel

- ML decoding
  - Disjoint paths are pairwise independent of the true path.
  - $E_r(R)$ analysis applies: *future* events dominate.

Nonblock codes without feedback

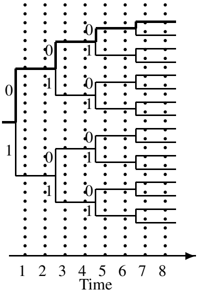


Infinite binary tree, with iid random labels:

- Choose a path through the tree based on data bits
- Transmit the path labels through the channel

- ML decoding
  - Disjoint paths are pairwise independent of the true path.
  - $E_r(R)$ analysis applies: *future* events dominate.
- Can implement with time-varying random convolutional code.

Nonblock codes without feedback



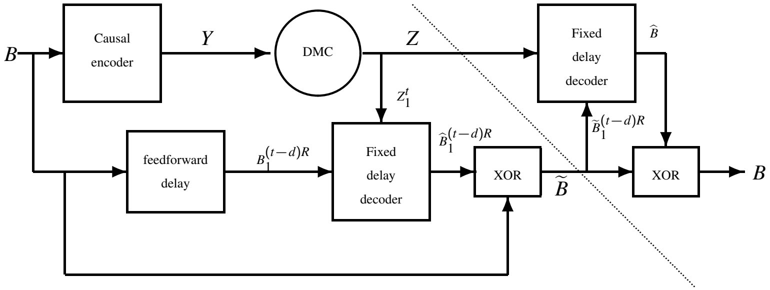
Infinite binary tree, with iid random labels:

- Choose a path through the tree based on data bits
- Transmit the path labels through the channel

- ML decoding
  - Disjoint paths are pairwise independent of the true path.
  - $E_r(R)$ analysis applies: *future* events dominate.
- Can implement with time-varying random convolutional code.
- **Achieves** $P_e(d) \leq K \exp(-E_r(R)d)$ **for every** d **for all** $R < C$

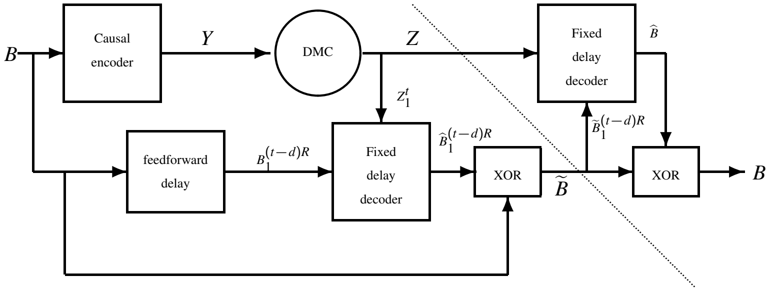
Pinsker's bounding construction explained

- Without feedback: $E_{sp}(R)$ continues to be a bound.
- Consider a code with target delay d
  - Use it to construct a block-code with blocksize $n >> d$
  - Genie-aided decoder: has the truth of all bits before i



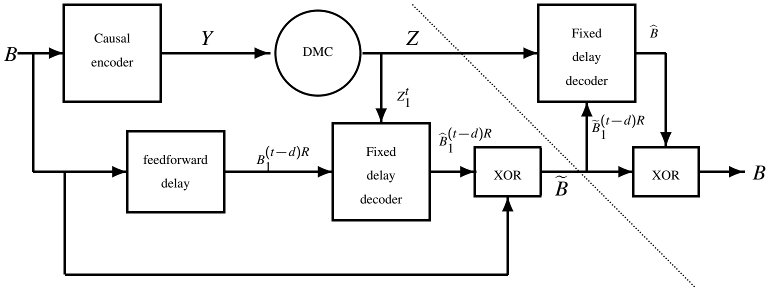
Pinsker's bounding construction explained

- Without feedback: $E_{sp}(R)$ continues to be a bound.
- Consider a code with target delay d
  - Use it to construct a block-code with blocksize $n >> d$
  - Genie-aided decoder: has the truth of all bits before i
  - Error events for genie-aided system depend only on last d

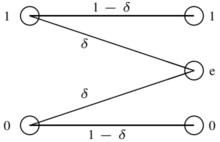


Pinsker's bounding construction explained

- Without feedback: $E_{sp}(R)$ continues to be a bound.
- Consider a code with target delay d
  - Use it to construct a block-code with blocksize $n >> d$
  - Genie-aided decoder: has the truth of all bits before i
  - Error events for genie-aided system depend only on last d
  - Apply a change of measure argument

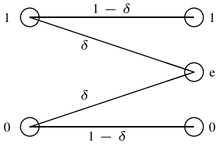


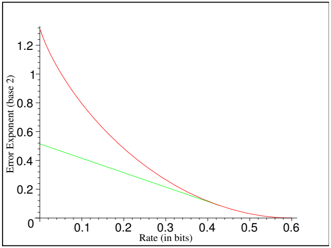
My favorite example: The BEC



- Simple capacity $1 - \delta$ bits per channel use
- With perfect feedback, simple to achieve: retransmit until it gets through

My favorite example: The BEC

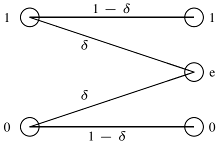




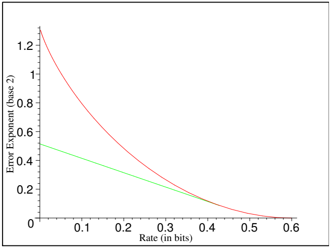
- Simple capacity $1 - \delta$ bits per channel use
- With perfect feedback, simple to achieve: retransmit until it gets through

- Classical bounds
  - Sphere-packing bound $D(1 - R||\delta)$
  - Random coding bound $\max_{\rho \in [0,1]} E_0(\rho) - \rho R$

My favorite example: The BEC



- Simple capacity $1 - \delta$ bits per channel use
- With perfect feedback, simple to achieve: retransmit until it gets through



- Classical bounds
  - Sphere-packing bound $D(1 - R||\delta)$
  - Random coding bound $\max_{\rho \in [0,1]} E_0(\rho) - \rho R$
- What happens with feedback?

BEC with feedback and fixed *blocks*

- At rate $R < 1$, have Rn bits to transmit in n channel uses.
- Typically $(1 - \delta)n$ code bits will be received.

BEC with feedback and fixed *blocks*

- At rate $R < 1$, have Rn bits to transmit in n channel uses.
- Typically $(1 - \delta)n$ code bits will be received.
- Block errors caused by atypical channel behavior.
  - Doomed if fewer than Rn bits arrive intact.

BEC with feedback and fixed *blocks*

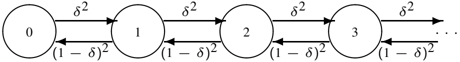
- At rate $R < 1$, have Rn bits to transmit in n channel uses.
- Typically $(1 - \delta)n$ code bits will be received.
- Block errors caused by atypical channel behavior.
  - Doomed if fewer than Rn bits arrive intact.
  - *Feedback can not save us.*
  - $D(1 - R||\delta)$

BEC with feedback and fixed *blocks*

- At rate $R < 1$, have Rn bits to transmit in n channel uses.
- Typically $(1-\delta)n$ code bits will be received.
- Block errors caused by atypical channel behavior.
  - Doomed if fewer than Rn bits arrive intact.
  - *Feedback can not save us.*
  - $D(1-R||\delta)$
- Dobrushin showed that this type of behavior is common.

BEC with feedback and fixed *delay*

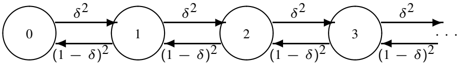
- $R = \frac{1}{2}$ example:



- Birth-death chain: positive recurrent if $\delta < \frac{1}{2}$

BEC with feedback and fixed *delay*

- $R = \frac{1}{2}$ example:

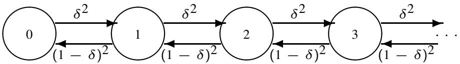


- Birth-death chain: positive recurrent if $\delta < \frac{1}{2}$
- Delay exponent easy to see:

$$P(D \geq d) = P(L > \frac{d}{2}) = K(\frac{\delta}{1-\delta})^d$$

BEC with feedback and fixed *delay*

- $R = \frac{1}{2}$ example:



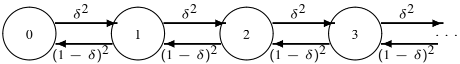
- Birth-death chain: positive recurrent if $\delta < \frac{1}{2}$
- Delay exponent easy to see:

$$P(D \geq d) = P(L > \frac{d}{2}) = K(\frac{\delta}{1-\delta})^d$$

- ≈ 0.584 vs 0.0294 for block-coding with $\delta = 0.4$

BEC with feedback and fixed *delay*

- $R = \frac{1}{2}$ example:



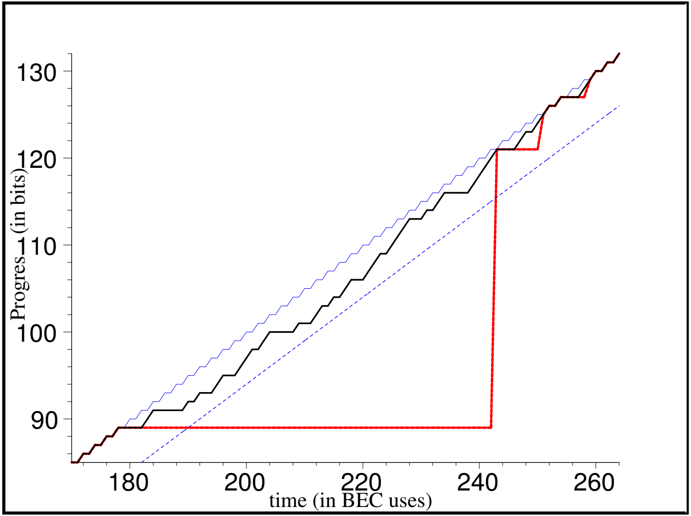
- Birth-death chain: positive recurrent if $\delta < \frac{1}{2}$
- Delay exponent easy to see:

$$P(D \geq d) = P(L > \frac{d}{2}) = K(\frac{\delta}{1-\delta})^d$$

- ≈ 0.584 vs 0.0294 for block-coding with $\delta = 0.4$

Pinsker was wrong!

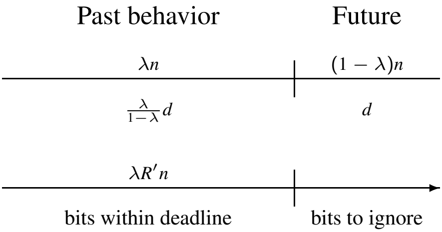
Where is this boost coming from?



Outline

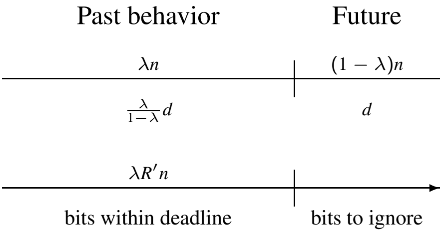
1. Motivation and introduction
2. Fixed-delay channel coding
   - Without feedback
   - The BEC example
   - The focusing bound
   - Approaching the focusing bound
3. The source-coding analog
   - Without side-information
   - With side-information
4. Conclusions

Using E_{sp} to bound α^* in general



- The block error probability is like $e^{-\alpha(1-\lambda)n}$ which cannot exceed the sphere-packing bound $e^{-E_{sp}(\lambda R)n}$

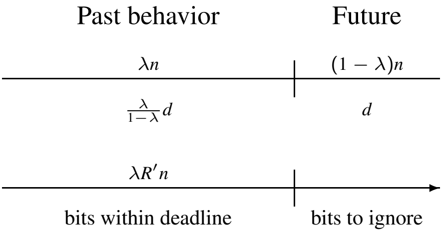
Using E_{sp} to bound α^* in general



- The block error probability is like $e^{-\alpha(1-\lambda)n}$ which cannot exceed the sphere-packing bound $e^{-E_{sp}(\lambda R)n}$

$$\alpha^*(R) \leq \frac{E_{sp}(\lambda R)}{1-\lambda}$$

Using E_{sp} to bound α^* in general



- The block error probability is like $e^{-\alpha(1-\lambda)n}$ which cannot exceed the sphere-packing bound $e^{-E_{sp}(\lambda R)n}$

$$\alpha^*(R) \leq \frac{E_{sp}(\lambda R)}{1-\lambda}$$

- The error events involve *both* the past and the future.

Uncertainty focusing bound for symmetric DMCs

Minimize over λ for symmetric DMCs to sweep out frontier by varying $\rho > 0$:

$$\begin{aligned} R(\rho) &= \frac{E_0(\rho)}{\rho} \\ E_a^+(\rho) &= E_0(\rho) \end{aligned}$$

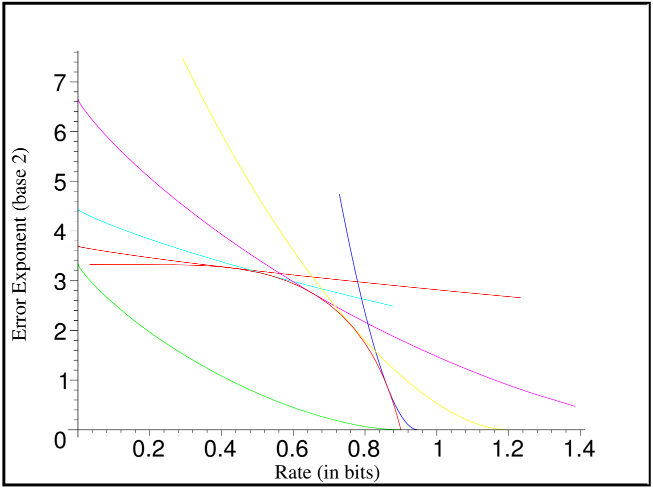
Uncertainty focusing bound for symmetric DMCs

Minimize over λ for symmetric DMCs to sweep out frontier by varying $\rho > 0$:

$$
\begin{aligned}
R(\rho) &= \frac{E_0(\rho)}{\rho} \\
E_a^+(\rho) &= E_0(\rho)
\end{aligned}
$$

Same form as Viterbi's "convolutional coding bound" for constraint-lengths, but a lot more fundamental!

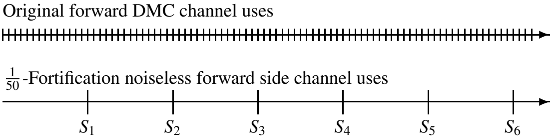
Upper bound tight for the BEC with feedback

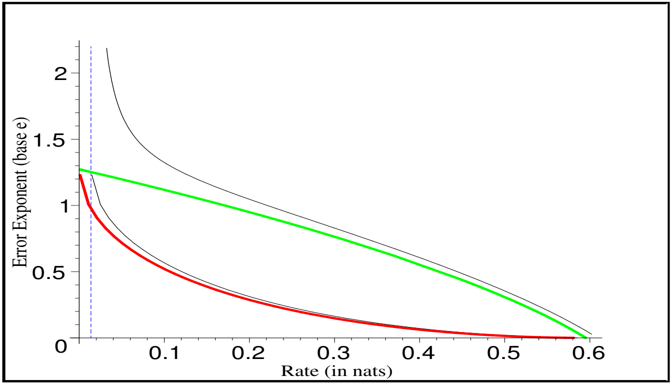


Outline

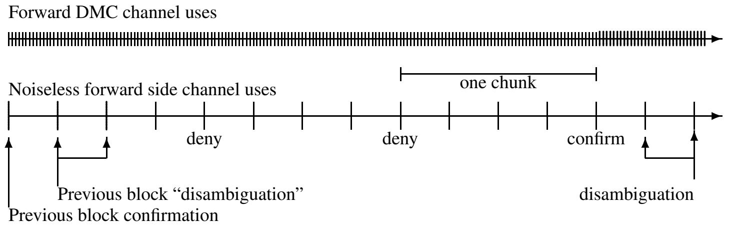
1. Motivation and introduction
2. Fixed-delay channel coding
   - Without feedback
   - The BEC example
   - The focusing bound
   - Approaching the focusing bound
3. The source-coding analog
   - Without side-information
   - With side-information
4. Conclusions

A spoonful of "sugar" helps the bits get across.





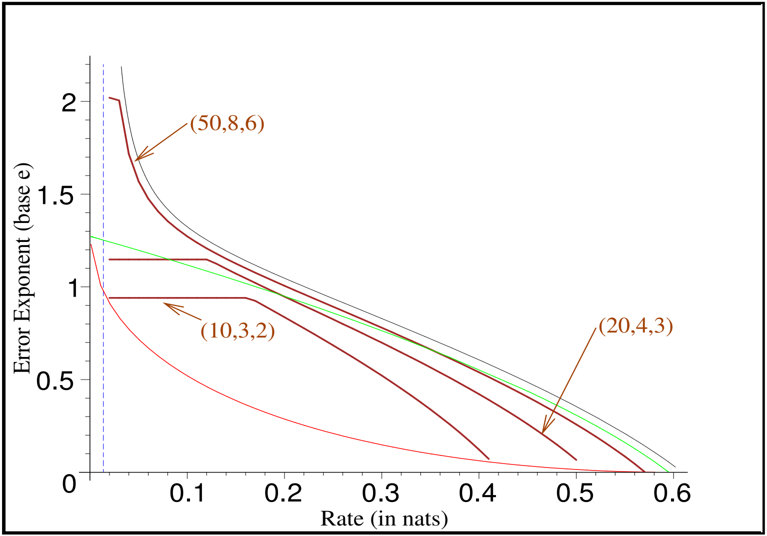
Harnessing the power of "flow control"



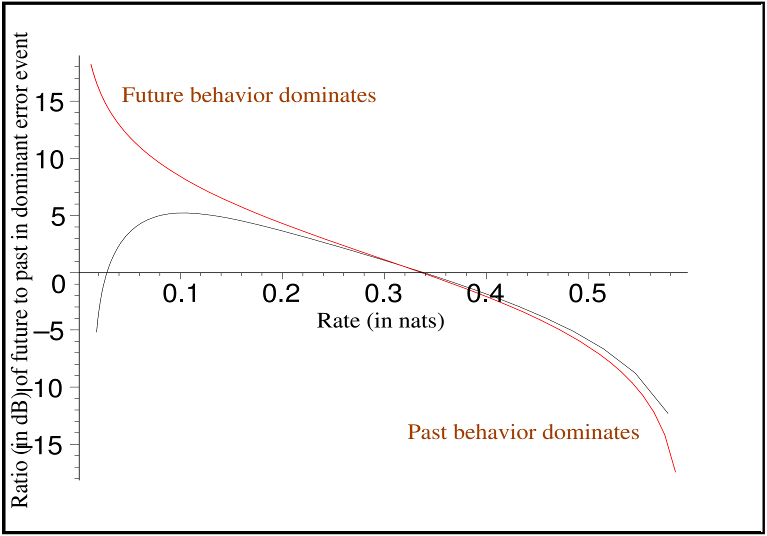
1. Group bits into miniblocks of size nR. ($n << d$)
2. Transmit using an ∞-length random codebook.
3. Use the "sugar" to tell decoder when it's done.

No decoding errors, just queuing plus transmission delays.

Approaching the focusing bound

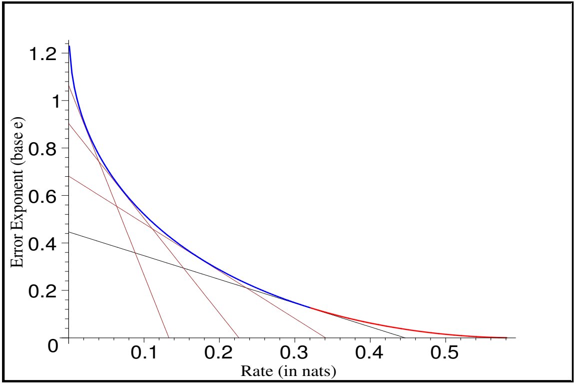


The dominant error events: past vs future



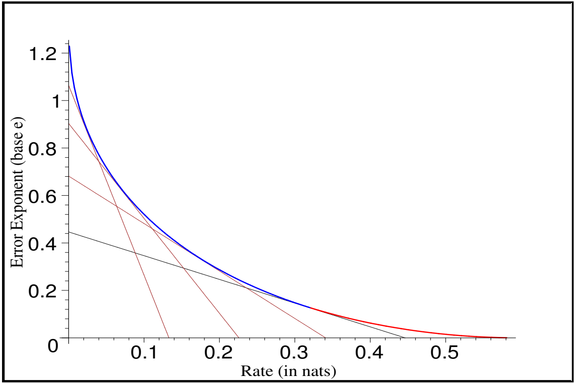
Why this works: operational interpretation of $E_0(\rho)$

Variable block transmission time T can be bounded by a constant plus a geometric random variable.



Why this works: operational interpretation of $E_0(\rho)$

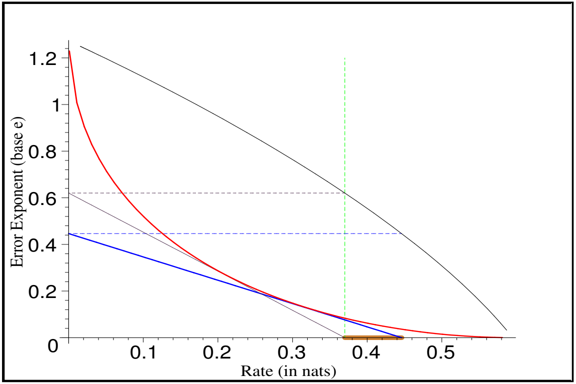
Variable block transmission time T can be bounded by a constant plus a geometric random variable.



Need to do list-decoding at low rates.

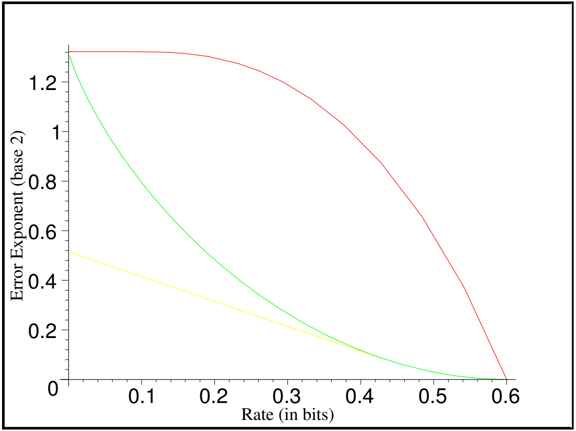
Reduces to the low-rate erasure case

Pick $R < R' < C$ and aim for $E^+(R') = E_0(\rho')$ exponent.



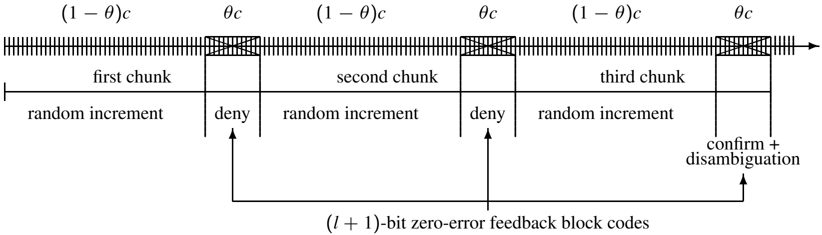
If n large, effective point-message rate $(n(1 - \frac{R}{R'}))^{-1}$ is small.

But low-rate erasure exponent $\approx -\log \delta$



$-\log \delta = E_0(\rho')$ in our context.

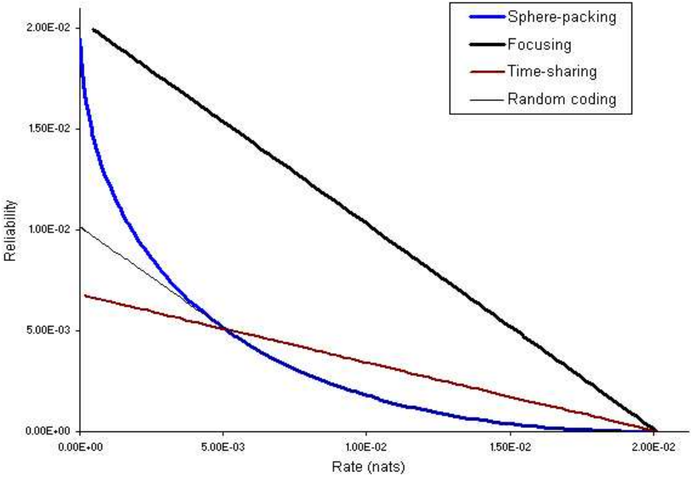
Channels with positive zero-error capacity



Throw away a fraction θ of channel uses for flow-control overhead
Asymptotically achieves the focusing bound as $\theta \to 0$.

Can do well even without "sugar"

Time-share flow-control and data and optimize fraction θ for flow-control.



Channel coding final comments

- Computation per channel use does not depend on probability of error = *infinite computational exponent*

Channel coding final comments

- Computation per channel use does not depend on probability of error = *infinite computational exponent*
- The code is “anytime” in that it is delay universal — application can pick what latency is desired.

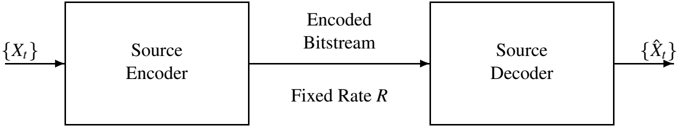
Channel coding final comments

- Computation per channel use does not depend on probability of error = *infinite computational exponent*
- The code is "anytime" in that it is delay universal — application can pick what latency is desired.
- Queuing delay dominates at all rates.
- Transmission delay exponents are bounded away from zero at all rates up to capacity. (partially explains Horstein's weird positive error exponents at capacity)

Outline

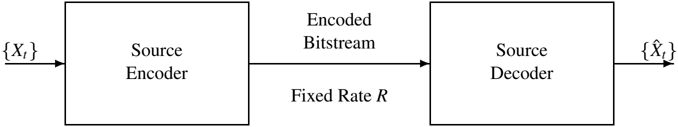
1. Motivation and introduction
2. Fixed-delay channel coding
   - Without feedback
   - The BEC example
   - The focusing bound
   - Approaching the focusing bound
3. The source-coding analog
   - Without side-information
   - With side-information
4. Conclusions

The source coding problem



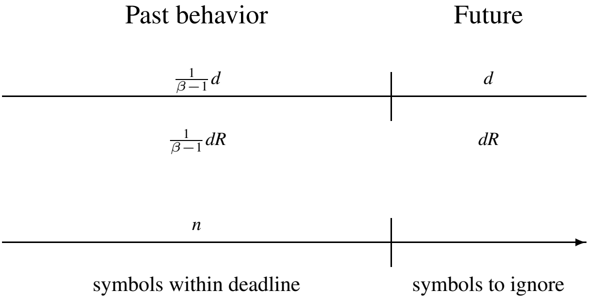
- *Assume* $\{X_t\}$ *iid*
- Application-level interface
  - Symbol error probability: $P_e = P(X_t \neq \hat{X}_t)$
  - End-to-end latency: d (measured in source timescale)
- Channel-code interface: fixed rate R (assumed noiseless)

The source coding problem



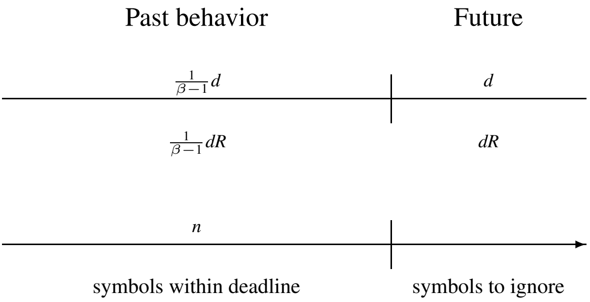
- *Assume* $\{X_t\}$ *iid*
- Application-level interface
  - Symbol error probability: $P_e = P(X_t \neq \hat{X}_t)$
  - End-to-end latency: d (measured in source timescale)
- Channel-code interface: fixed rate R (assumed noiseless)
- What are the fundamental tradeoffs?

Using E_b to bound E_s in general



The error probability is bounded by $K\exp(-dE_s(R))$ which cannot exceed the block-coding bound $\exp(-nE_b(\beta R))$

Using E_b to bound E_s in general



The error probability is bounded by $K \exp(-dE_s(R))$ which cannot exceed the block-coding bound $\exp(-nE_b(\beta R))$

$$E_s(R) \leq \frac{E_b(\beta R)}{\beta - 1}$$

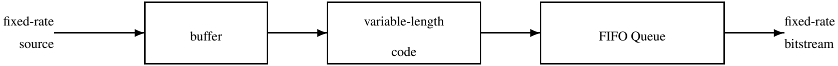
Only the past matters!

Achieving the focusing bound

$$
\begin{aligned}
R(\rho) &= \frac{E_0(\rho)}{\rho} \\
E_s(\rho) &= E_0(\rho)
\end{aligned}
$$

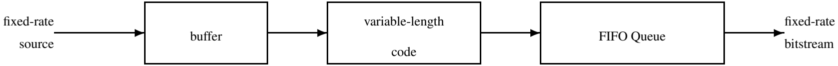
Achieving the focusing bound

$$
\begin{aligned}
R(\rho) &= \frac{E_0(\rho)}{\rho} \\
E_s(\rho) &= E_0(\rho)
\end{aligned}
$$



Achieving the focusing bound

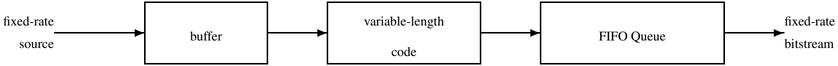
$$
\begin{aligned}
R(\rho) &= \frac{E_0(\rho)}{\rho} \\
E_s(\rho) &= E_0(\rho)
\end{aligned}
$$



- Pick miniblock n large enough but small relative to d
- Variable-length codes turn into variable delay at the receiver.

Achieving the focusing bound

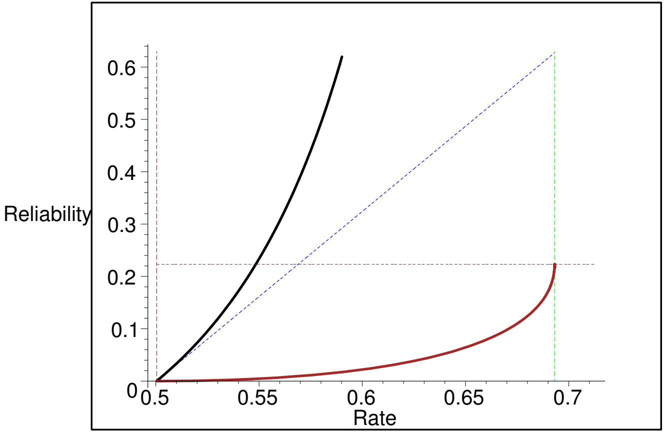
$$R(\rho) = \frac{E_0(\rho)}{\rho}$$
$$E_s(\rho) = E_0(\rho)$$



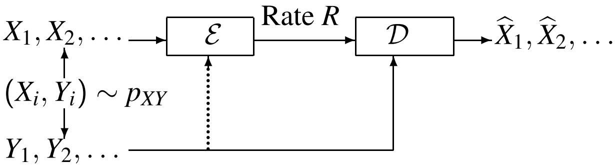
- Pick miniblock n large enough but small relative to d
- Variable-length codes turn into variable delay at the receiver.
- Queuing delay dominates

A simple example

- Unfair coin tosses $P(H) = 0.2$

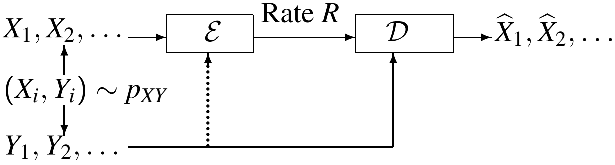


Side-information at the decoder

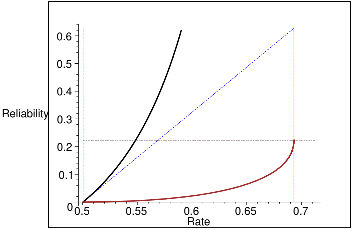


- Encoder may or may not be ignorant of the side-information
- If $P_{X,Y}$ symmetric with uniform marginals, can do no better than $E_b(R)$ with delay.

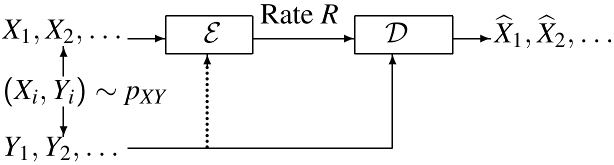
Side-information at the decoder



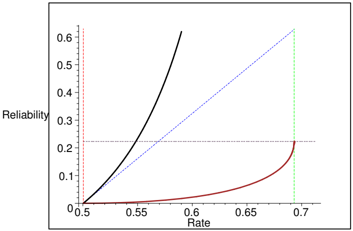
- Encoder may or may not be ignorant of the side-information
- If $P_{X,Y}$ symmetric with uniform marginals, can do no better than $E_b(R)$ with delay.

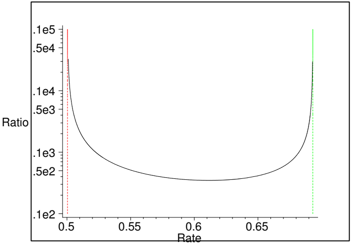


Side-information at the decoder

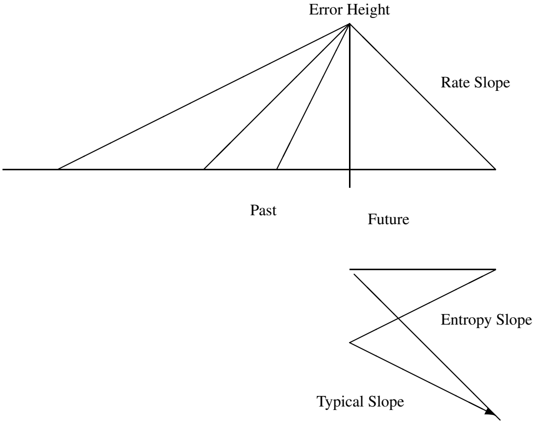


- Encoder may or may not be ignorant of the side-information
- If $P_{X,Y}$ symmetric with uniform marginals, can do no better than $E_b(R)$ with delay.





Long vs. Large deviations



- Shorter deviation periods must be larger.
- Smaller deviations must be over longer periods.

Conclusions

> *"[The duality between source and channel coding] can be pursued further and is related to a duality between past and future and the notions of control and knowledge. Thus we may have knowledge of the past and cannot control it; we may control the future but have no knowledge of it." — Claude Shannon 1959*

- Error events are dominated by the:
  - Future: Channel coding without feedback. [Pinsker]
  - Past: Point-to-point lossless source coding [ITW06]

Conclusions

> *"[The duality between source and channel coding] can be pursued further and is related to a duality between past and future and the notions of control and knowledge. Thus we may have knowledge of the past and cannot control it; we may control the future but have no knowledge of it."* — *Claude Shannon 1959*

- Error events are dominated by the:
  - Future: Channel coding without feedback. [Pinsker]
  - Past: Point-to-point lossless source coding [ITW06]
  - Future: Symmetric source coding with receiver side-information [ISIT06]

Conclusions

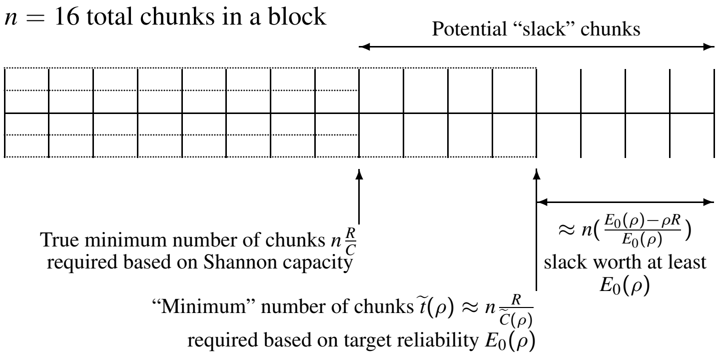
> *"[The duality between source and channel coding] can be pursued further and is related to a duality between past and future and the notions of control and knowledge. Thus we may have knowledge of the past and cannot control it; we may control the future but have no knowledge of it."* — *Claude Shannon 1959*

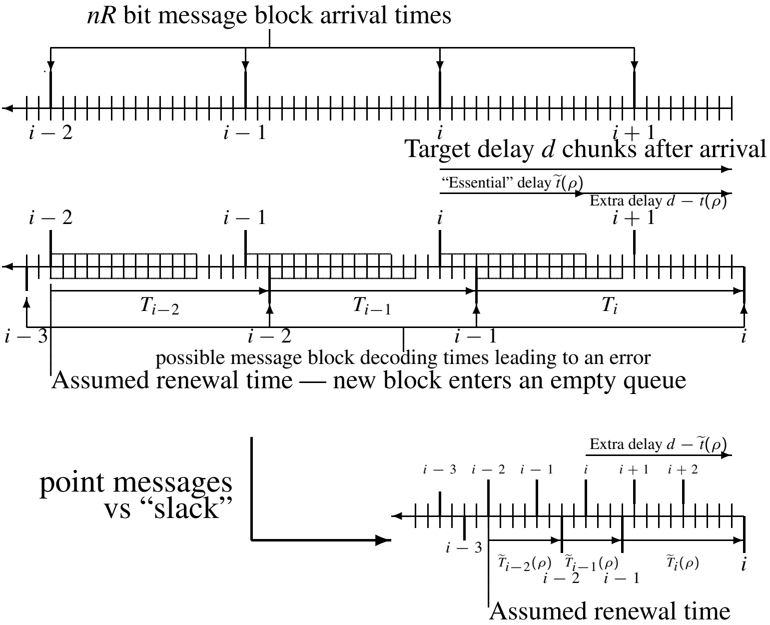
- Error events are dominated by the:
  - Future: Channel coding without feedback. [Pinsker]
  - Past: Point-to-point lossless source coding [ITW06]
  - Future: Symmetric source coding with receiver side-information [ISIT06]
  - Combination: Channel coding with feedback
  - Combination: Non-symmetric source-coding with receiver side-information.

Conclusions

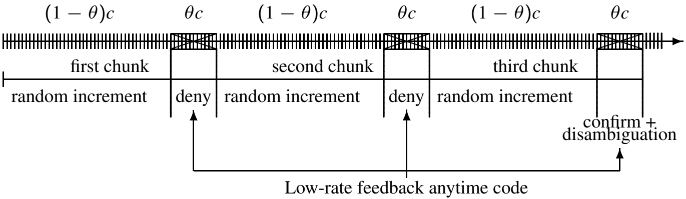
> *"[The duality between source and channel coding] can be pursued further and is related to a duality between past and future and the notions of control and knowledge. Thus we may have knowledge of the past and cannot control it; we may control the future but have no knowledge of it."* — *Claude Shannon 1959*

- Error events are dominated by the:
  - Future: Channel coding without feedback. [Pinsker]
  - Past: Point-to-point lossless source coding [ITW06]
  - Future: Symmetric source coding with receiver side-information [ISIT06]
  - Combination: Channel coding with feedback
  - Combination: Non-symmetric source-coding with receiver side-information.
- Architectural guidance:
  - Make your messages as short as possible while avoiding integer effects.
  - Use variable-length coding where you can
  - Use feedback for hybrid ARQ, not retransmissions
  - Use queues to smooth out your data rates


$n = 16$ total chunks in a block
Potential "slack" chunks
True minimum number of chunks $n\frac{R}{C}$
required based on Shannon capacity
$\approx n(\frac{E_0(\rho)-\rho R}{E_0(\rho)})$
slack worth at least
$E_0(\rho)$
"Minimum" number of chunks $\widetilde{t}(\rho) \approx n\frac{R}{\widetilde{C}(\rho)}$
required based on target reliability $E_0(\rho)$

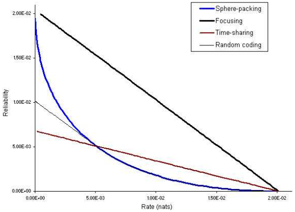

nR bit message block arrival times
i − 2
i − 1
i
i + 1
Target delay d chunks after arrival
"Essential" delay t̃(ρ)
Extra delay d − t̃(ρ)
T_{i−2}
T_{i−1}
T_i
i − 3
possible message block decoding times leading to an error
Assumed renewal time — new block enters an empty queue
point messages
vs "slack"
i − 3 i − 2 i − 1 i i + 1 i + 2
T̃_{i−2}(ρ) T̃_{i−1}(ρ) T̃_i(ρ)
Assumed renewal time

Optimize fraction θ for flow-control



Flow-control encoded with "∞-length" convolutional code.

- Flow-control exponent $\approx \theta E_0(1)$
- Data exponent $\approx (1-\theta)E_0(\rho)$
- Data rate $\approx (1-\theta)\frac{E_0(\rho)}{\rho}$



$$\theta^* = \frac{E_0(\rho)}{E_0(1)+E_0(\rho)}$$

$$E_0^*(\rho) = \frac{E_0(\rho)E_0(1)}{E_0(\rho)+E_0(1)}$$

$$R^*(\rho) = \frac{E_0^*(\rho)}{\rho}$$

$$E^*(\rho) = E_0^*(\rho)$$

Low rate feedback convolutional codes

- At $R < E_0(1)$, sequential decoding expands only a finite number of nodes on average.
- Each expansion costs the (growing) constraint length.

Low rate feedback convolutional codes

- At $R < E_0(1)$, sequential decoding expands only a finite number of nodes on average.
- Each expansion costs the (growing) constraint length.
- Idea: run a copy of the decoder at the encoder

Low rate feedback convolutional codes

- At $R < E_0(1)$, sequential decoding expands only a finite number of nodes on average.
- Each expansion costs the (growing) constraint length.
- Idea: run a copy of the decoder at the encoder
  - Convolve against: $(B_1 + \widehat{B}_1(n)), (B_2 + \widehat{B}_2(n)), \ldots, (B_{n-1} + \widehat{B}_{n-1}(n)), B_n$

Low rate feedback convolutional codes

- At $R < E_0(1)$, sequential decoding expands only a finite number of nodes on average.
- Each expansion costs the (growing) constraint length.
- Idea: run a copy of the decoder at the encoder
  - Convolve against: $(B_1 + \widehat{B}_1(n)), (B_2 + \widehat{B}_2(n)), \ldots, (B_{n-1} + \widehat{B}_{n-1}(n)), B_n$
  - Identical distance properties

Low rate feedback convolutional codes

- At $R < E_0(1)$, sequential decoding expands only a finite number of nodes on average.
- Each expansion costs the (growing) constraint length.
- Idea: run a copy of the decoder at the encoder
  - Convolve against: $(B_1 + \widehat{B}_1(n)), (B_2 + \widehat{B}_2(n)), \ldots, (B_{n-1} + \widehat{B}_{n-1}(n)), B_n$
  - Identical distance properties
  - But only a finite number of expected nonzero terms
  - Infinite-constraint length performance at a finite price!

Low rate feedback convolutional codes

- At $R < E_0(1)$, sequential decoding expands only a finite number of nodes on average.
- Each expansion costs the (growing) constraint length.
- Idea: run a copy of the decoder at the encoder
  - Convolve against: $(B_1 + \widehat{B}_1(n)), (B_2 + \widehat{B}_2(n)), \ldots, (B_{n-1} + \widehat{B}_{n-1}(n)), B_n$
  - Identical distance properties
  - But only a finite number of expected nonzero terms
  - Infinite-constraint length performance at a finite price!
- *Achieves $E_r(R)$ exponent with delay.*

Low rate feedback convolutional codes

- At $R < E_0(1)$, sequential decoding expands only a finite number of nodes on average.
- Each expansion costs the (growing) constraint length.
- Idea: run a copy of the decoder at the encoder
  - Convolve against: $(B_1 + \widehat{B}_1(n)), (B_2 + \widehat{B}_2(n)), \ldots, (B_{n-1} + \widehat{B}_{n-1}(n)), B_n$
  - Identical distance properties
  - But only a finite number of expected nonzero terms
  - Infinite-constraint length performance at a finite price!
- *Achieves $E_r(R)$ exponent with delay.*
- Another trick due to Pinsker can extend computational advantage to higher rates at the cost of lower delay exponents.