Feedback and Side-Information in Information Theory

Anant Sahai and Sekhar Tatikonda

UC Berkeley and Yale
sahai@eecs.berkeley.edu and sekhar.tatikonda@yale.edu

ISIT 2007 Tutorial T2
Nice, France
June 24, 2007

Our Collaborators

- Vivek Borkar (TIFR)
- Cheng Chang (Berkeley)
- Giacomo Como (Torino)
- Stark Draper (Wisconsin)
- Nicola Elia (Iowa)
- Alek Kavcic (Hawaii)
- Jialing Liu (Iowa)
- Sanjoy Mitter (MIT)
- Hari Palaiyanur (Berkeley)
- Tunc Simsek (Mathworks)
- Shaohua Yang (Marvell)
- Serdar Yuksel (Queens)

Outline

Motivation

We want to engineer reliable and robust communication systems that appropriately share limited communication resources to deliver high performance at reasonable cost.

Motivation

We want to engineer reliable and robust communication systems that appropriately share limited communication resources to deliver high performance at reasonable cost.

What is the role of information theory?

Motivation

We want to engineer reliable and robust communication systems that appropriately share limited communication resources to deliver high performance at reasonable cost.

What is the role of information theory?

- Determine fundamental limits to performance

Motivation

We want to engineer reliable and robust communication systems that appropriately share limited communication resources to deliver high performance at reasonable cost.

What is the role of information theory?

- Determine fundamental limits to performance
  - Develop metrics that reflect the goals

Motivation

We want to engineer reliable and robust communication systems that appropriately share limited communication resources to deliver high performance at reasonable cost.

What is the role of information theory?

- Determine fundamental limits to performance
  - Develop metrics that reflect the goals
  - Give *computable* targets to aim for (e.g. asymptotic capacity and scaling)

Motivation

We want to engineer reliable and robust communication systems that appropriately share limited communication resources to deliver high performance at reasonable cost.

What is the role of information theory?

- Determine fundamental limits to performance
  - Develop metrics that reflect the goals
  - Give *computable* targets to aim for (e.g. asymptotic capacity and scaling)
  - Identify bottlenecks and key constraints

Motivation

We want to engineer reliable and robust communication systems that appropriately share limited communication resources to deliver high performance at reasonable cost.

What is the role of information theory?

- Determine fundamental limits to performance
  - Develop metrics that reflect the goals
  - Give *computable* targets to aim for (e.g. asymptotic capacity and scaling)
  - Identify bottlenecks and key constraints
- Guide overall system architecture

Motivation

We want to engineer reliable and robust communication systems that appropriately share limited communication resources to deliver high performance at reasonable cost.

What is the role of information theory?

- Determine fundamental limits to performance
  - Develop metrics that reflect the goals
  - Give *computable* targets to aim for (e.g. asymptotic capacity and scaling)
  - Identify bottlenecks and key constraints
- Guide overall system architecture
- Develop *nonasymptotic* algorithms that are *robustly implementable*.

Feedback and side-information

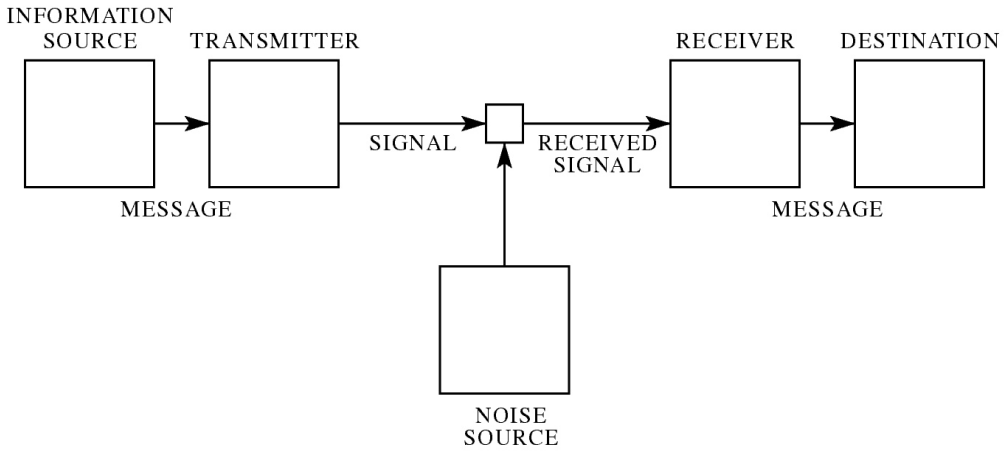


Fig. 1 — Schematic diagram of a general communication system.

- Communication is idealized as a one-way flow of information.

Feedback and side-information

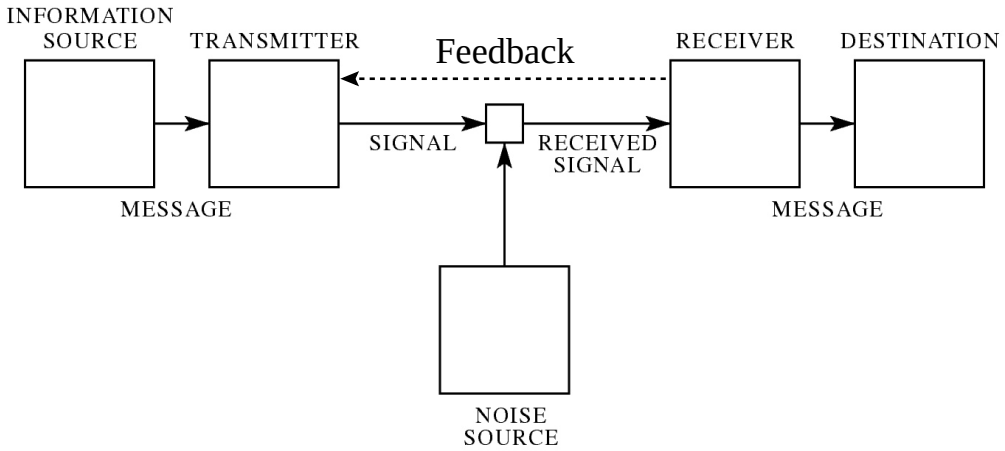


Fig. 1 — Schematic diagram of a general communication system.

- Communication is idealized as a one-way flow of information.
- But the medium usually supports a path in reverse.
- How should it be used?

Feedback and side-information

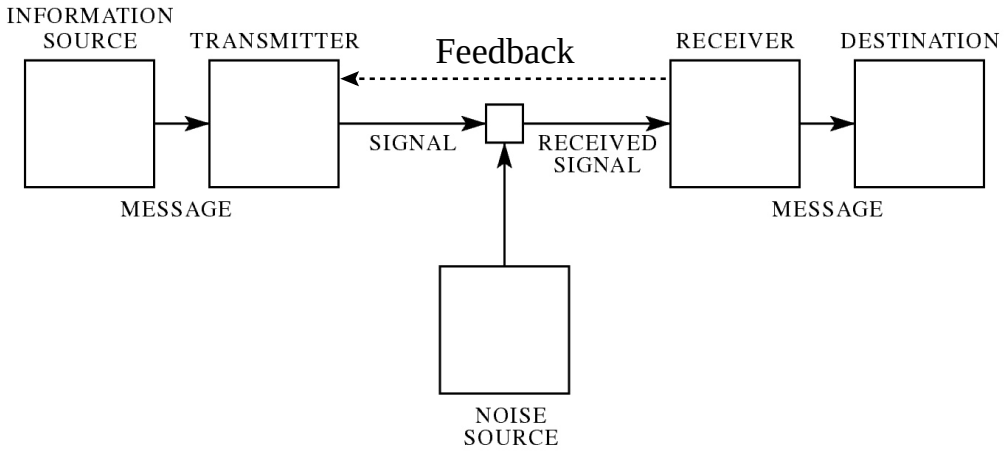


Fig. 1 — Schematic diagram of a general communication system.

- Communication is idealized as a one-way flow of information.
- But the medium usually supports a path in reverse.
- How should it be used?
  - Left alone for other users.

Feedback and side-information

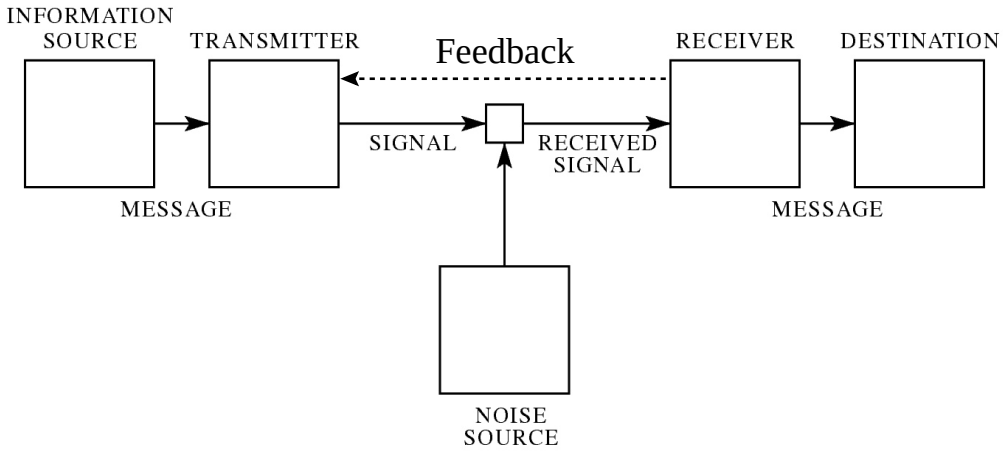


Fig. 1—Schematic diagram of a general communication system.

- Communication is idealized as a one-way flow of information.
- But the medium usually supports a path in reverse.
- How should it be used?
  - Left alone for other users.
  - Reduce communication needs. (Yao ’79, Orlitsky/El-Gamal ’84,’90,’92)

Feedback and side-information

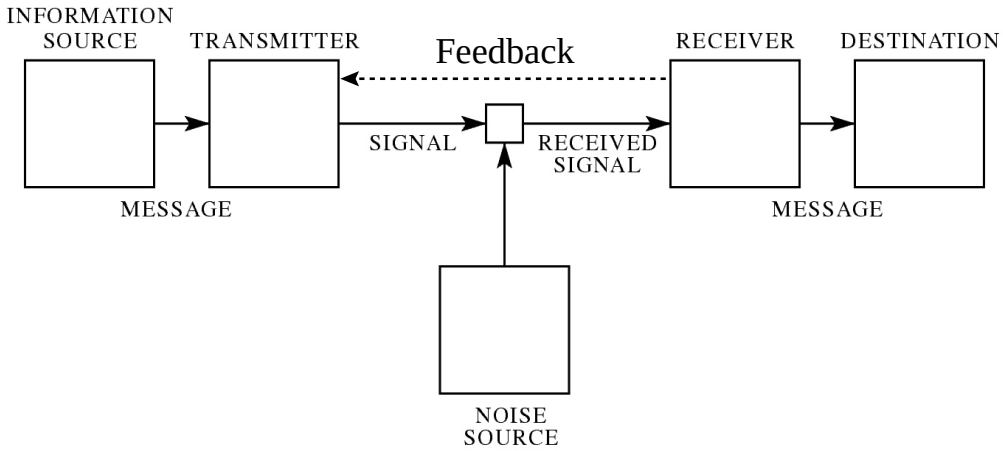


Fig. 1—Schematic diagram of a general communication system.

- Communication is idealized as a one-way flow of information.
- But the medium usually supports a path in reverse.
- How should it be used?
  - Left alone for other users.
  - Reduce communication needs. (Yao ’79, Orlitsky/El-Gamal ’84,’90,’92)
  - For learning/adaptation/universality.

Feedback and side-information

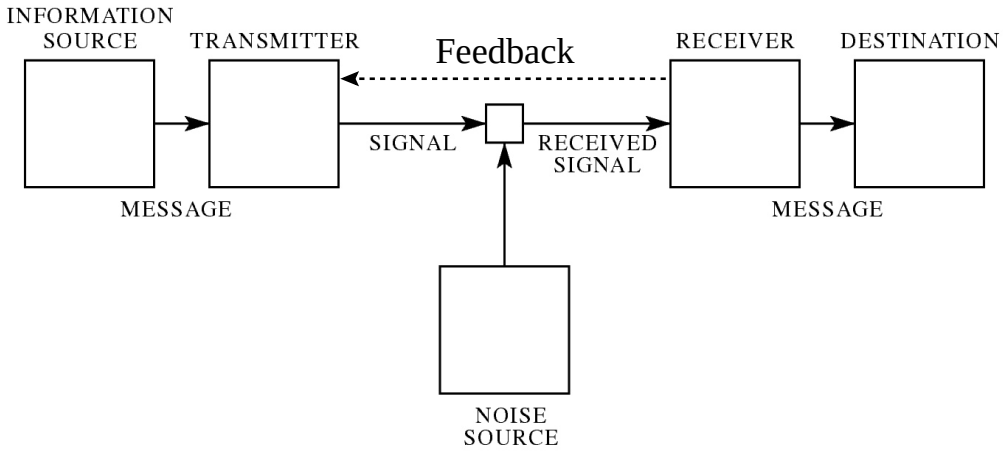


Fig. 1—Schematic diagram of a general communication system.

- Communication is idealized as a one-way flow of information.
- But the medium usually supports a path in reverse.
- How should it be used?
  - Left alone for other users.
  - Reduce communication needs. (Yao '79, Orlitsky/El-Gamal '84,'90,'92)
  - For learning/adaptation/universality.
  - Improve QoS: delay, P_e, rate.

Feedback and side-information

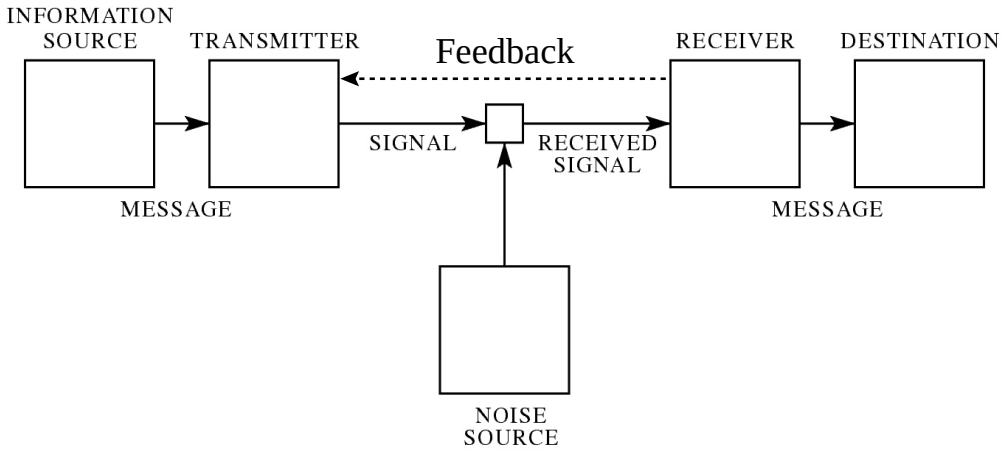


Fig. 1—Schematic diagram of a general communication system.

- Communication is idealized as a one-way flow of information.
- But the medium usually supports a path in reverse.
- How should it be used?
  - Left alone for other users.
  - Reduce communication needs. (Yao '79, Orlitsky/El-Gamal '84,'90,'92)
  - For learning/adaptation/universality.
  - Improve QoS: delay, P_e, rate.
  - Simplify implementation.

Reason for hope: Schalkwijk/Kailath ('68) scheme

$$
\begin{aligned}
Y_t &= X_t + V_t \text{ where } \{V_t\} \text{ iid } N(0,1) \\
E[X_t^2] &\leq P \\
X_t &= f_t(m, Y_0^{t-1})
\end{aligned}
$$

Reason for hope: Schalkwijk/Kailath ('68) scheme

$$\begin{aligned} Y_t &= X_t + V_t \text{ where } \{V_t\} \text{ iid } N(0,1) \\ E[X_t^2] &\leq P \\ X_t &= f_t(m, Y_0^{t-1}) \end{aligned}$$

- Encode message m using 2^{nR}-level PAM into X_0.
- Use feedback to capture first noise Gaussian V_0.

Reason for hope: Schalkwijk/Kailath ('68) scheme

$$\begin{aligned} Y_t &= X_t + V_t \text{ where } \{V_t\} \text{ iid } N(0,1) \\ E[X_t^2] &\leq P \\ X_t &= f_t(m, Y_0^{t-1}) \end{aligned}$$

- Encode message m using 2^{nR}-level PAM into X_0.
- Use feedback to capture first noise Gaussian V_0.
- Recursively MMSE estimate that Gaussian.
  - Channel-input: linear scaling of MMSE error
  - Optimal *factor* reduction in error with each use.

Reason for hope: Schalkwijk/Kailath ('68) scheme

$$\begin{aligned} Y_t &= X_t + V_t \text{ where } \{V_t\} \text{ iid } N(0,1) \\ E[X_t^2] &\leq P \\ X_t &= f_t(m, Y_0^{t-1}) \end{aligned}$$

- Encode message m using 2^{nR}-level PAM into X_0.
- Use feedback to capture first noise Gaussian V_0.
- Recursively MMSE estimate that Gaussian.
  - Channel-input: linear scaling of MMSE error
  - Optimal *factor* reduction in error with each use.
- Subtract noise estimate from original Y_0 and quantize.
  - Error if MMSE error is larger than quantization bin.
  - Probability exponentially small in MMSE error standard deviation.

Reason for hope: Schalkwijk/Kailath ('68) scheme

$$\begin{aligned} Y_t &= X_t + V_t \text{ where } \{V_t\} \text{ iid } N(0,1) \\ E[X_t^2] &\leq P \\ X_t &= f_t(m, Y_0^{t-1}) \end{aligned}$$

- Encode message m using 2^{nR}-level PAM into X_0.
- Use feedback to capture first noise Gaussian V_0.
- Recursively MMSE estimate that Gaussian.
  - Channel-input: linear scaling of MMSE error
  - Optimal *factor* reduction in error with each use.
- Subtract noise estimate from original Y_0 and quantize.
  - Error if MMSE error is larger than quantization bin.
  - Probability exponentially small in MMSE error standard deviation.
- Double-exponential decay of P_e and very simple scheme!

Reasons for despair: memoryless channels

- Even the S/K scheme does not beat $\frac{1}{2}\log(1+\frac{P}{N})$

Reasons for despair: memoryless channels

- Even the S/K scheme does not beat $\frac{1}{2}\log(1+\frac{P}{N})$
- Shannon proved that capacity does not increase for general memoryless channels with even perfect feedback.

Reasons for despair: memoryless channels

- Even the S/K scheme does not beat $\frac{1}{2}\log(1+\frac{P}{N})$
- Shannon proved that capacity does not increase for general memoryless channels with even perfect feedback.
  - Intuition: consider BEC case.

Reasons for despair: memoryless channels

- Even the S/K scheme does not beat $\frac{1}{2}\log(1+\frac{P}{N})$
- Shannon proved that capacity does not increase for general memoryless channels with even perfect feedback.
  - Intuition: consider BEC case.
  - Proof trick: consider mutual-information between uniformly-drawn *message* and channel outputs.

Reasons for despair: memoryless channels

- Even the S/K scheme does not beat $\frac{1}{2}\log(1+\frac{P}{N})$
- Shannon proved that capacity does not increase for general memoryless channels with even perfect feedback.
  - Intuition: consider BEC case.
  - Proof trick: consider mutual-information between uniformly-drawn *message* and channel outputs.
- What is wrong with $I(X_1^n; Y_1^n)$?
  - Consider X, Y disconnected. Y is just random noise. $C = 0$

Reasons for despair: memoryless channels

- Even the S/K scheme does not beat $\frac{1}{2}\log(1+\frac{P}{N})$
- Shannon proved that capacity does not increase for general memoryless channels with even perfect feedback.
  - Intuition: consider BEC case.
  - Proof trick: consider mutual-information between uniformly-drawn *message* and channel outputs.
- What is wrong with $I(X_1^n; Y_1^n)$?
  - Consider X, Y disconnected. Y is just random noise. $C = 0$
  - Use $X_i = Y_{i-1}$ as encoding. $I(X_1^n; Y_1^n) = (n-1)H(Y) > 0$.

It gets worse

"Feedback communications was an area of intense activity in 1968...A number of authors had shown constructive, even simple, schemes using noiseless feedback to achieve Shannon-like behavior...The situation in 1973 is dramatically different...The subject itself seems to be a burned out case...

In extending the simple noiseless feedback model to allow for more realistic situations, such as noisy feedback channels, bandlimited channels, and peak power constraints, theorists discovered a certain "brittleness" or sensitivity in their previous results..."

Robert Lucky (1973)

Outline

1. Motivation and background
2. Feedback, delay, and error probability: memoryless channels
   - Idealized cases with “magical” feedback
     - ★ Block-oriented results
     - ★ Fixed-delay with hard deadlines
     - ★ Fixed-delay with soft deadlines
   - Unreliable/Noisy feedback
3. Feedback and memory
4. Conclusions and open problems

Shannon's Prophecy:

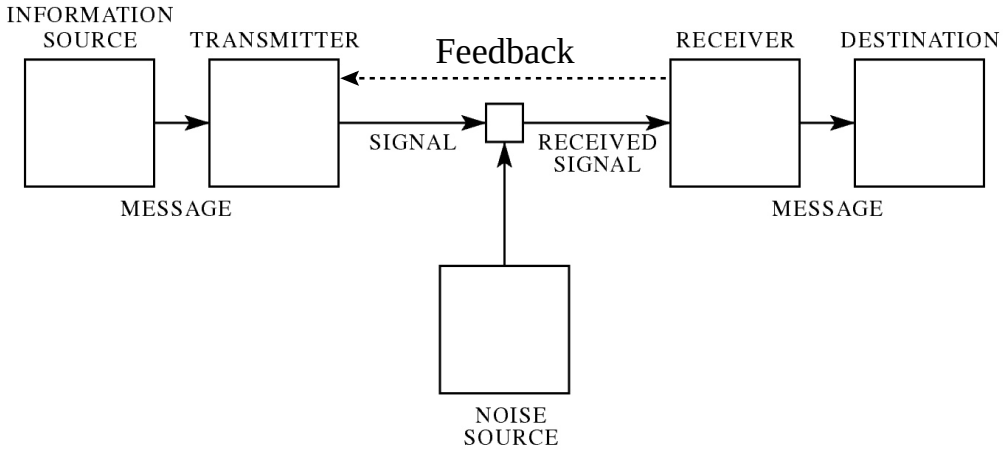


Fig. 1—Schematic diagram of a general communication system.

- 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 '59*

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: $B_1^{Rn} \rightarrow X_1^n \rightarrow Y_1^n \rightarrow \widehat{B}_1^{Rn}$

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: $B_1^{Rn} \rightarrow X_1^n \rightarrow Y_1^n \rightarrow \widehat{B}_1^{Rn}$
- No real sense of time, except trivial interpretation
  - Source-coding: randomness is before encoding
  - Channel-coding: randomness is after encoding

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: $B_1^{Rn} \rightarrow X_1^n \rightarrow Y_1^n \rightarrow \widehat{B}_1^{Rn}$
- No real sense of time, except trivial interpretation
  - Source-coding: randomness is before encoding
  - Channel-coding: randomness is after encoding
- Block error exponents: $P_e \propto \exp(-nE(R))$

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: $B_1^{Rn} \rightarrow X_1^n \rightarrow Y_1^n \rightarrow \widehat{B}_1^{Rn}$
- 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: $X_1^n \rightarrow B_1^{Rn} \rightarrow \widehat{X}_1^n$
  - Channel: $B_1^{Rn} \rightarrow X_1^n \rightarrow Y_1^n \rightarrow \widehat{B}_1^{Rn}$
- No real sense of time, except trivial interpretation
  - Source-coding: randomness is before encoding
  - Channel-coding: randomness is after encoding
- Block error exponents: $P_e \propto \exp(-nE(R))$
- Generic channels: (Haroutunian '77)

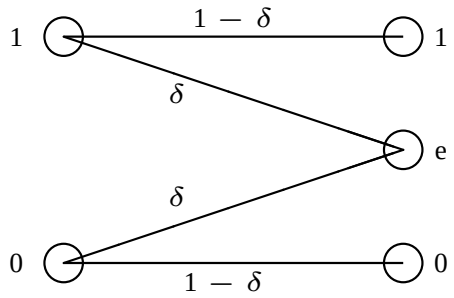
$$E^+(R) \quad = \quad \inf_{G:C(G)<R} \max_{\vec{q}} D\left(G||P|\vec{q}\right)$$

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: $B_1^{Rn} \rightarrow X_1^n \rightarrow Y_1^n \rightarrow \widehat{B}_1^{Rn}$
- No real sense of time, except trivial interpretation
  - Source-coding: randomness is before encoding
  - Channel-coding: randomness is after encoding
- Block error exponents: $P_e \propto \exp(-nE(R))$
- Channel "sphere-packing" bound: (Haroutunian '68, Blahut '74)

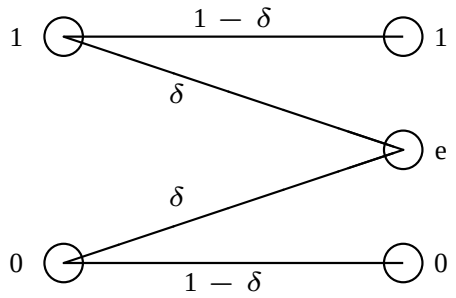
$$\begin{aligned} E_{sp}(R) &= \max_{\vec{q}} \min_{G:I(\vec{q},G)\leq R} D\left(G||P|\vec{q}\right) \\ &= \sup_{\rho\geq 0} E_0(\rho) - \rho R \\ E_0(\rho) &= \max_{\vec{q}} -\ln \sum_y \left[\sum_x q_x p_{y|x}^{\frac{1}{1+\rho}} \right]^{(1+\rho)} \end{aligned}$$

My favorite example: the BEC

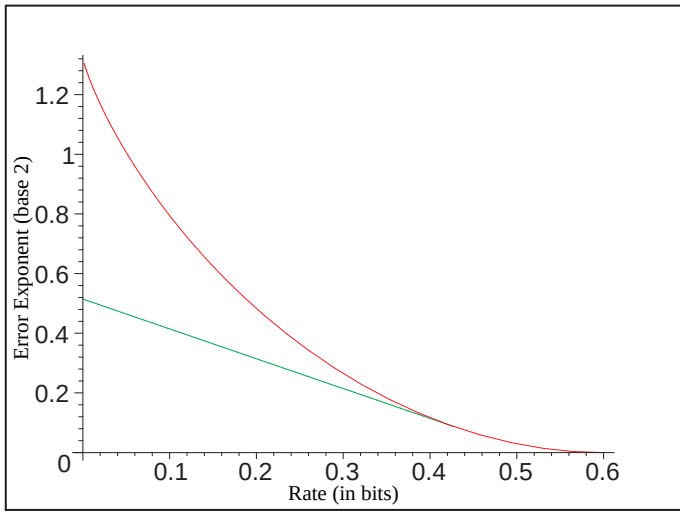


- Simple capacity $1 - \delta$ bits per channel use
- With perfect feedback, simple to achieve: retransmit until it gets through
  - Time till success: Geometric$(1 - \delta)$
  - Expected time to get through: $\frac{1}{1-\delta}$

My favorite example: the BEC

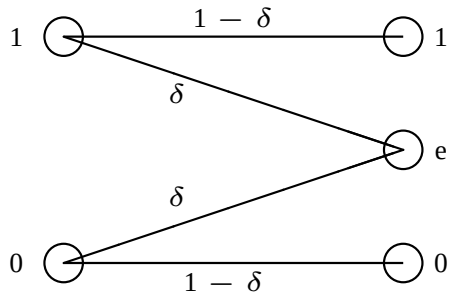


- Simple capacity $1-\delta$ bits per channel use
- With perfect feedback, simple to achieve: retransmit until it gets through
  - Time till success: Geometric$(1-\delta)$
  - Expected time to get through: $\frac{1}{1-\delta}$

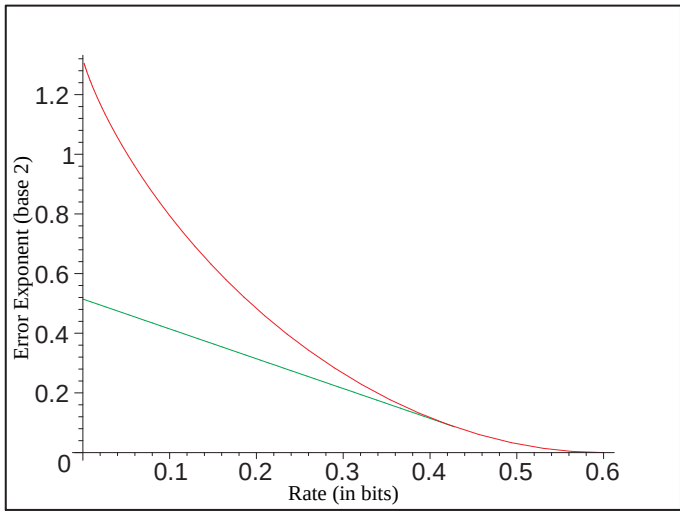


- Classical bounds
  - Sphere-packing bound $D(1-R||\delta)$
  - Random coding bound $\max_{\rho\in[0,1]} E_0(\rho) - \rho R$

My favorite example: the BEC



- Simple capacity $1-\delta$ bits per channel use
- With perfect feedback, simple to achieve: retransmit until it gets through
  - Time till success: Geometric$(1-\delta)$
  - Expected time to get through: $\frac{1}{1-\delta}$



- 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.

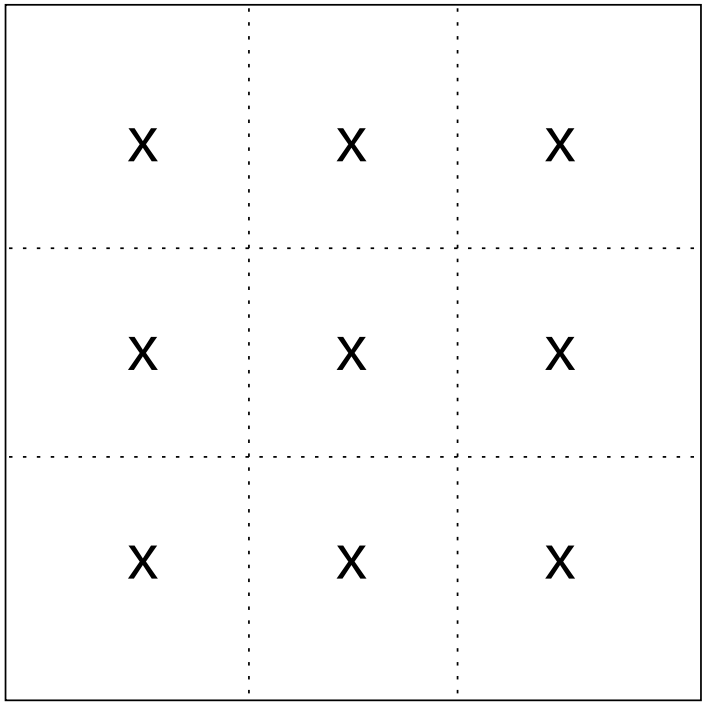
BEC with feedback and fixed *blocks*

- 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-62 showed that this type of behavior is common: $E^+(R) = E_{sp}(R)$ for symmetric channels.

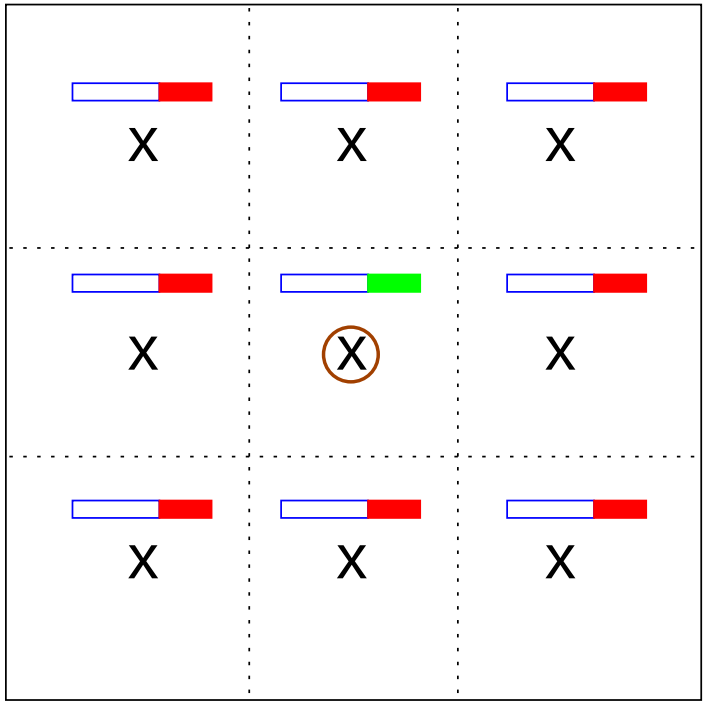
Review: Fixed blocks



- Hard decision regions cover space

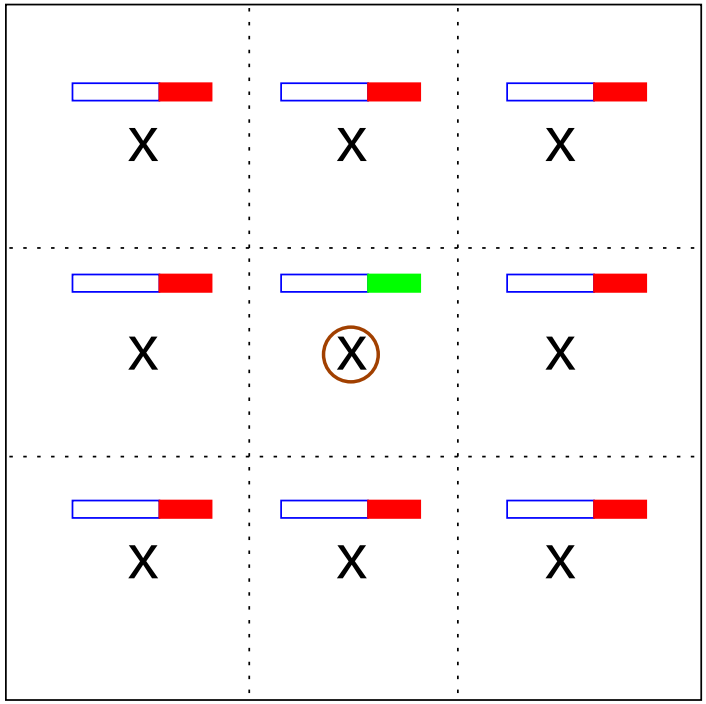
- Feedback is pointless

Review: Fixed blocks, Soft deadlines



- Hard decisions regions but check hash signatures

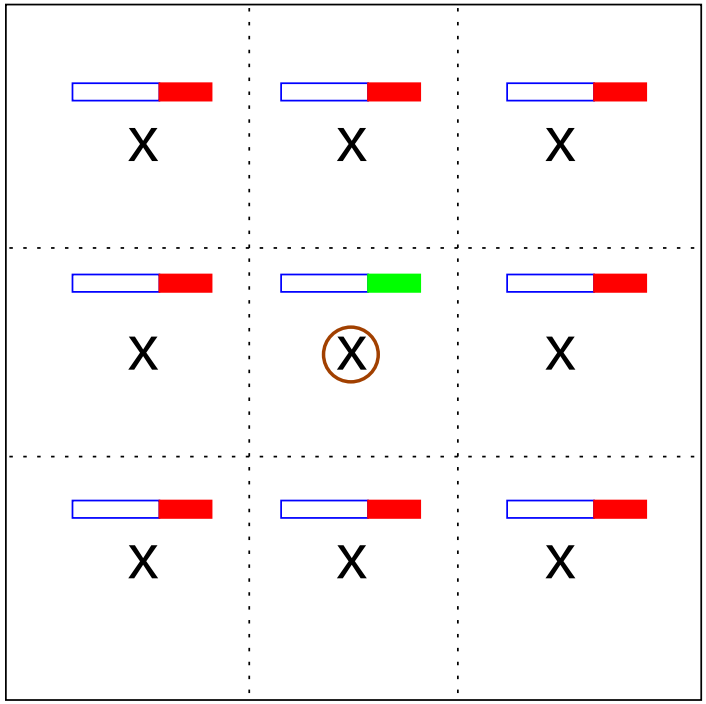
Review: Fixed blocks, Soft deadlines



- 1 bit feedback can request retransmissions
- Can interpret as expected block-length

- Hard decisions regions but check hash signatures

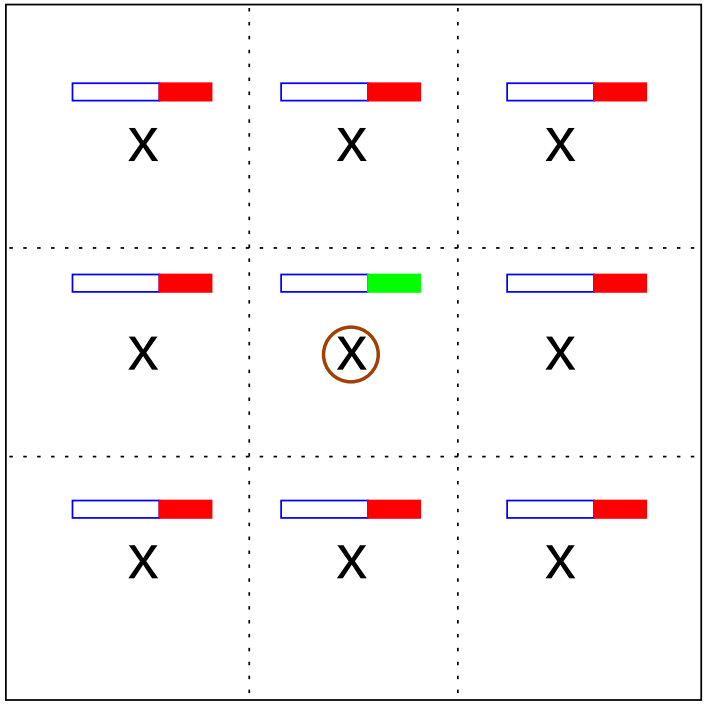
Review: Fixed blocks, Soft deadlines



- Hard decisions regions but check hash signatures

- 1 bit feedback can request retransmissions
- Can interpret as expected block-length
- Run close to capacity
- Use $\approx n(C - R)$ bits for signatures

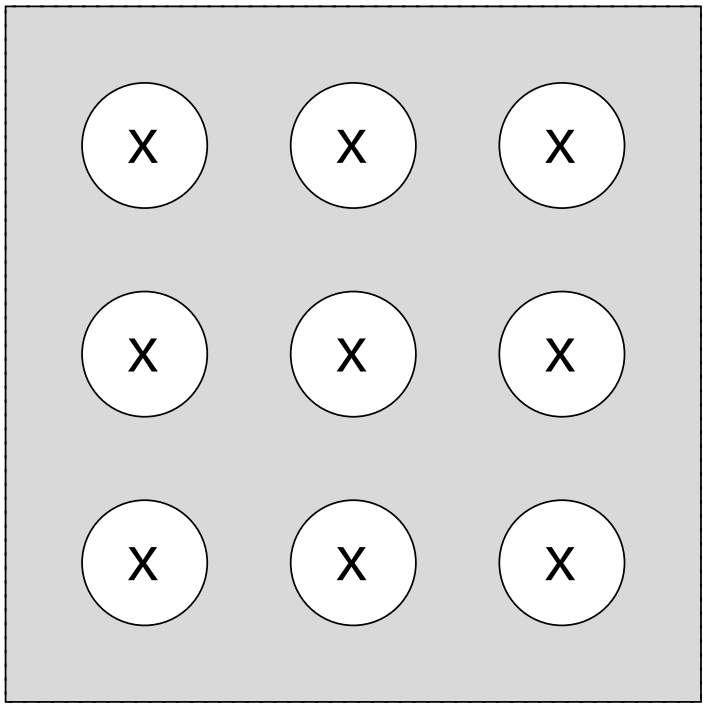
Review: Fixed blocks, Soft deadlines



- Hard decisions regions but check hash signatures

- 1 bit feedback can request retransmissions
- Can interpret as expected block-length
- Run close to capacity
- Use $\approx n(C - R)$ bits for signatures
- Linear slope -1 for error exponent

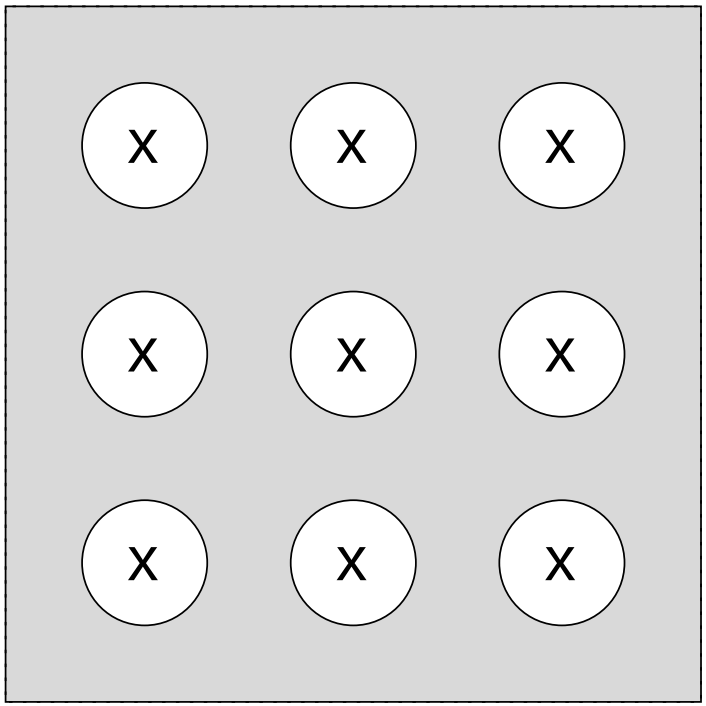
Review: Fixed blocks, Soft deadlines: Forney-68



- 1 bit feedback can request retransmissions
- Can interpret as expected block-length

- Refuse to decide when ambiguous

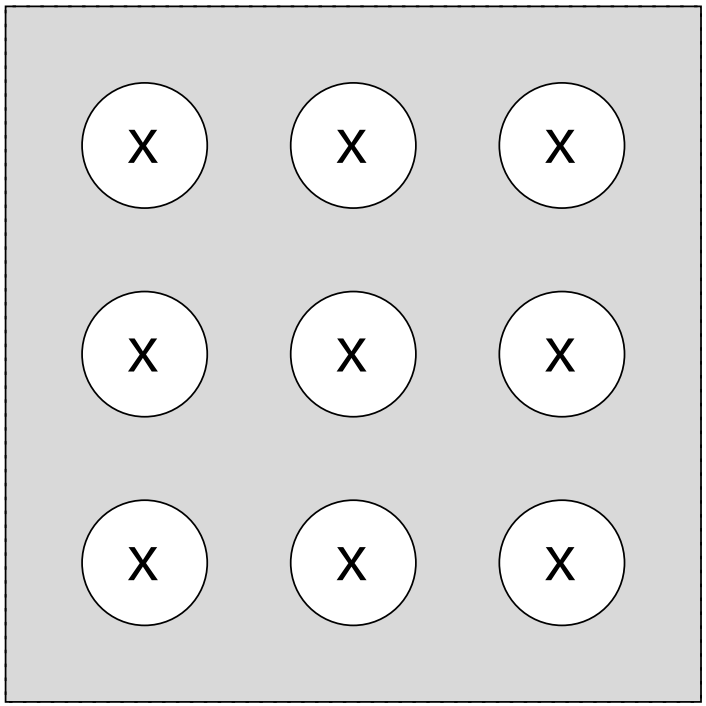
Review: Fixed blocks, Soft deadlines: Forney-68



- 1 bit feedback can request retransmissions
- Can interpret as expected block-length
- Decision regions catch the typical sets only

- Refuse to decide when ambiguous

Review: Fixed blocks, Soft deadlines: Forney-68



- Refuse to decide when ambiguous

- 1 bit feedback can request retransmissions
- Can interpret as expected block-length
- Decision regions catch the typical sets only
- Better error exponents at lower rates

Review: Fixed blocks, Soft deadlines: Burnashev-76

- Is more feedback helpful?
- Burnashev said yes:

Review: Fixed blocks, Soft deadlines: Burnashev-76

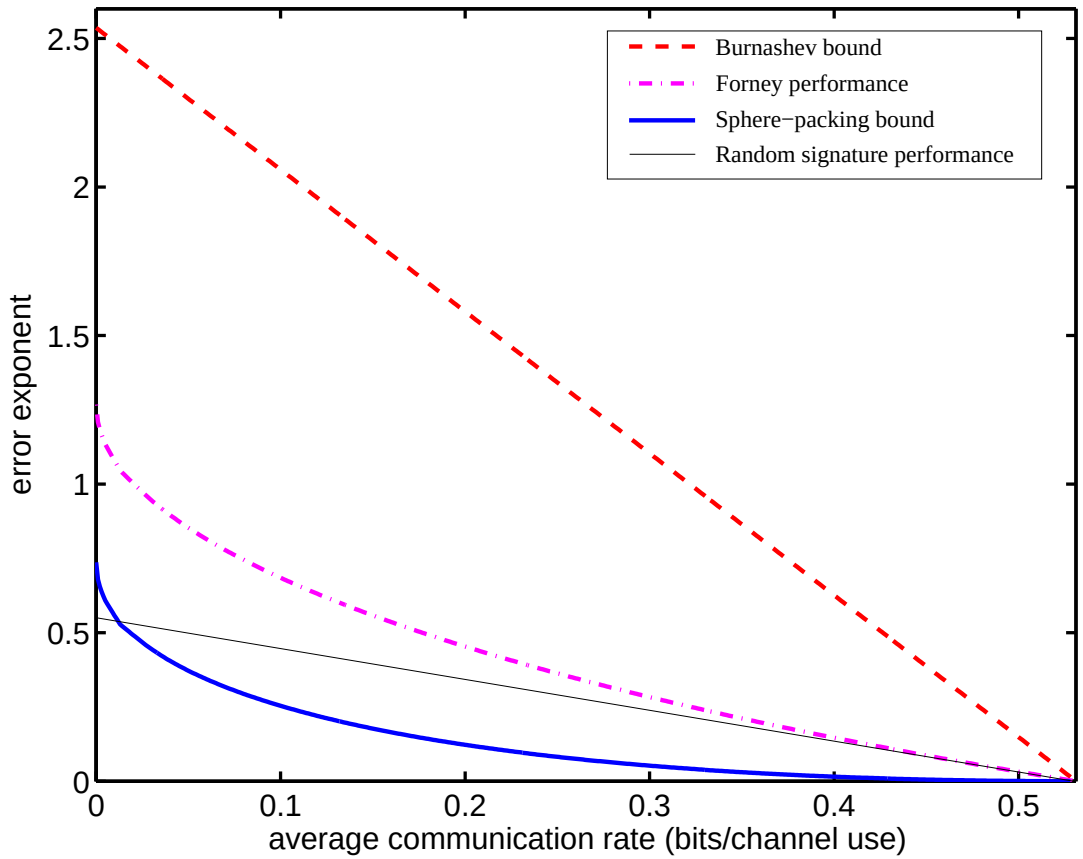
- Is more feedback helpful?
- Burnashev said yes:
  - Considered expected stopping time and used Martingale arguments.

Review: Fixed blocks, Soft deadlines: Burnashev-76

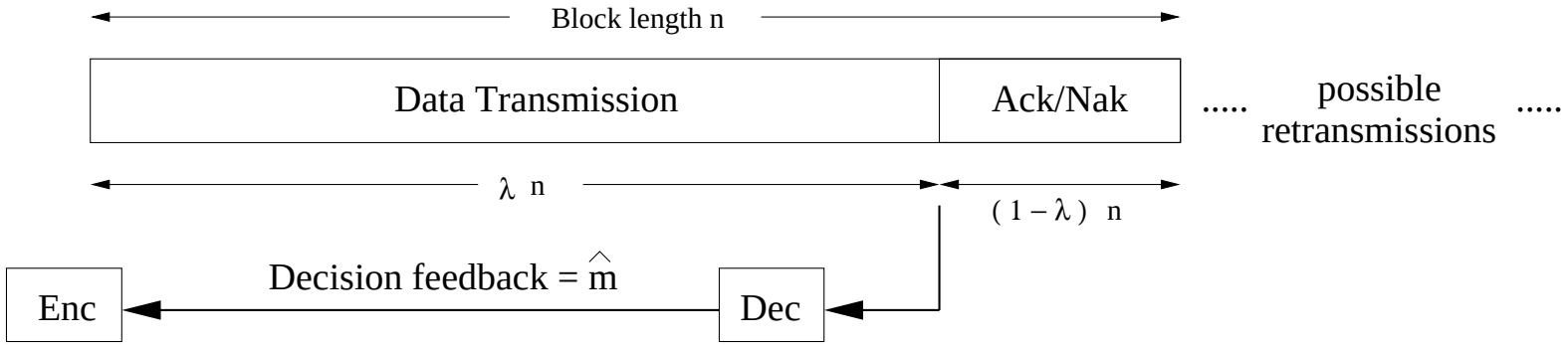
- Is more feedback helpful?
- Burnashev said yes:
  - Considered expected stopping time and used Martingale arguments.
  - Showed $C_1(1 - \frac{R}{C})$ was a bound where $C_1 = \max_{i,j} D(p_i||p_j)$

Review: Fixed blocks, Soft deadlines: Burnashev-76

- Is more feedback helpful?
- Burnashev said yes:
  - Considered expected stopping time and used Martingale arguments.
  - Showed $C_1(1 - \frac{R}{C})$ was a bound where $C_1 = \max_{i,j} D(p_i||p_j)$

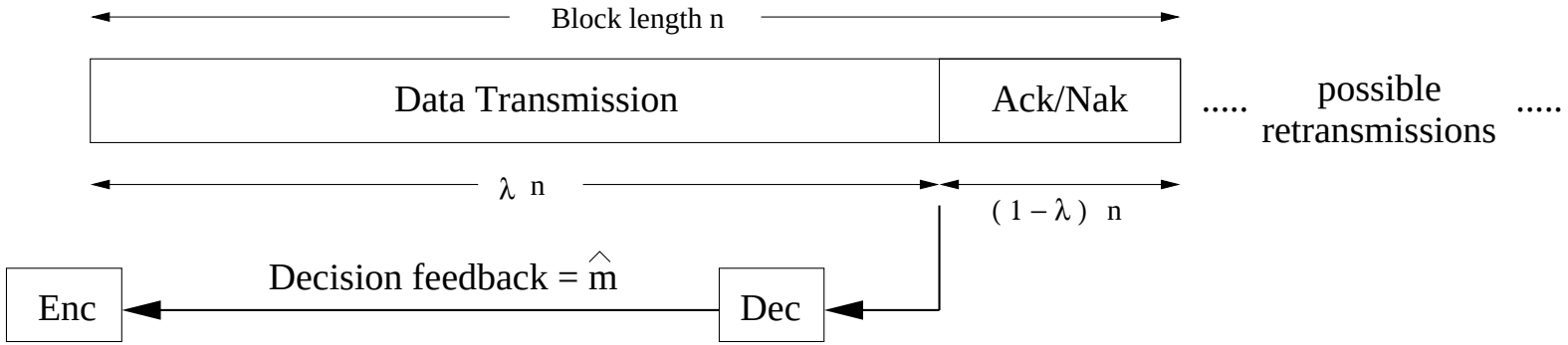


Review: Yamamoto-Itoh-79 strategy attains the Burnashev bound



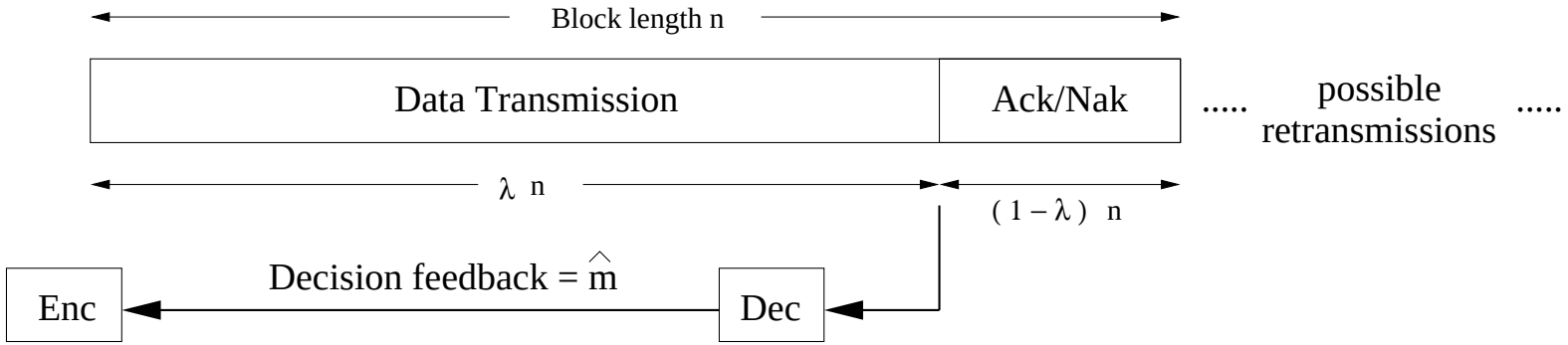
- **Data Transmission:** λn channel uses for block code at $R \simeq C$

Review: Yamamoto-Itoh-79 strategy attains the Burnashev bound



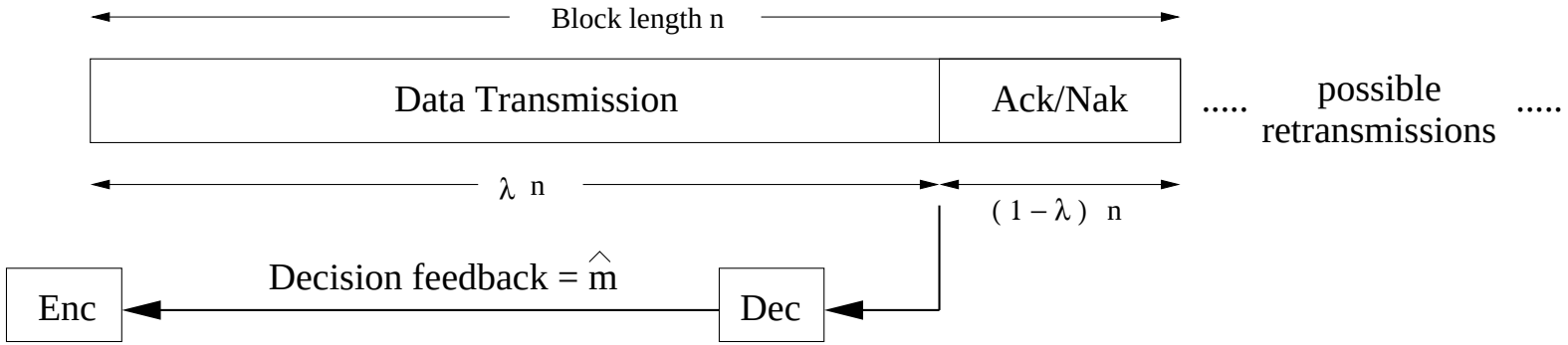
- **Data Transmission:** λn channel uses for block code at $R \simeq C$
- **Decision Feedback:** $\hat{m}$ sent back

Review: Yamamoto-Itoh-79 strategy attains the Burnashev bound



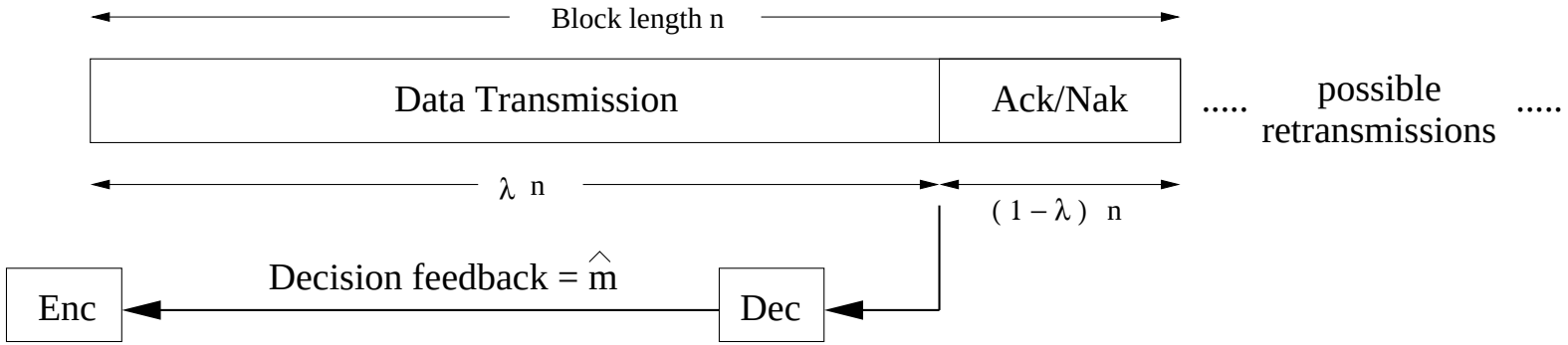
- **Data Transmission:** λn channel uses for block code at $R \simeq C$
- **Decision Feedback:** $\hat{m}$ sent back
- **Confirm/Deny:** $(1-\lambda)n$ channel uses to ACK or NAK

Review: Yamamoto-Itoh-79 strategy attains the Burnashev bound



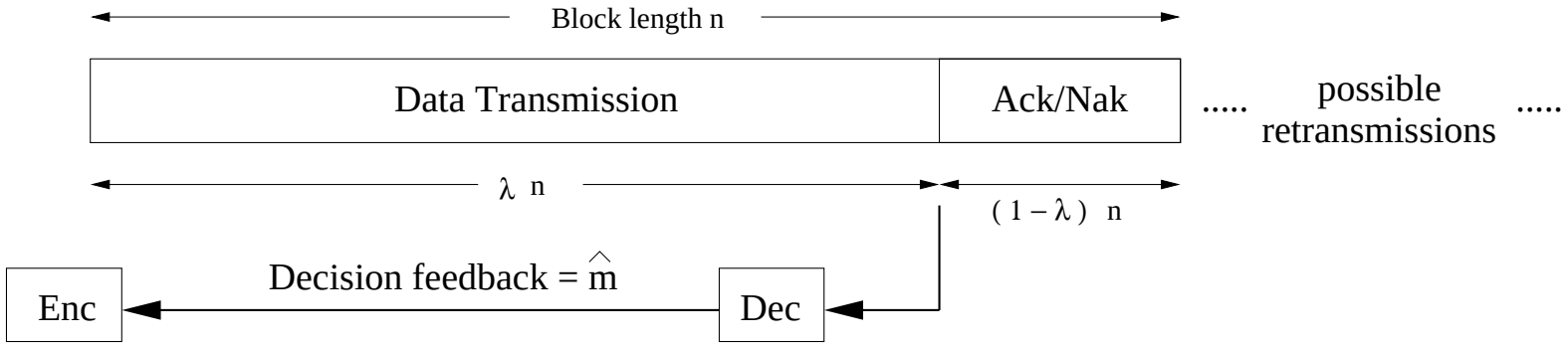
- **Data Transmission:** λn channel uses for block code at $R \simeq C$
- **Decision Feedback:** $\hat{m}$ sent back
- **Confirm/Deny:** $(1-\lambda)n$ channel uses to ACK or NAK
- If confirmed, decode to $\hat{m}$ otherwise erase.

Review: Yamamoto-Itoh-79 strategy attains the Burnashev bound



- **Data Transmission:** λn channel uses for block code at $R \simeq C$
- **Decision Feedback:** $\hat{m}$ sent back
- **Confirm/Deny:** $(1-\lambda)n$ channel uses to ACK or NAK
- If confirmed, decode to $\hat{m}$ otherwise erase.
- $\Pr[\text{err}] = \Pr[\text{NAK} \rightarrow \text{ACK}] = 2^{-(1-\lambda)nC_1} \simeq 2^{-nC_1(1-\frac{R}{C})}$

Review: Yamamoto-Itoh-79 strategy attains the Burnashev bound

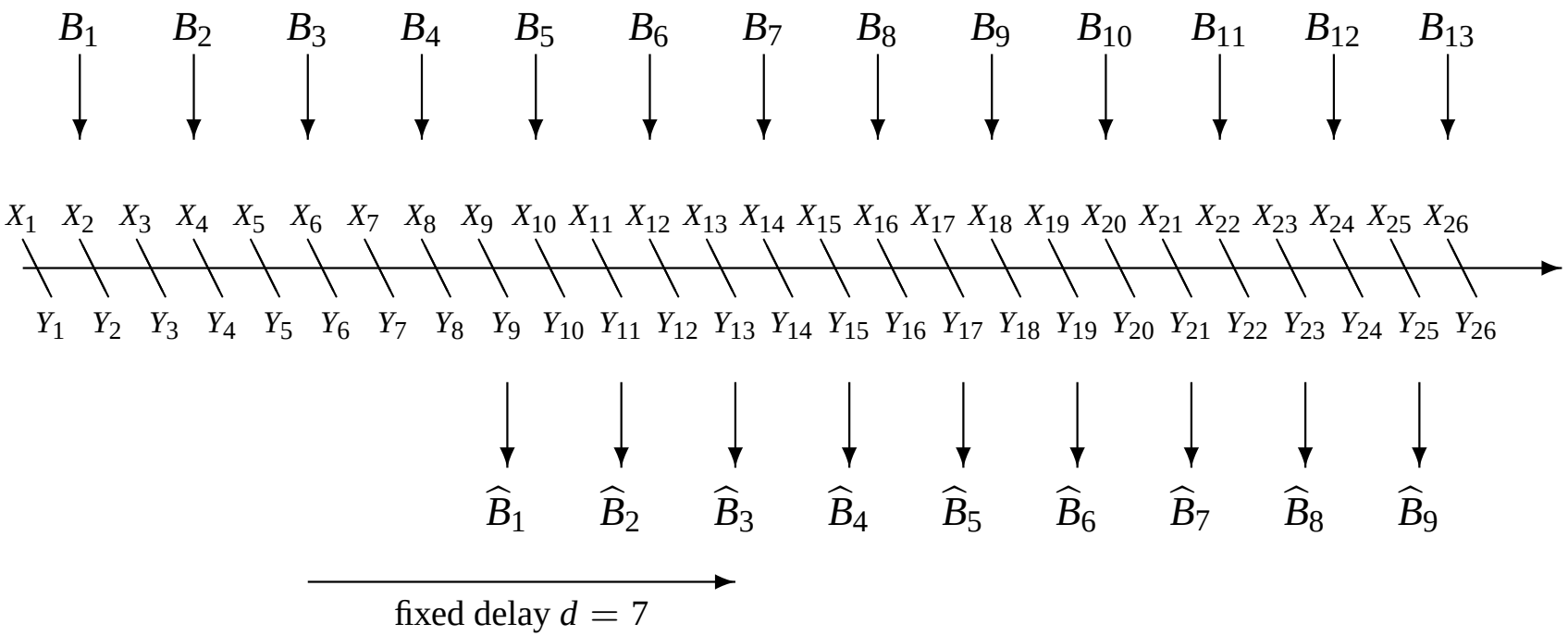


- **Data Transmission:** λn channel uses for block code at $R \simeq C$
- **Decision Feedback:** $\hat{m}$ sent back
- **Confirm/Deny:** $(1-\lambda)n$ channel uses to ACK or NAK
- If confirmed, decode to $\hat{m}$ otherwise erase.
- $\Pr[\text{err}] = \Pr[\text{NAK} \rightarrow \text{ACK}] = 2^{-(1-\lambda)nC_1} \simeq 2^{-nC_1(1-\frac{R}{C})}$
- Moral: **Collective** reward/punishment is good for reliability

Outline

1. Motivation and background
2. Feedback, delay, and error probability: memoryless channels
   - Block-oriented results
   - Idealized case: fixed-delay with hard deadlines
     - ★ The BEC example
     - ★ Without feedback
     - ★ The focusing bound
     - ★ The source-coding analogy
     - ★ Approaching the focusing bound (with help)
     - ★ No help and universality
   - Idealized case: Fixed-delay with soft deadlines
   - Unreliable/Noisy feedback
3. Feedback and memory
4. Conclusions and open problems

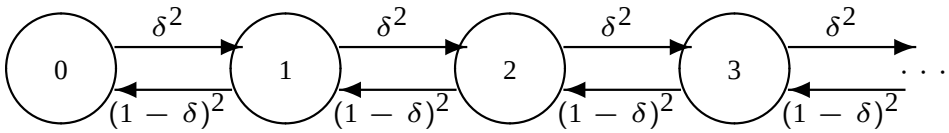
What do we mean by fixed delay?



- "Hard" deadlines: must commit to each bit
- "Soft" deadlines: allow saying "I don't know"

BEC with feedback and fixed *delay*

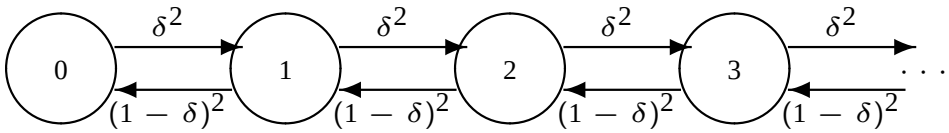
- $R = \frac{1}{2}$ example:



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

BEC with feedback and fixed *delay*

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

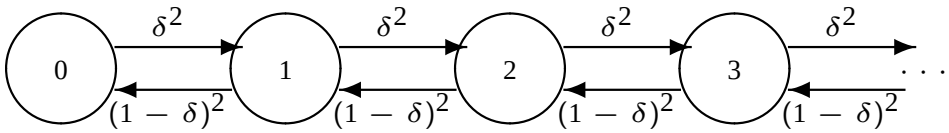


- Birth-death chain: positive recurrent if $\delta < \frac{1}{2}$
- Delay exponent easy to see:

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

BEC with feedback and fixed *delay*

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



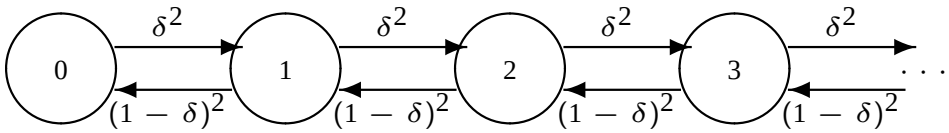
- Birth-death chain: positive recurrent if $\delta < \frac{1}{2}$
- Delay exponent easy to see:

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

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

BEC with feedback and fixed *delay*

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



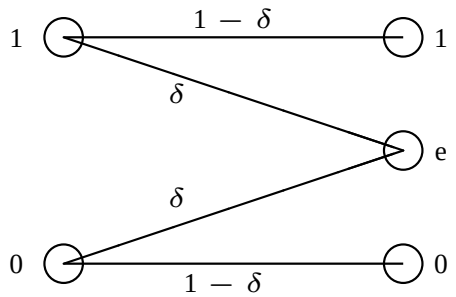
- Birth-death chain: positive recurrent if $\delta < \frac{1}{2}$
- Delay exponent easy to see:

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

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

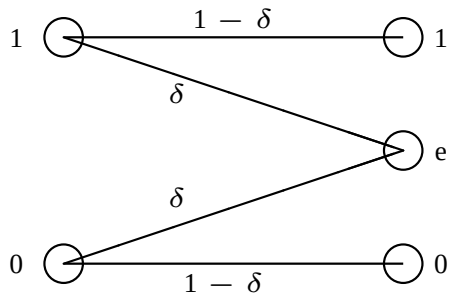
Block-coding is misleading!

The BEC: understanding why?



- Without feedback: send random parities and solve the equations
- With perfect feedback, simple to achieve: retransmit until success

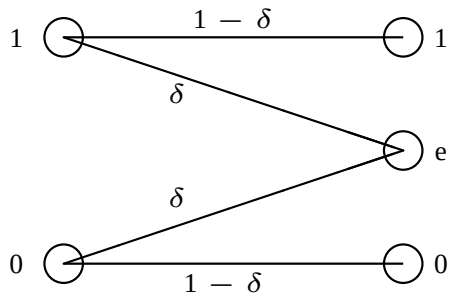
The BEC: understanding why?



- Without feedback: send random parities and solve the equations

- With perfect feedback, simple to achieve: retransmit until success
- Queuing perspective: deterministic arrivals with independent geometric service times.

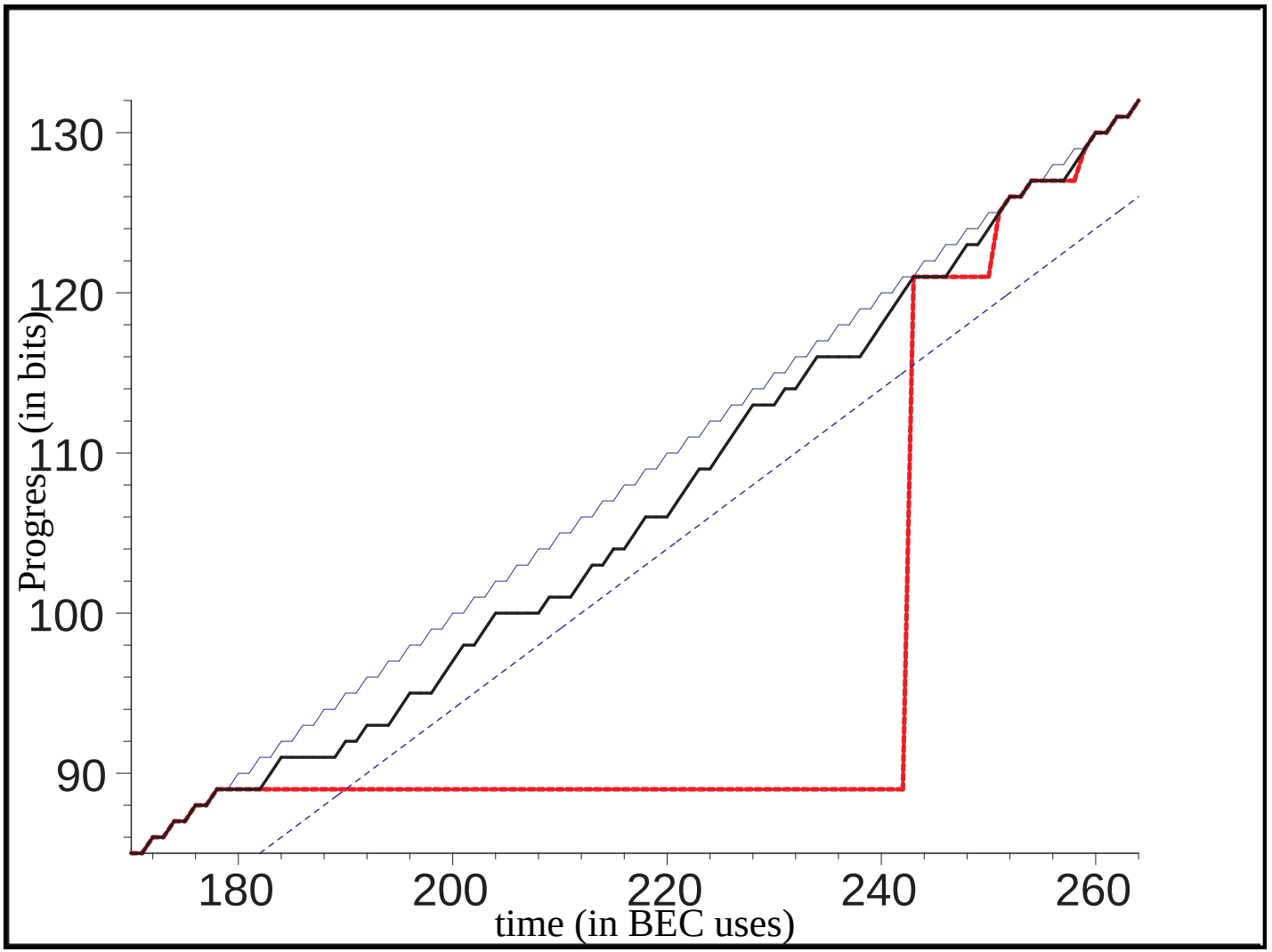
The BEC: understanding why?



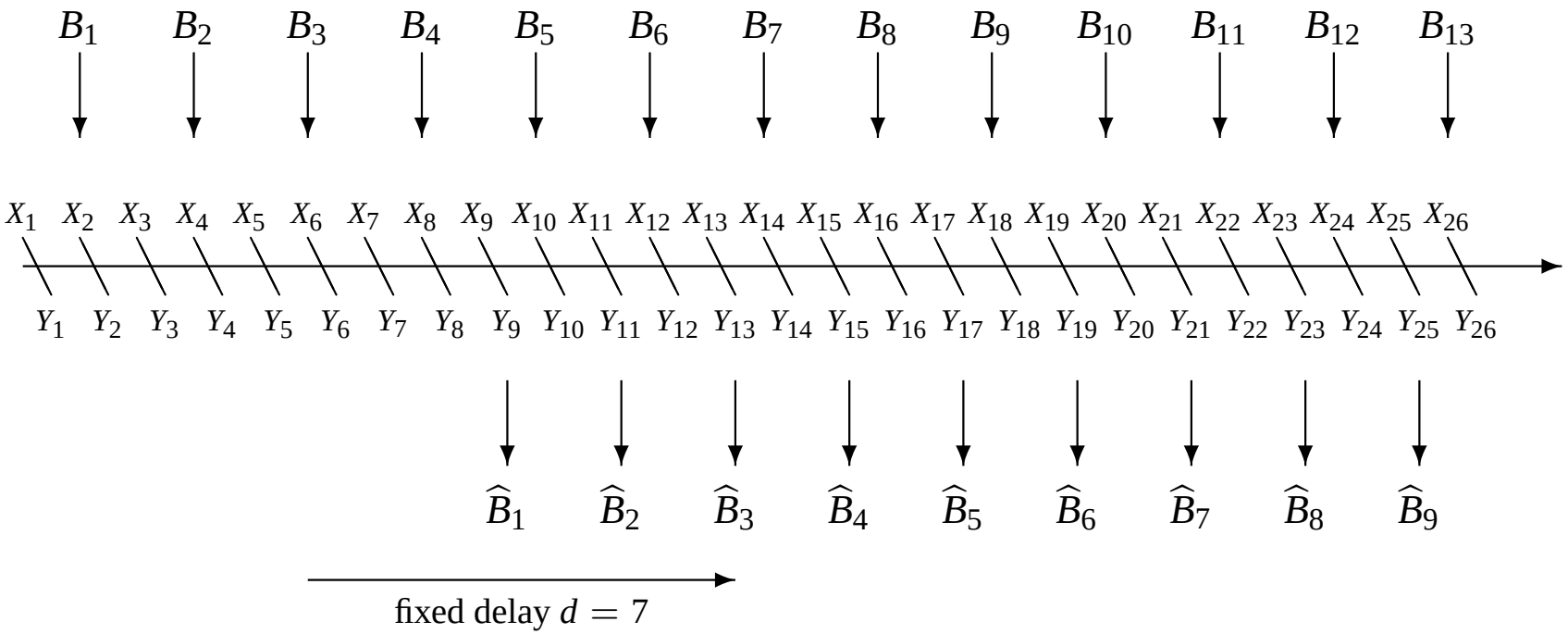
- Without feedback: send random parities and solve the equations

- With perfect feedback, simple to achieve: retransmit until success
- Queuing perspective: deterministic arrivals with independent geometric service times.
- Would behave at least as well if the service times were independent and dominated by a geometric.

So where is this boost coming from?

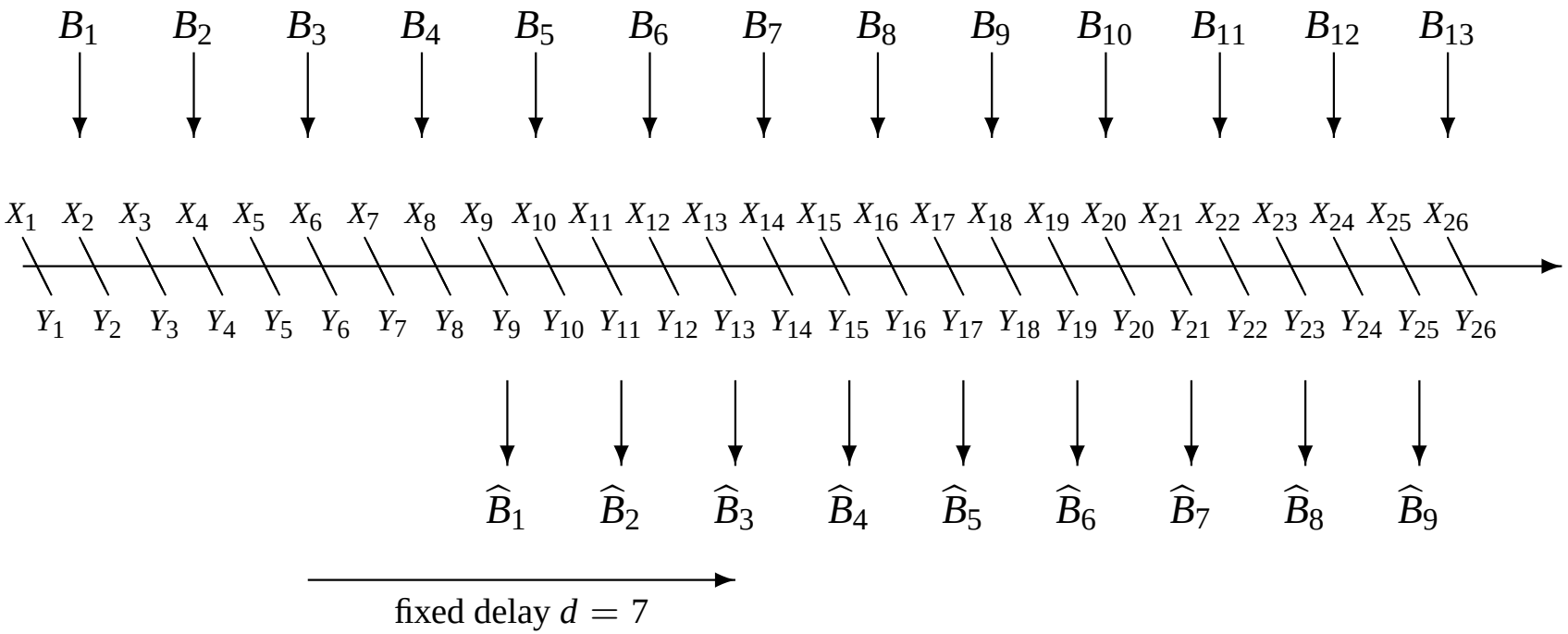


Is it feedback, delay, or the combination?



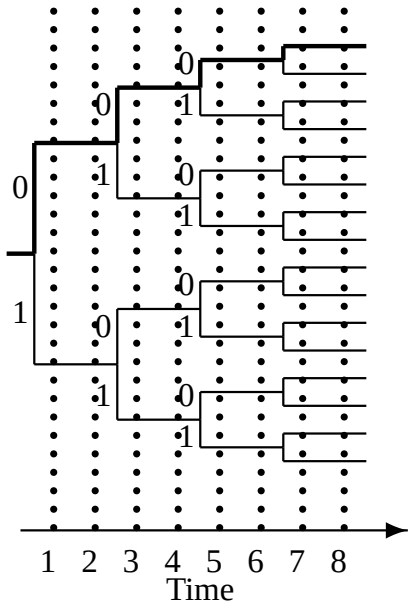
- Can generally achieve $E_r(R)$ with delay using convolutional codes.

Is it feedback, delay, or the combination?



- Can generally 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*.

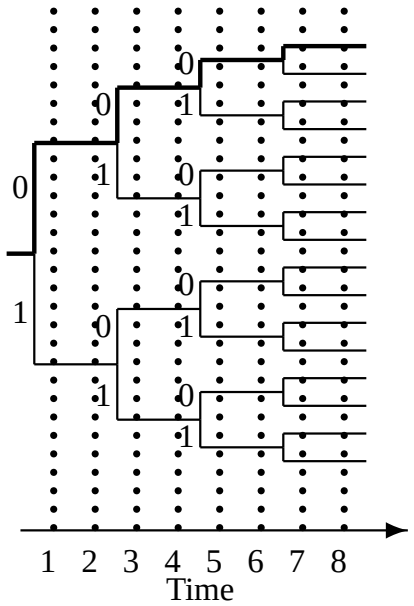
Nonblock codes without feedback



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

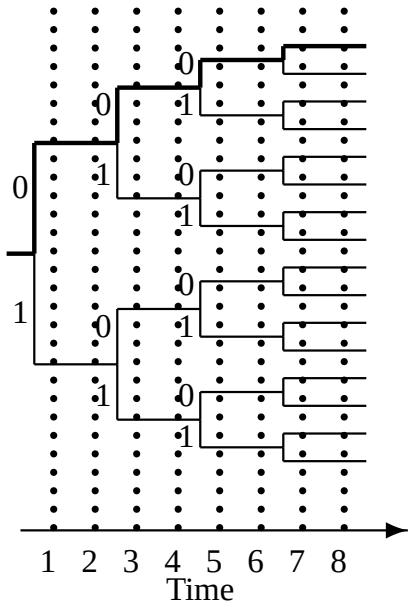


Infinite binary tree, with iid random labels:

- 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

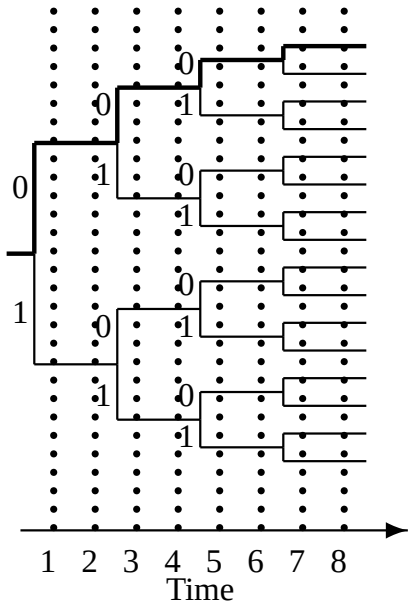


Infinite binary tree, with iid random labels:

- 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** $R < C$

Convolutional codes, feedback, and computation

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

Convolutional codes, feedback, and computation

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

Convolutional codes, feedback, and computation

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

Convolutional codes, feedback, and computation

- At $R < E_0(1)$, sequential decoding expands only a finite number of nodes on average.
- But 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

Convolutional codes, feedback, and computation

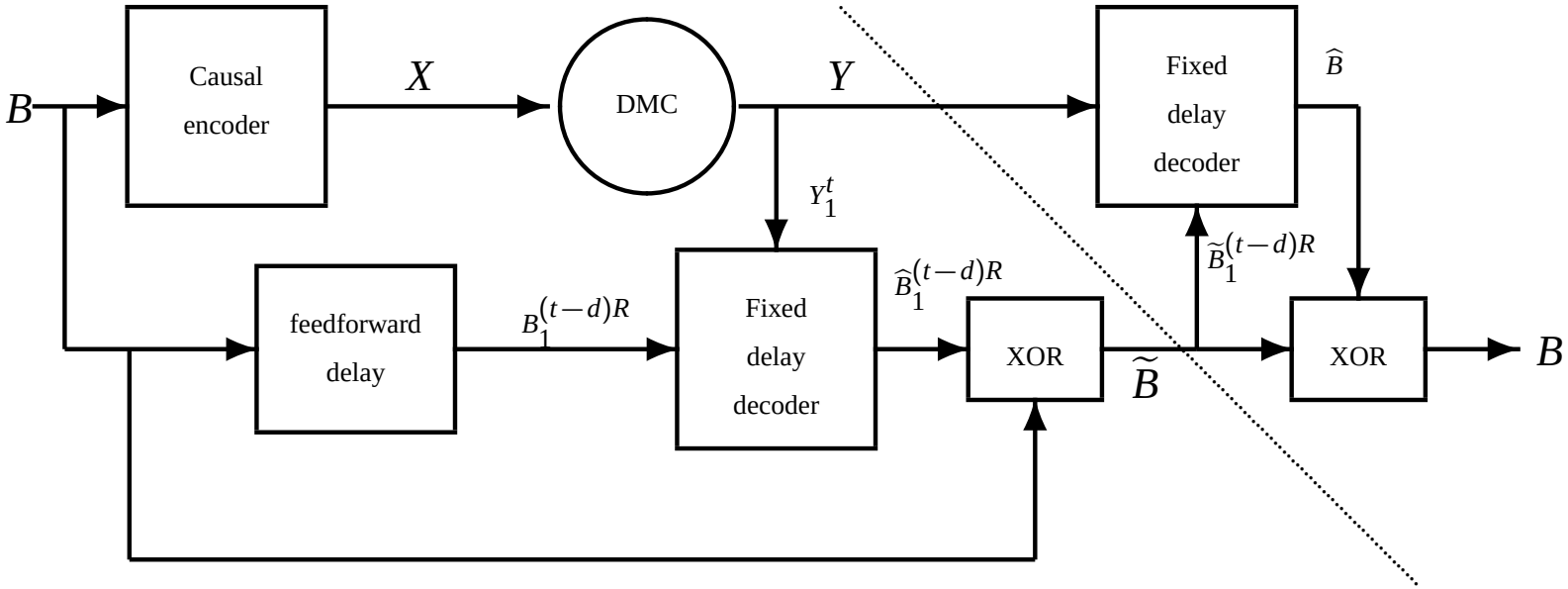
- At $R < E_0(1)$, sequential decoding expands only a finite number of nodes on average.
- But each expansion costs the (growing) constraint length.
- Idea: run a copy of the decoder at the encoder
  - Convolve against: $(B_1 + \widehat{B}_1(n)), (B_2 + \widehat{B}_2(n)), \ldots, (B_{n-1} + \widehat{B}_{n-1}(n)),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

Convolutional codes, feedback, and computation

- At $R < E_0(1)$, sequential decoding expands only a finite number of nodes on average.
- But 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!
- *Same probability of error with delay: achieves* $E_r(R)$

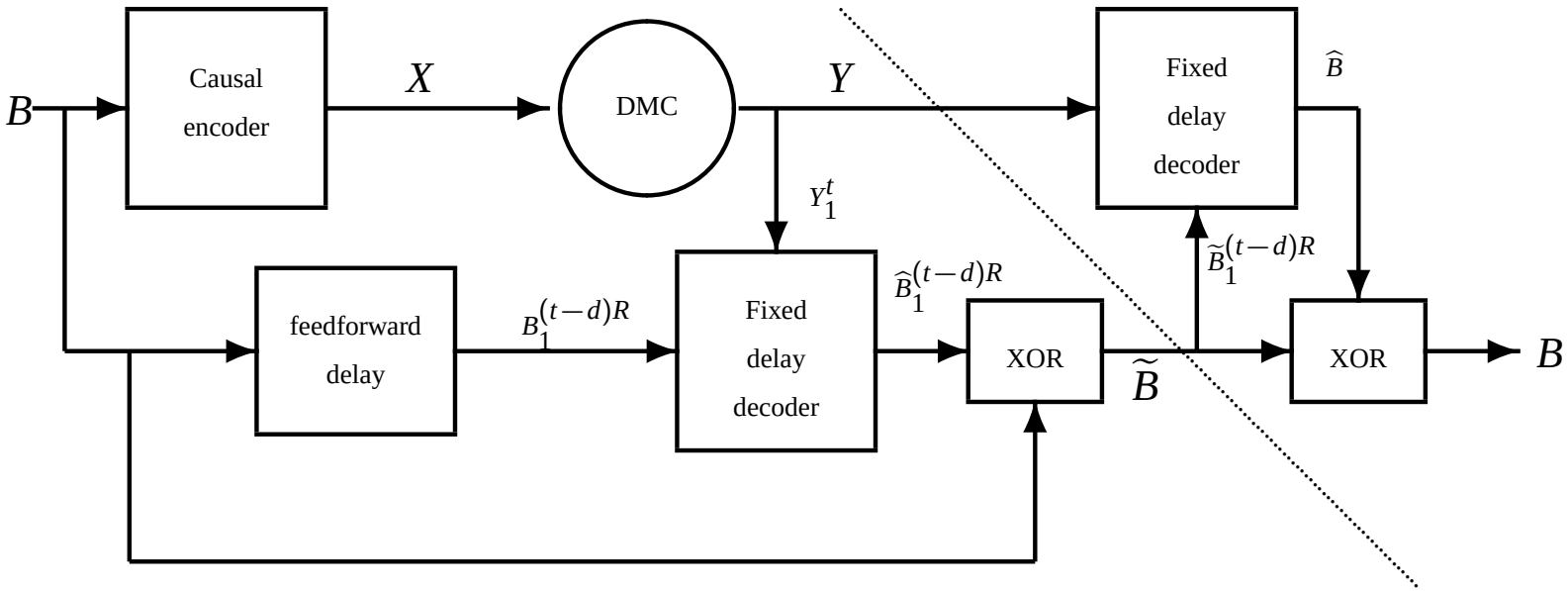
Pinsker's bounding construction explained

- Without feedback: $E^+(R)$ continues to be a bound.
- Consider a code with target delay d
  - Use it to construct a block-code with blocksize $n \gg d$
  - Genie-aided decoder: has the truth of all bits before i



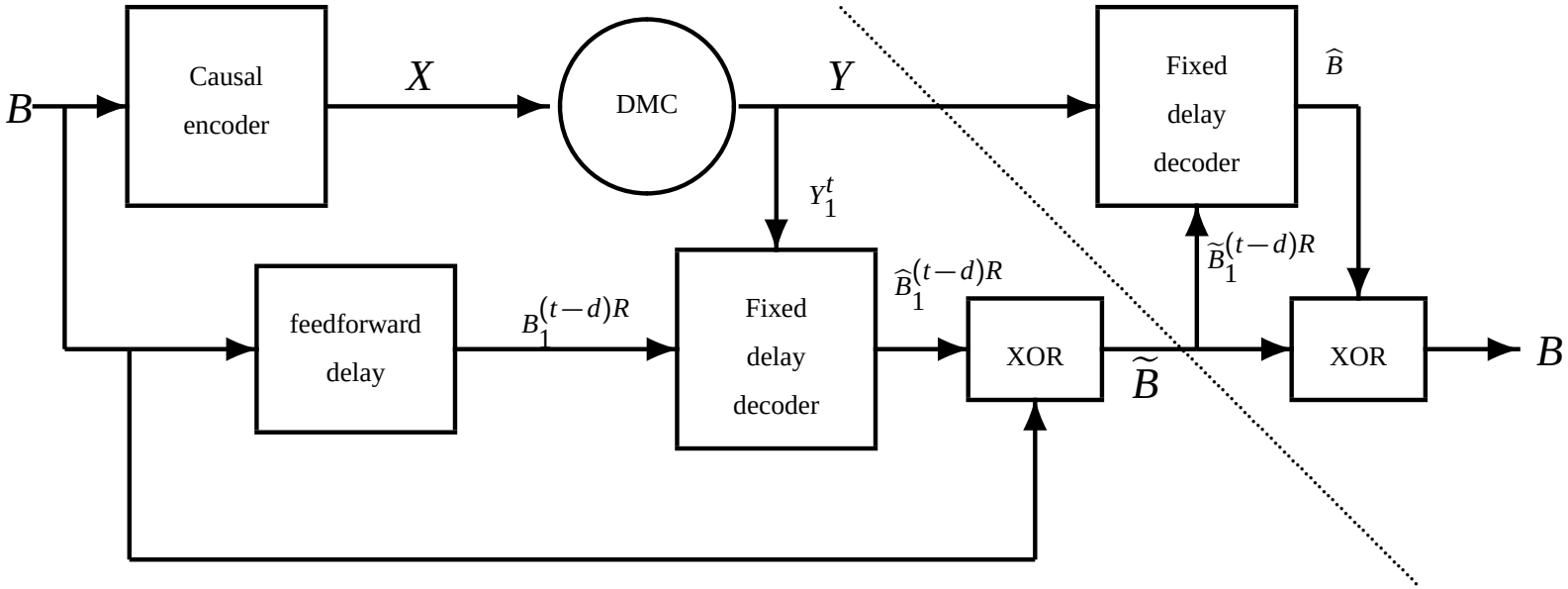
Pinsker's bounding construction explained

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

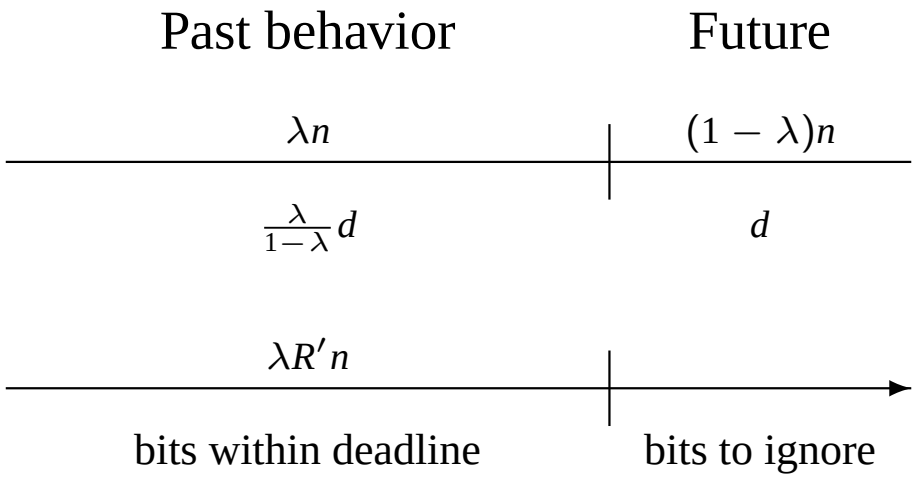


Pinsker's bounding construction explained

- Without feedback: $E^+(R)$ continues to be a bound.
- Consider a code with target delay d
  - Use it to construct a block-code with blocksize $n \gg d$
  - 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

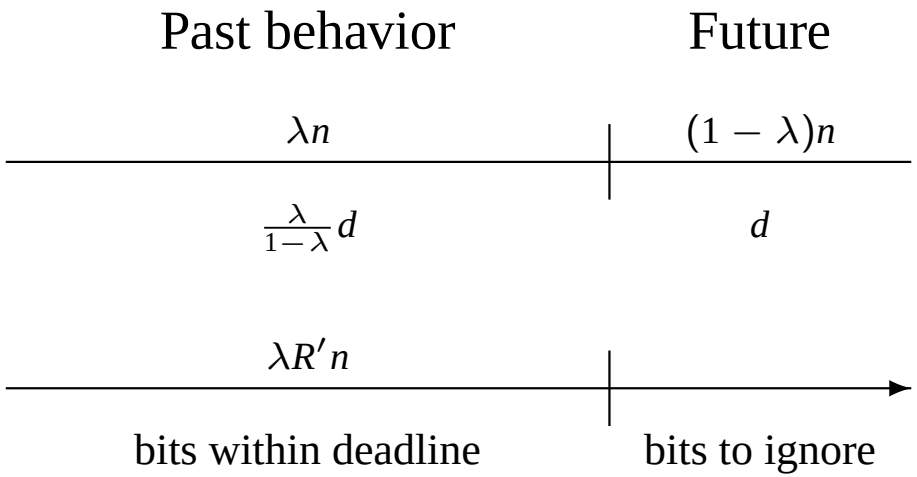


Using E^+ to bound α^* in general



- The block error probability is like $e^{-\alpha(1-\lambda)n}$ which cannot exceed the Haroutunian bound $e^{-E^+(\lambda R)n}$

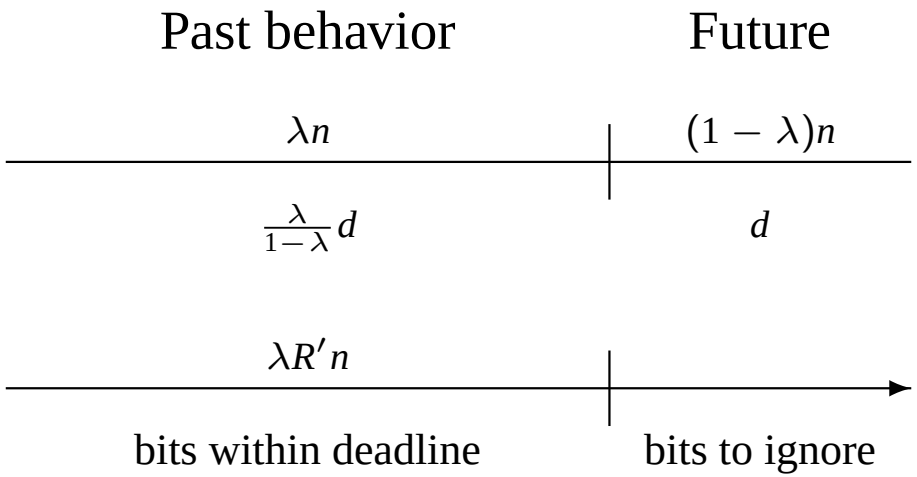
Using E^+ to bound α^* in general



- The block error probability is like $e^{-\alpha(1-\lambda)n}$ which cannot exceed the Haroutunian bound $e^{-E^+(\lambda R)n}$

$$\alpha^*(R) \leq \frac{E^+(\lambda R)}{1-\lambda}$$

Using E^+ to bound α^* in general



- The block error probability is like $e^{-\alpha(1-\lambda)n}$ which cannot exceed the Haroutunian bound $e^{-E^+(\lambda R)n}$

$$\alpha^*(R) \leq \frac{E^+(\lambda R)}{1-\lambda}$$

- 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 $\rho > 0$:

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

Using the Gallager function:

$$E_0(\rho) = -\max_q \ln \sum_j \left(\sum_i q_i p_{ij}^{\frac{1}{1+\rho}} \right)^{1+\rho}$$

Uncertainty focusing bound for symmetric DMCs

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

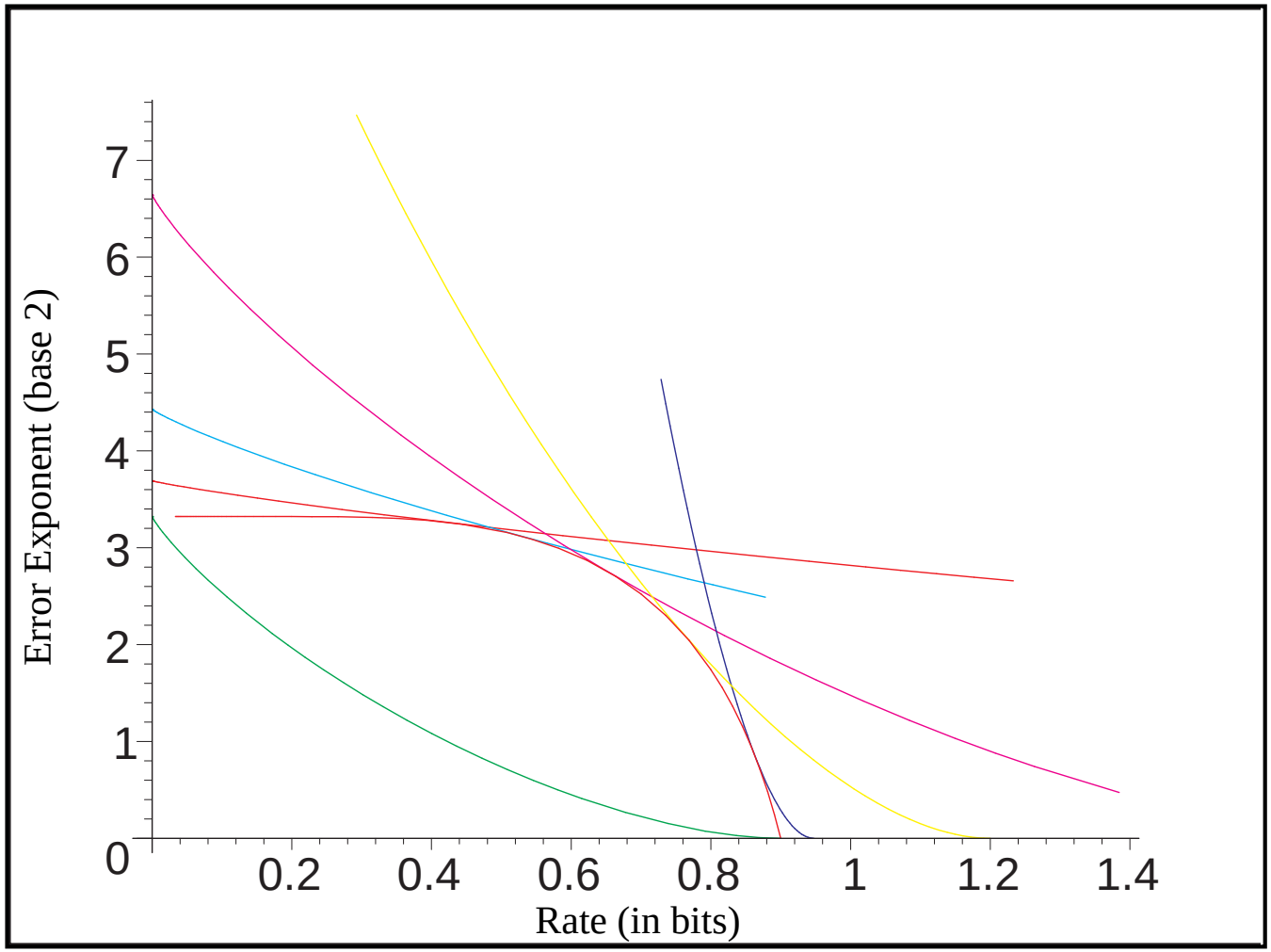
$$\begin{aligned} R(\rho) &= \frac{E_0(\rho)}{\rho} \\ E_a^+(\rho) &= E_0(\rho) \end{aligned}$$

Using the Gallager function:

$$E_0(\rho) = -\max_q \ln \sum_j \left(\sum_i q_i p_{ij}^{\frac{1}{1+\rho}} \right)^{1+\rho}$$

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

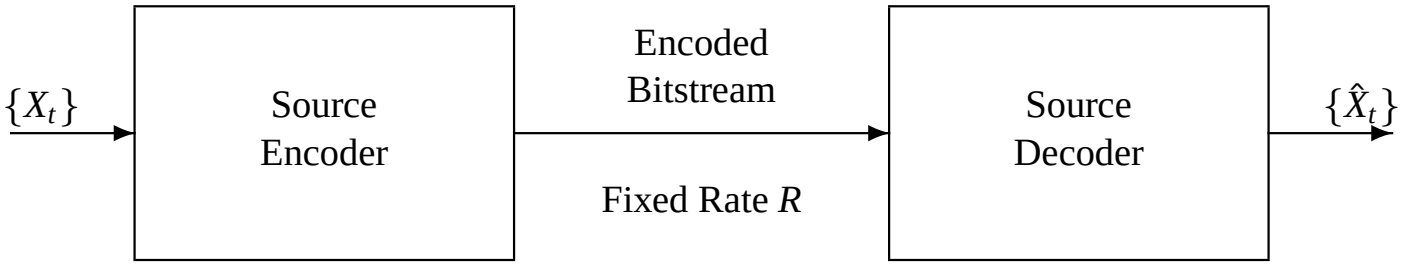
Upper bound tight for the BEC with feedback



Outline

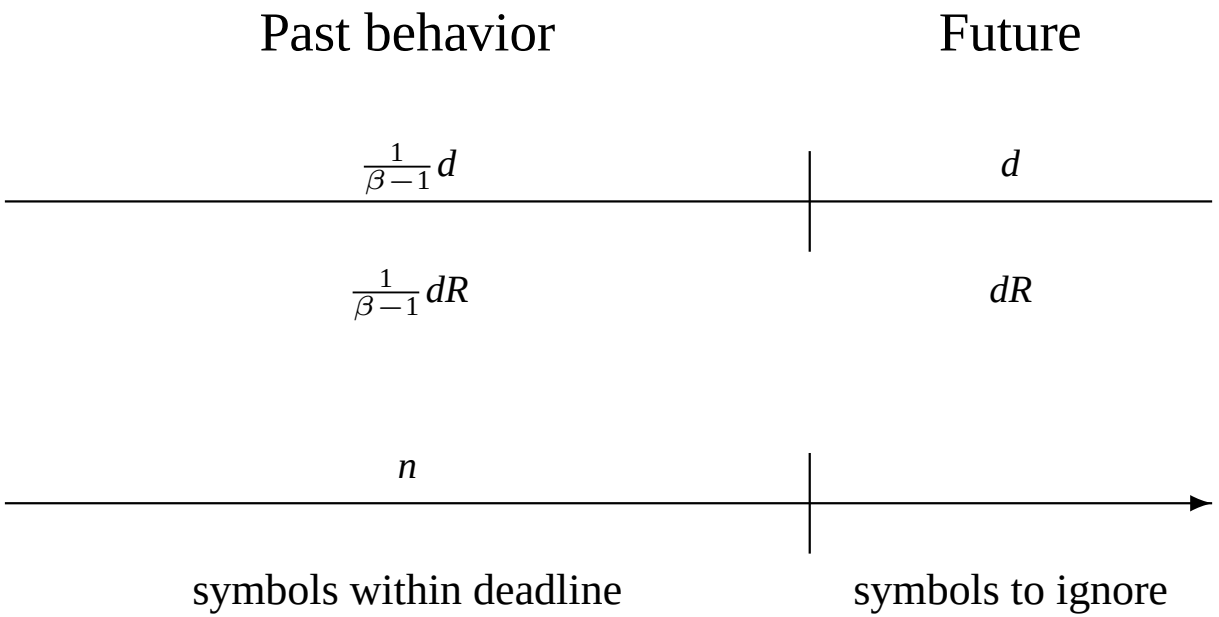
1. Motivation and background
2. Feedback, delay, and error probability: memoryless channels
   - Block-oriented results
   - Idealized case: fixed-delay with hard deadlines
     - ★ The BEC example
     - ★ Without feedback
     - ★ The focusing bound
     - ★ The source-coding analogy
     - ★ Approaching the focusing bound (with help)
     - ★ No help and universality
   - Idealized case: Fixed-delay with soft deadlines
   - Unreliable/Noisy feedback
3. Feedback and memory
4. Conclusions and open problems

The source coding problem



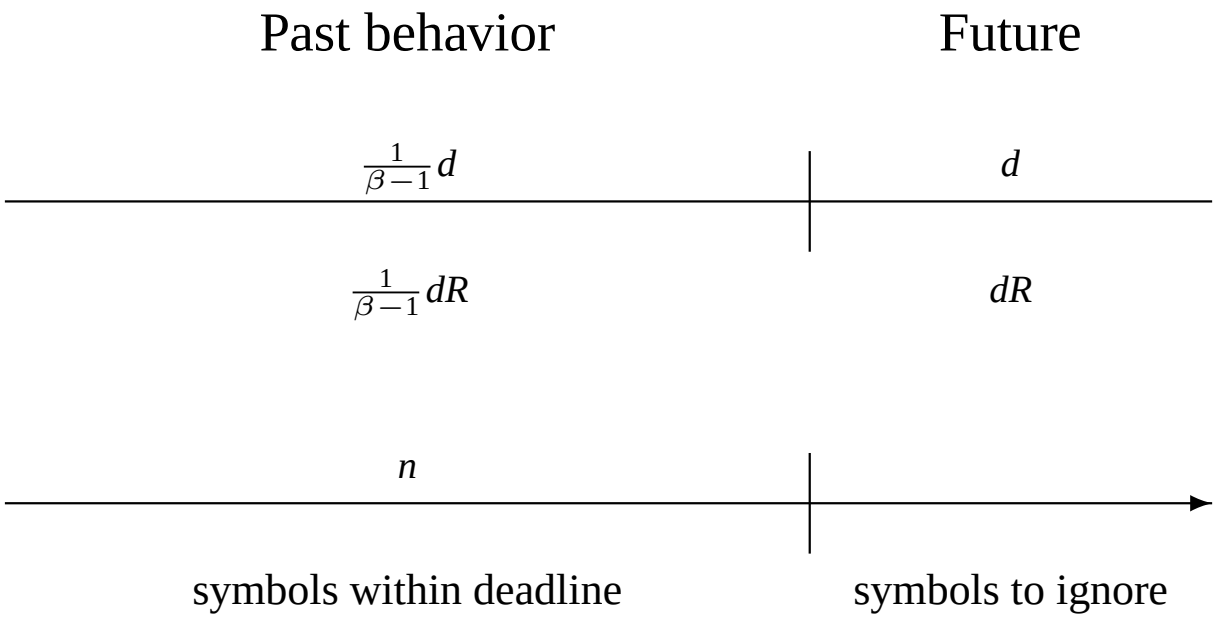
- *Assume* $\{X_t\}$ *iid*
- Application-level interface
  - Symbol error probability: $P_e = P(X_t \neq \hat{X}_t)$
  - End-to-end latency: d (measured in source timescale)
- Channel-code interface: fixed rate R (assumed noiseless)

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 $\exp(-nE_b(\beta R))$

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

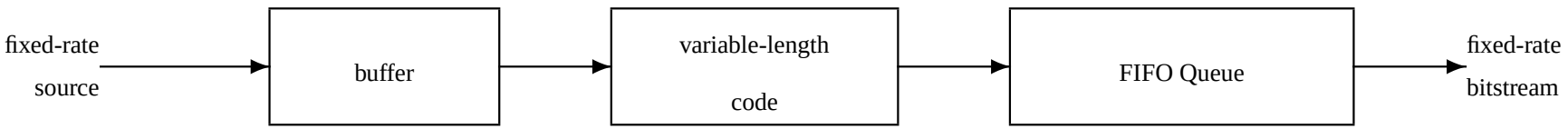
Only the past matters!

Achieving the focusing bound

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

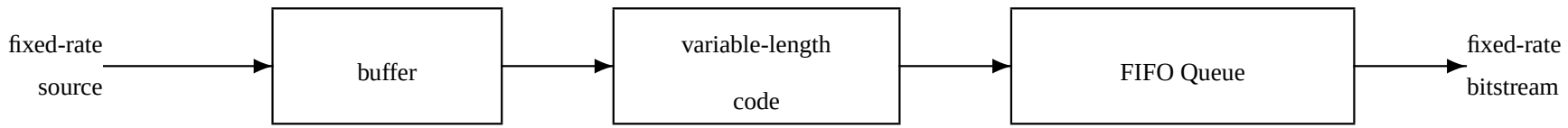
Achieving the focusing bound

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



Achieving the focusing bound

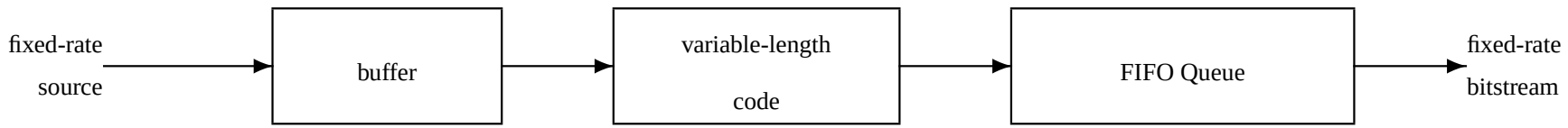
$$
\begin{aligned}
R(\rho) &= \frac{E_0(\rho)}{\rho} \\
E_s(\rho) &= E_0(\rho)
\end{aligned}
$$



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

Achieving the focusing bound

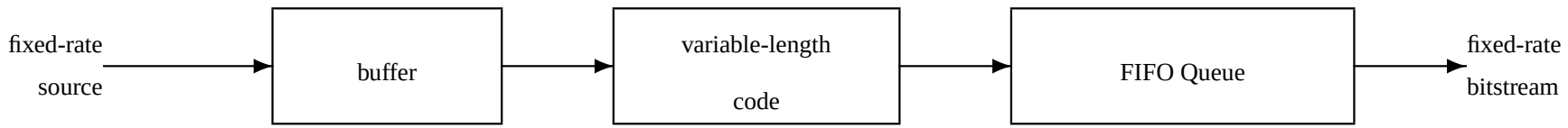
$$\begin{aligned} R(\rho) &= \frac{E_0(\rho)}{\rho} \\ E_s(\rho) &= E_0(\rho) \end{aligned}$$



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

Achieving the focusing bound

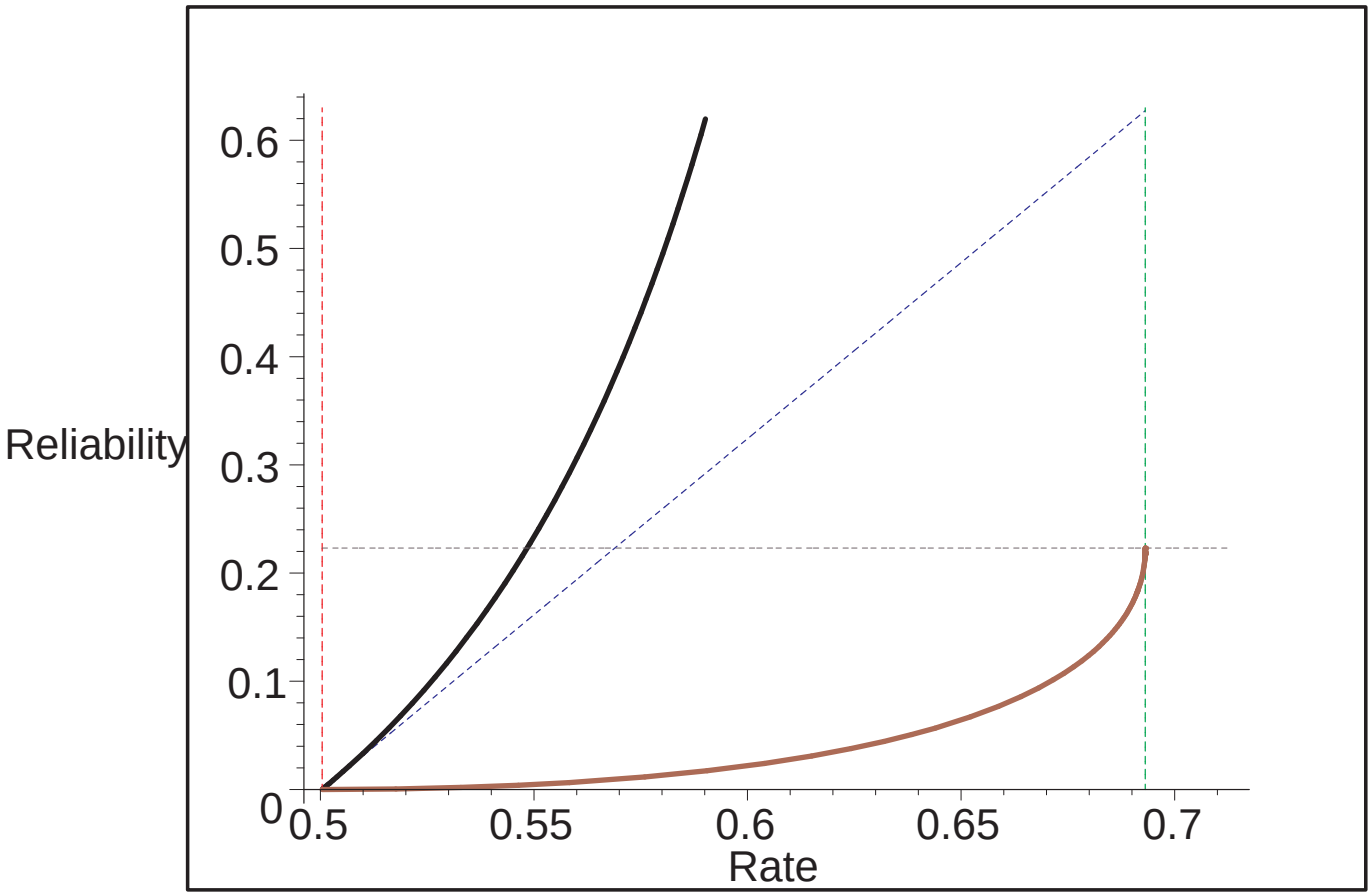
$$R(\rho) = \frac{E_0(\rho)}{\rho}$$
$$E_s(\rho) = E_0(\rho)$$



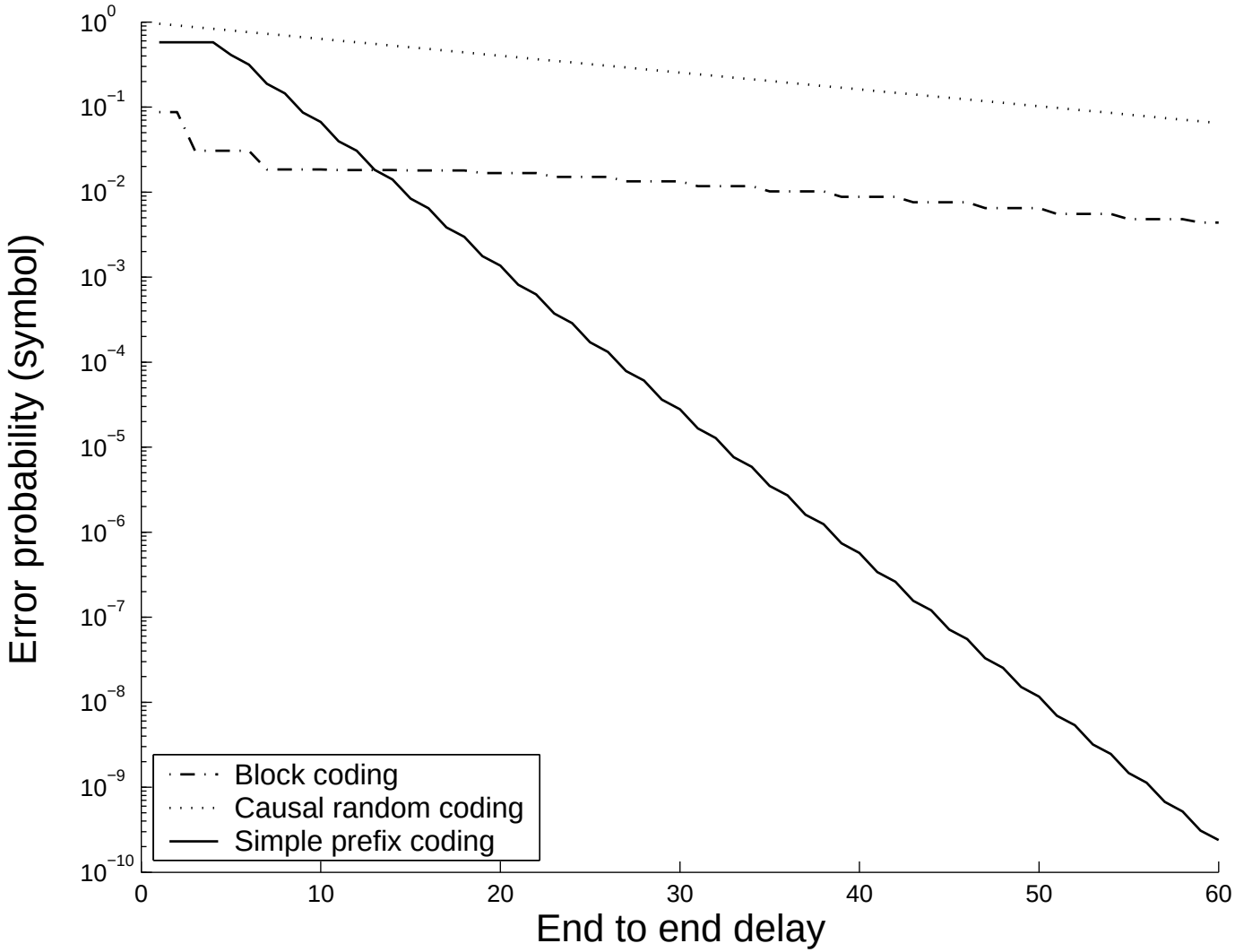
- Pick miniblock n large enough but small relative to d.
- Variable-length codes turn into variable delay at the receiver.
- Queuing delay dominates.
- R translates queue length into delay.

A simple example: related to Jelinek-68

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

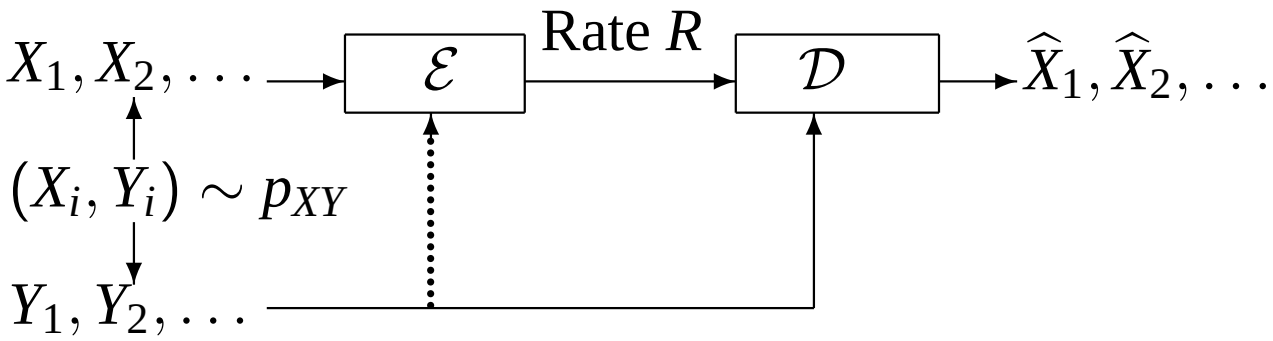


How far till Asymptopia?



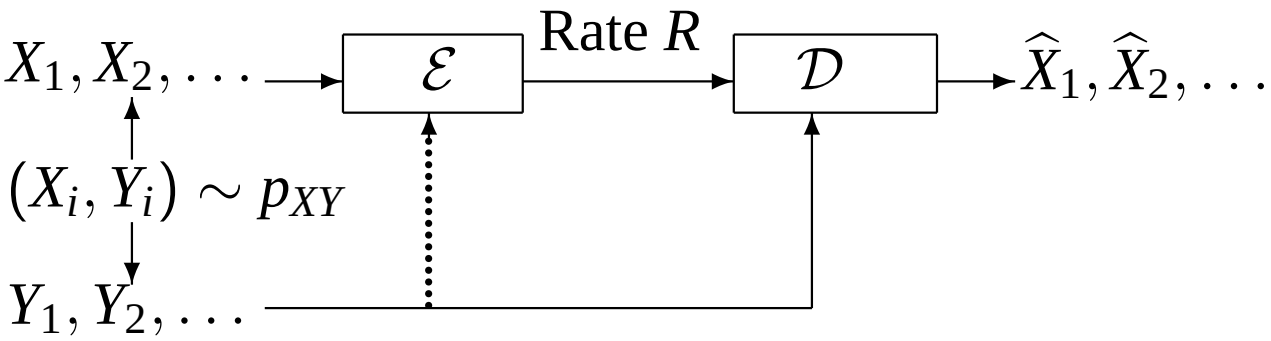
Simple example with a ternary source and a specific code ($n = 2$).

Side-information at the decoder

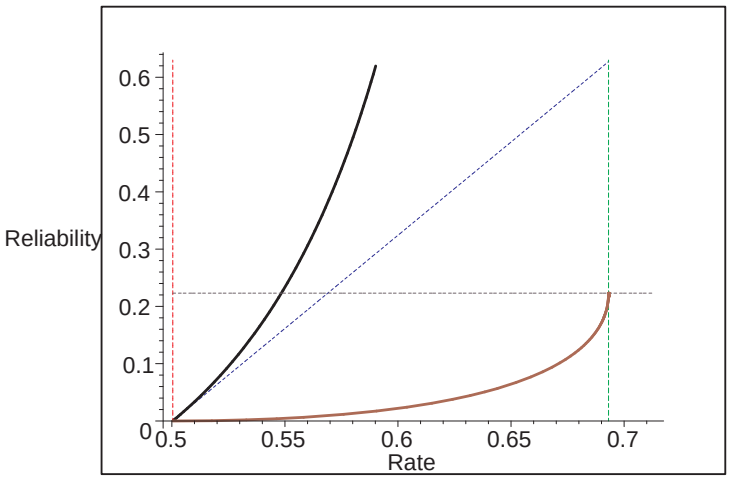


- Encoder may be ignorant of the side-information. (Slepian-Wolf)
- If $P_{X,Y}$ symmetric with uniform marginals, can do no better than $E_b(R)$ with delay. (analogous to Pinsker's argument)

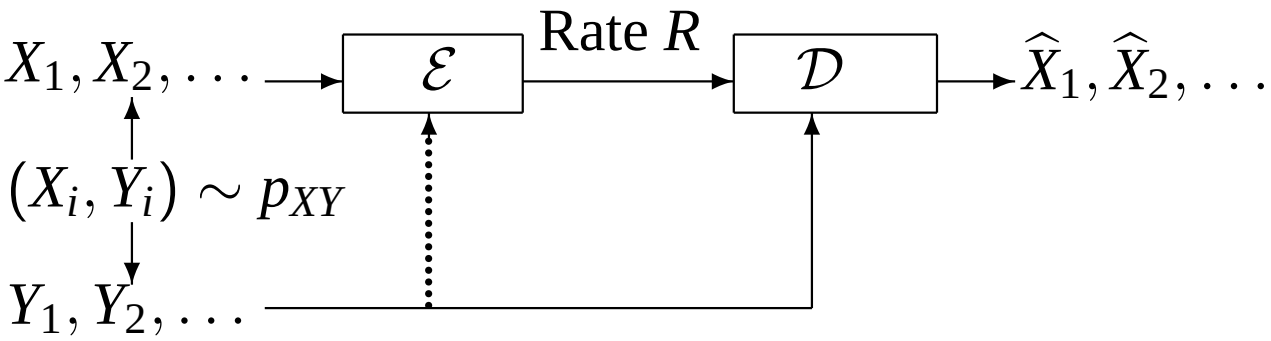
Side-information at the decoder



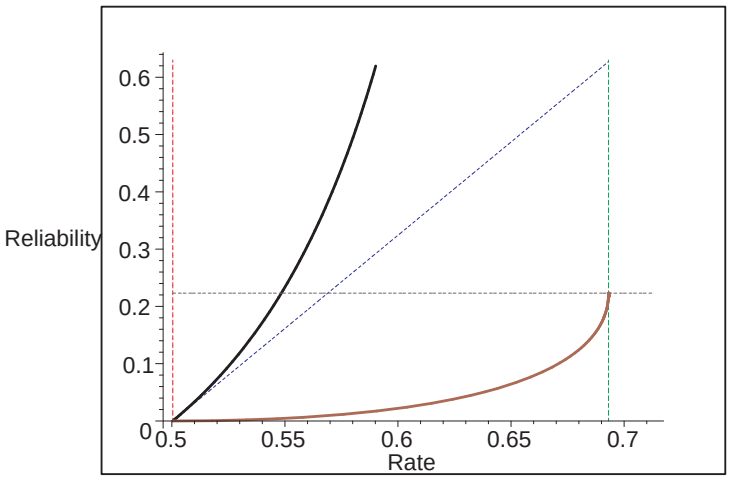
- Encoder may be ignorant of the side-information. (Slepian-Wolf)
- If $P_{X,Y}$ symmetric with uniform marginals, can do no better than $E_b(R)$ with delay. (analogous to Pinsker's argument)

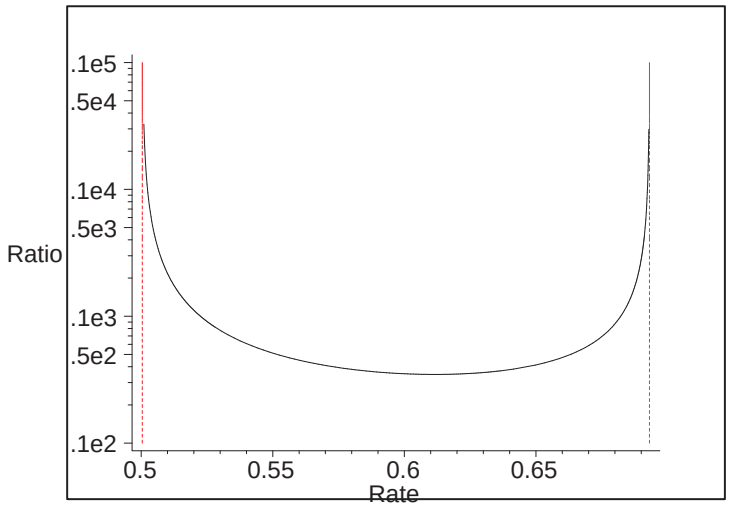


Side-information at the decoder

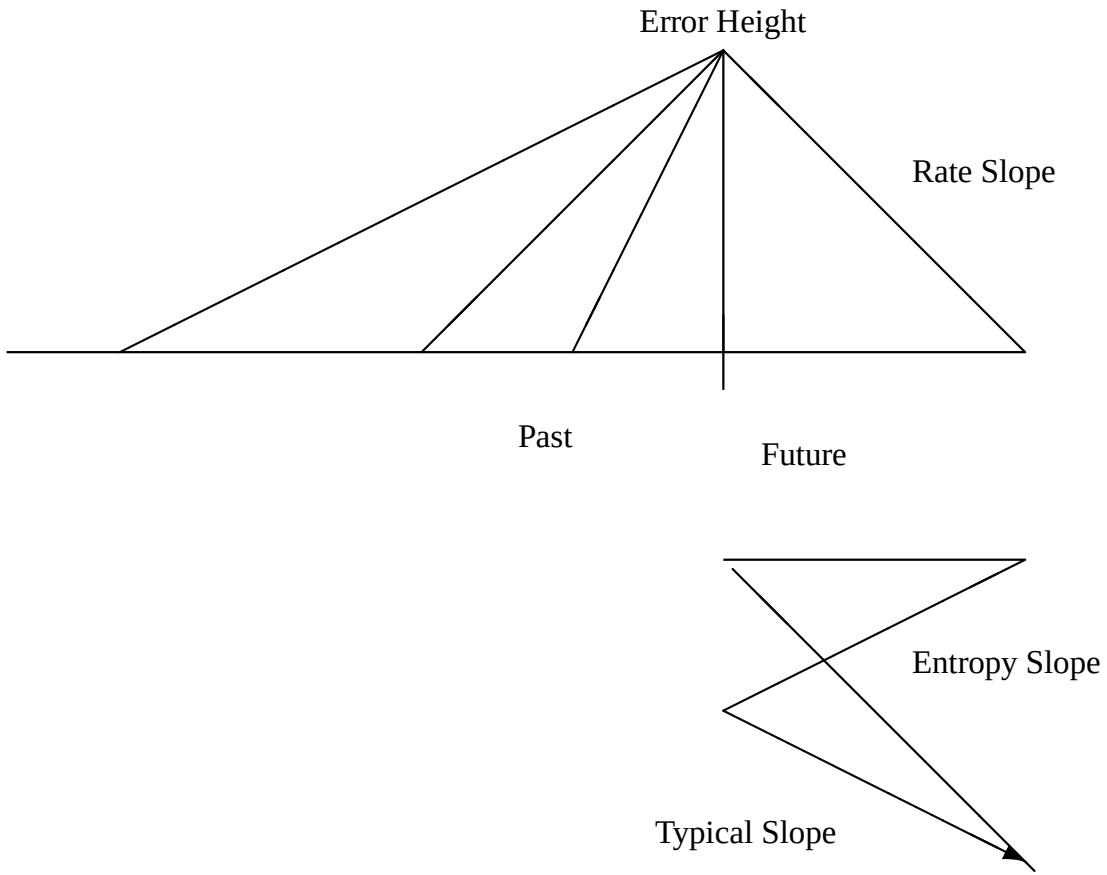


- Encoder may be ignorant of the side-information. (Slepian-Wolf)
- If $P_{X,Y}$ symmetric with uniform marginals, can do no better than $E_b(R)$ with delay. (analogous to Pinsker's argument)





Intuition: Long vs. Large deviations

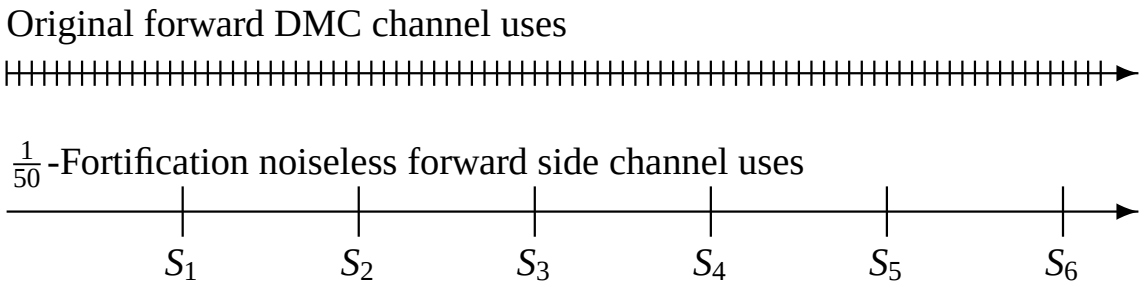


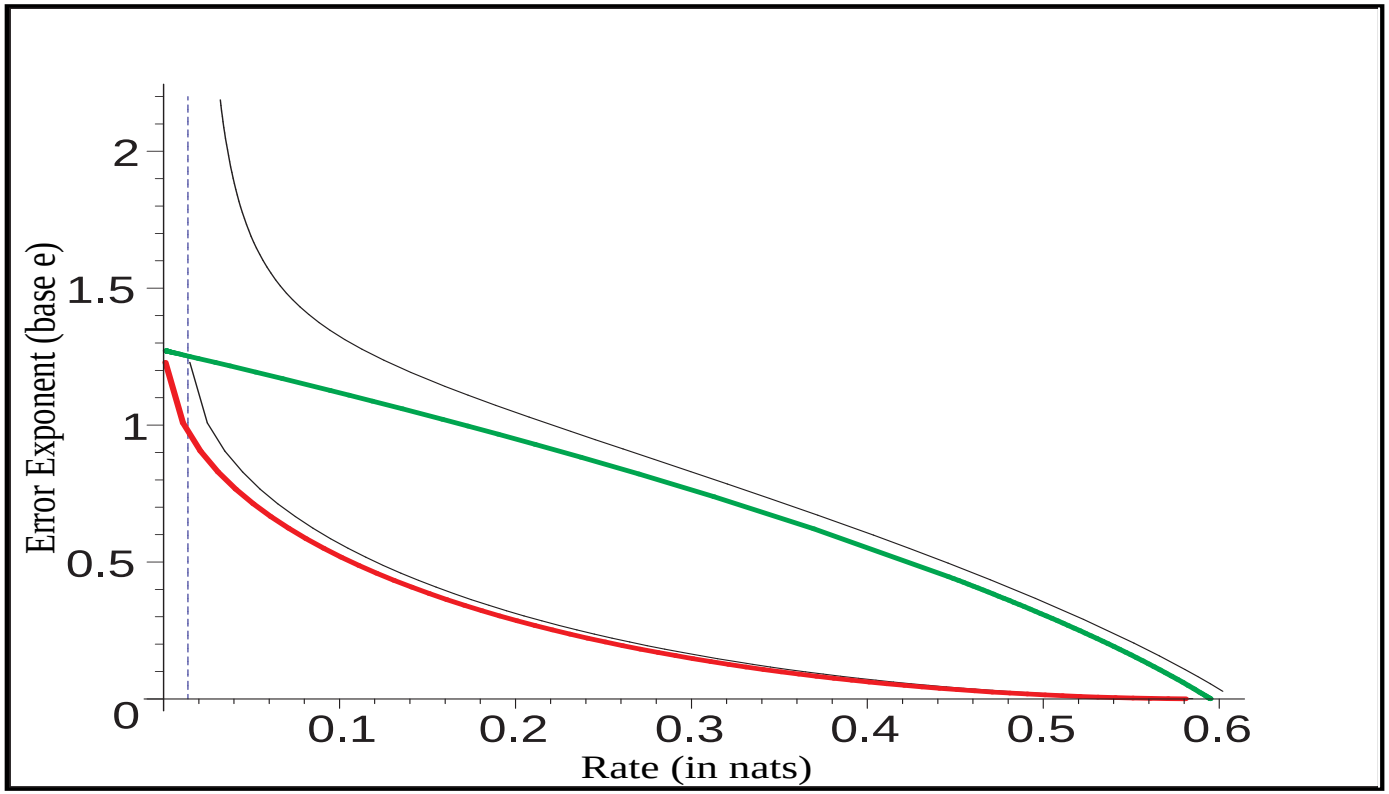
- Shorter deviation periods must be larger.
- Smaller deviations must be over longer periods.

Outline

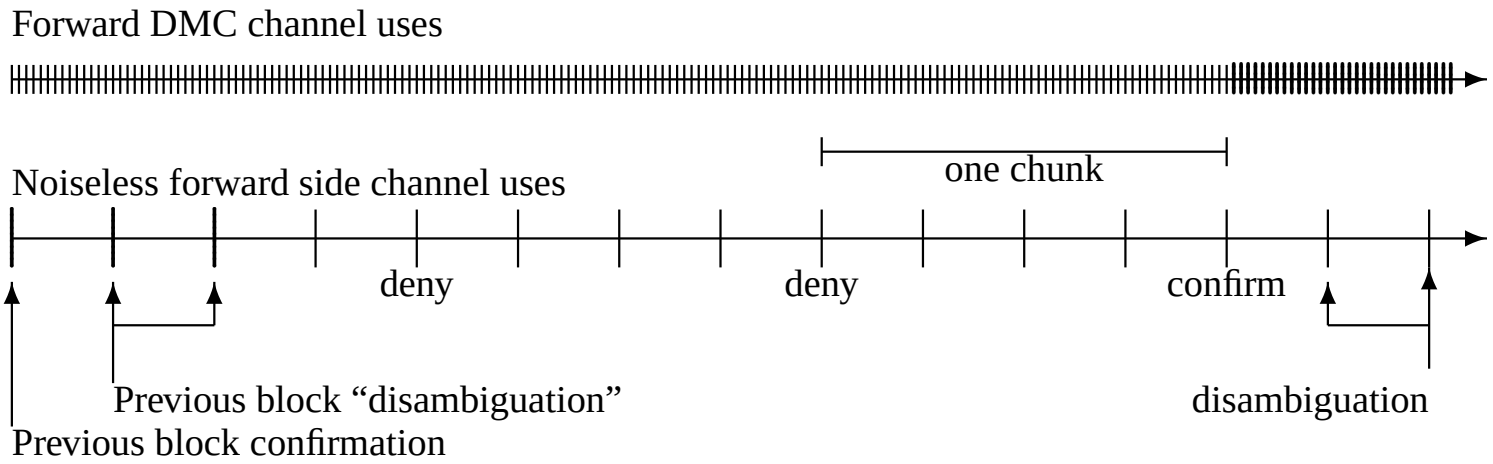
1. Motivation and background
2. Feedback, delay, and error probability: memoryless channels
   - Block-oriented results
   - Idealized case: fixed-delay with hard deadlines
     - ★ The BEC example
     - ★ Without feedback
     - ★ The focusing bound
     - ★ The source-coding analogy
     - ★ Approaching the focusing bound (with help)
     - ★ No help and universality
   - Idealized case: Fixed-delay with soft deadlines
   - Unreliable/Noisy feedback
3. Feedback and memory
4. Conclusions and open problems

A spoonful of “sugar” helps the bits get across.





Harnessing the power of “flow control”

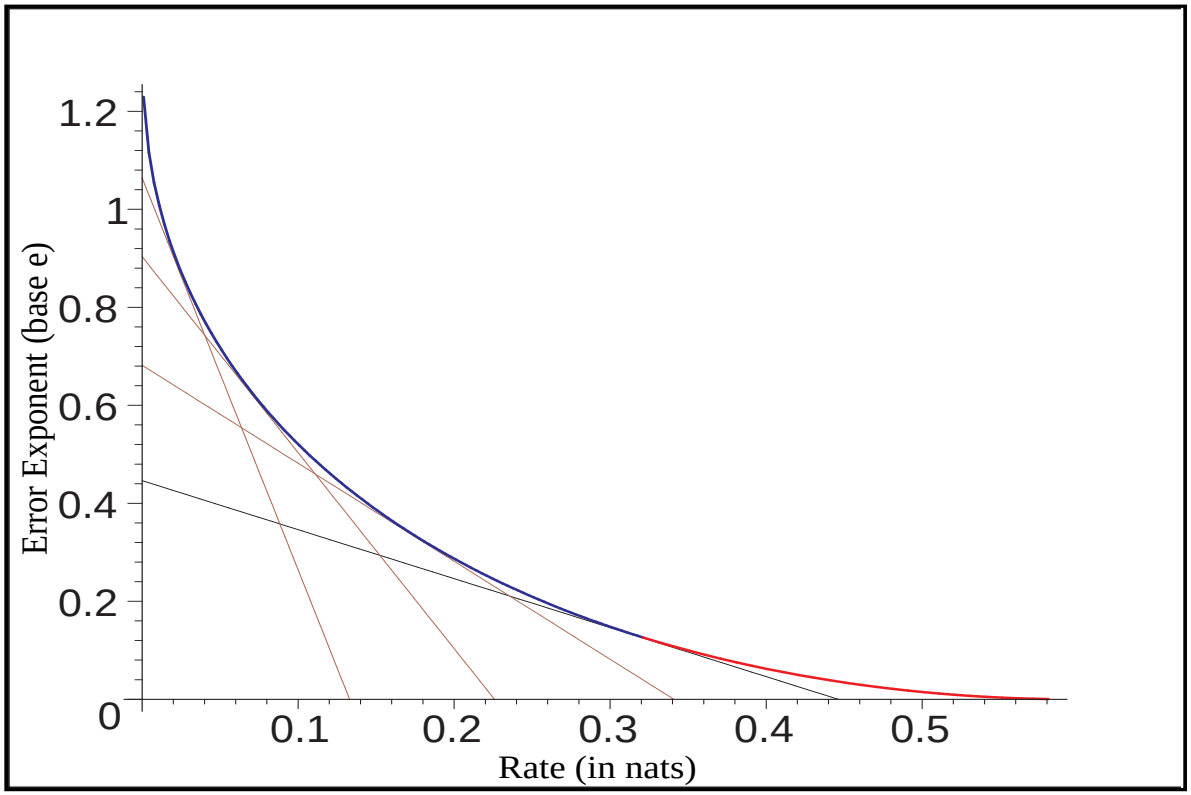


1. Group bits into miniblocks of size nR. ($n \ll d$)
2. Transmit using an ∞-length random codebook.
3. Use the “sugar” to tell decoder when it’s done.

No decoding errors, just queuing plus transmission delays.

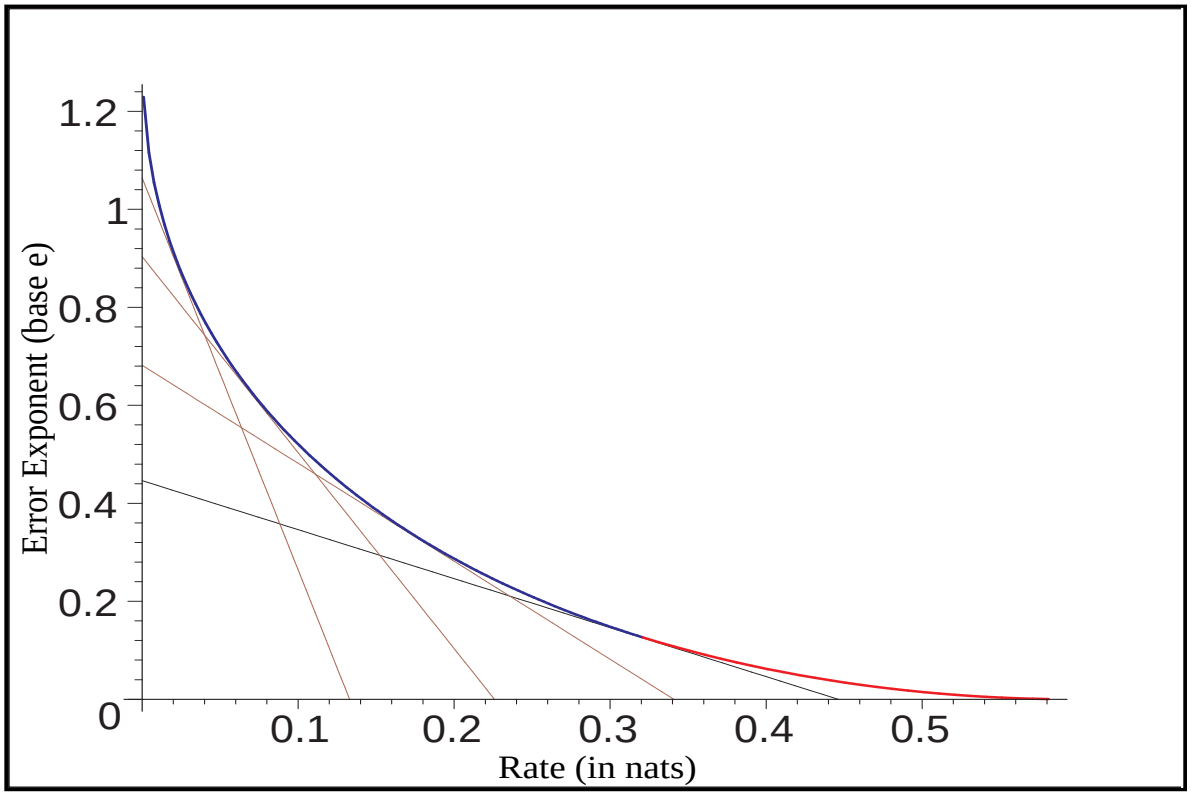
Why this works: operational interpretation of $E_0(\rho)$

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



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

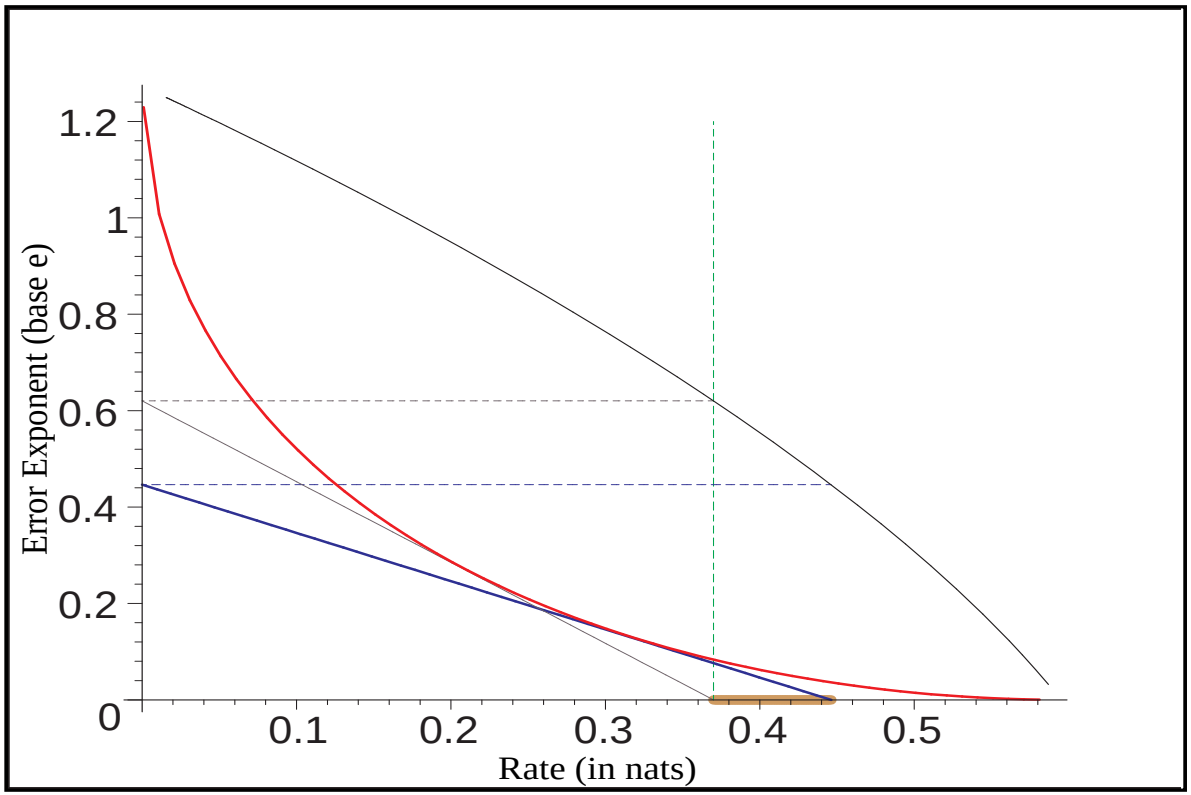
Variable block transmission time T can be bounded by a constant plus a geometric random variable.



Need to do list-decoding at low rates.

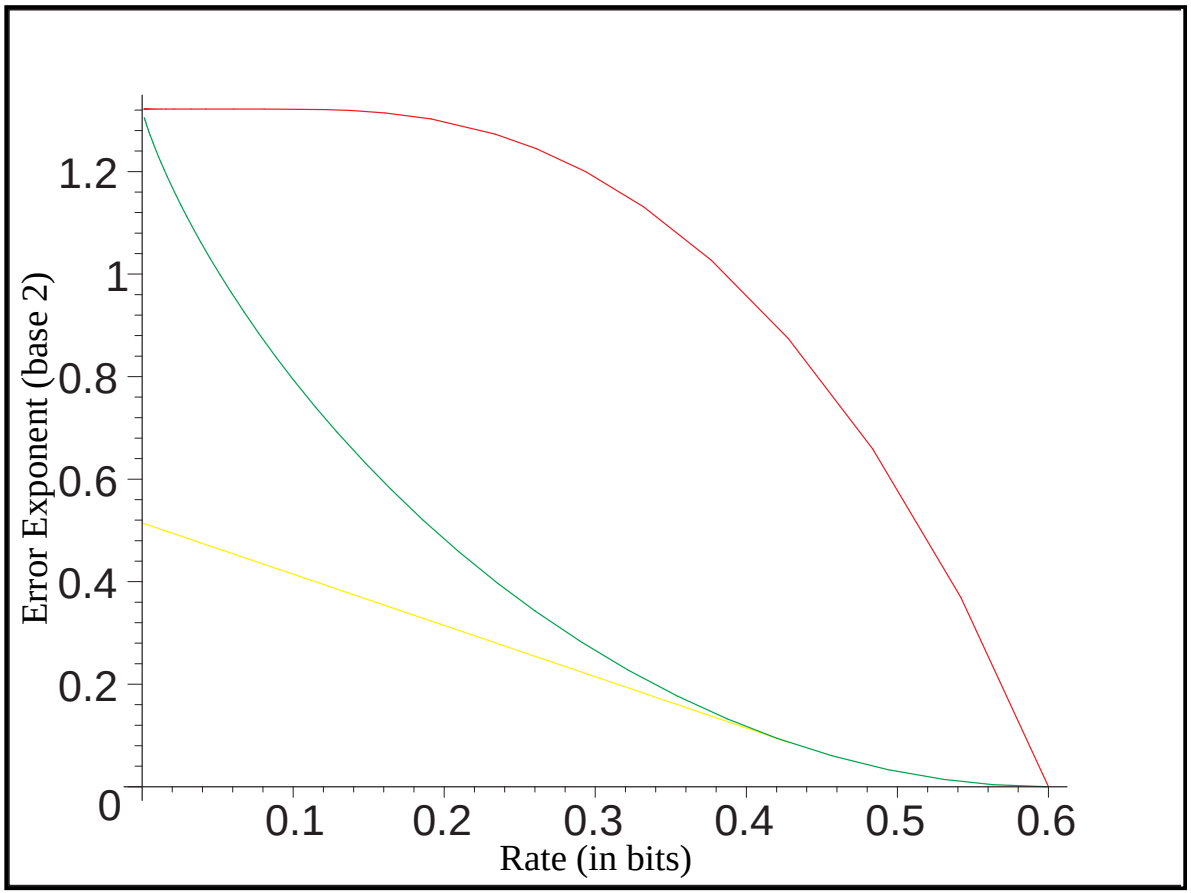
Reduces to the low-rate erasure case

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



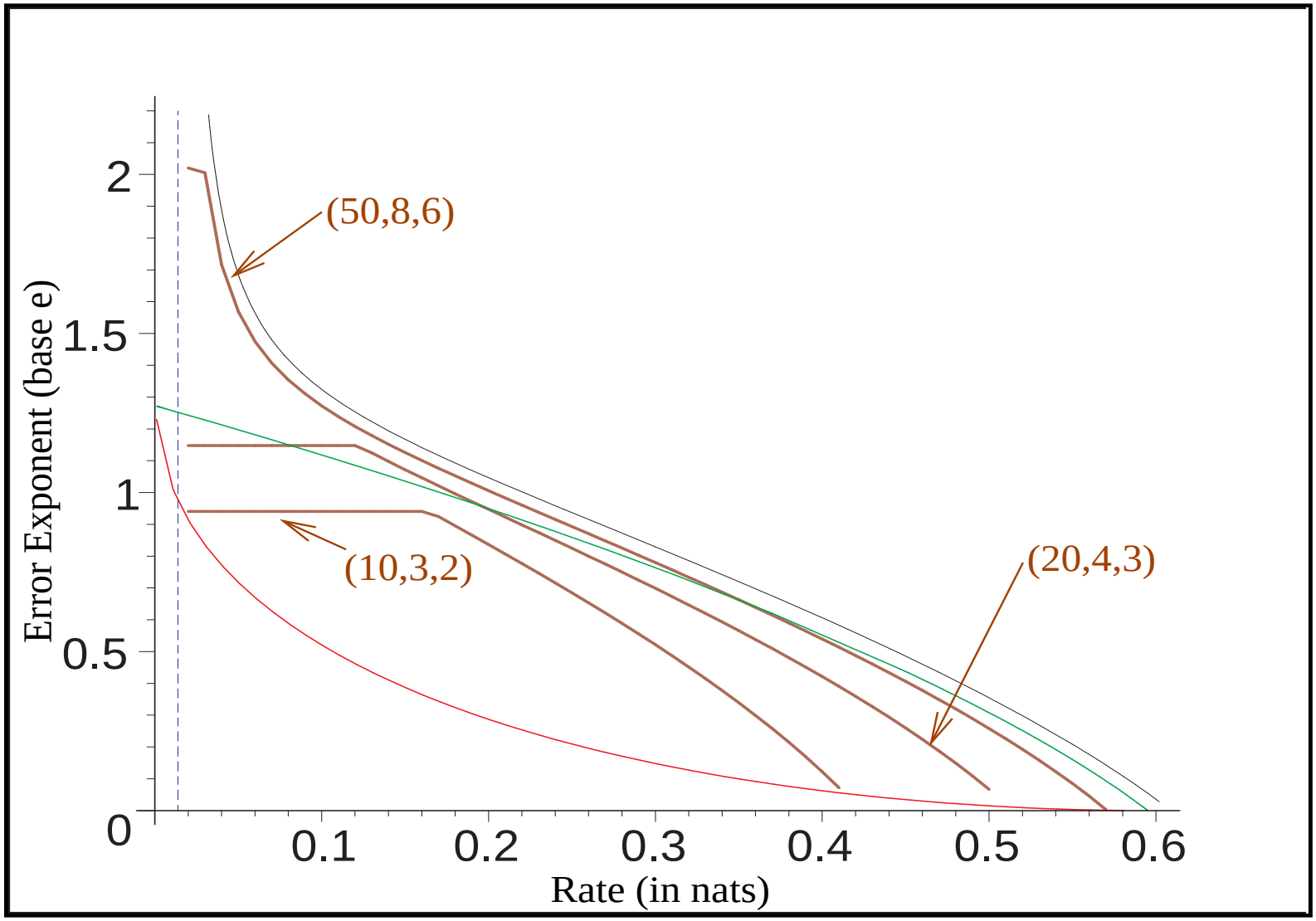
If n large, effective point-message rate $(n(1 - \frac{R}{R'}))^{-1}$ is small.

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

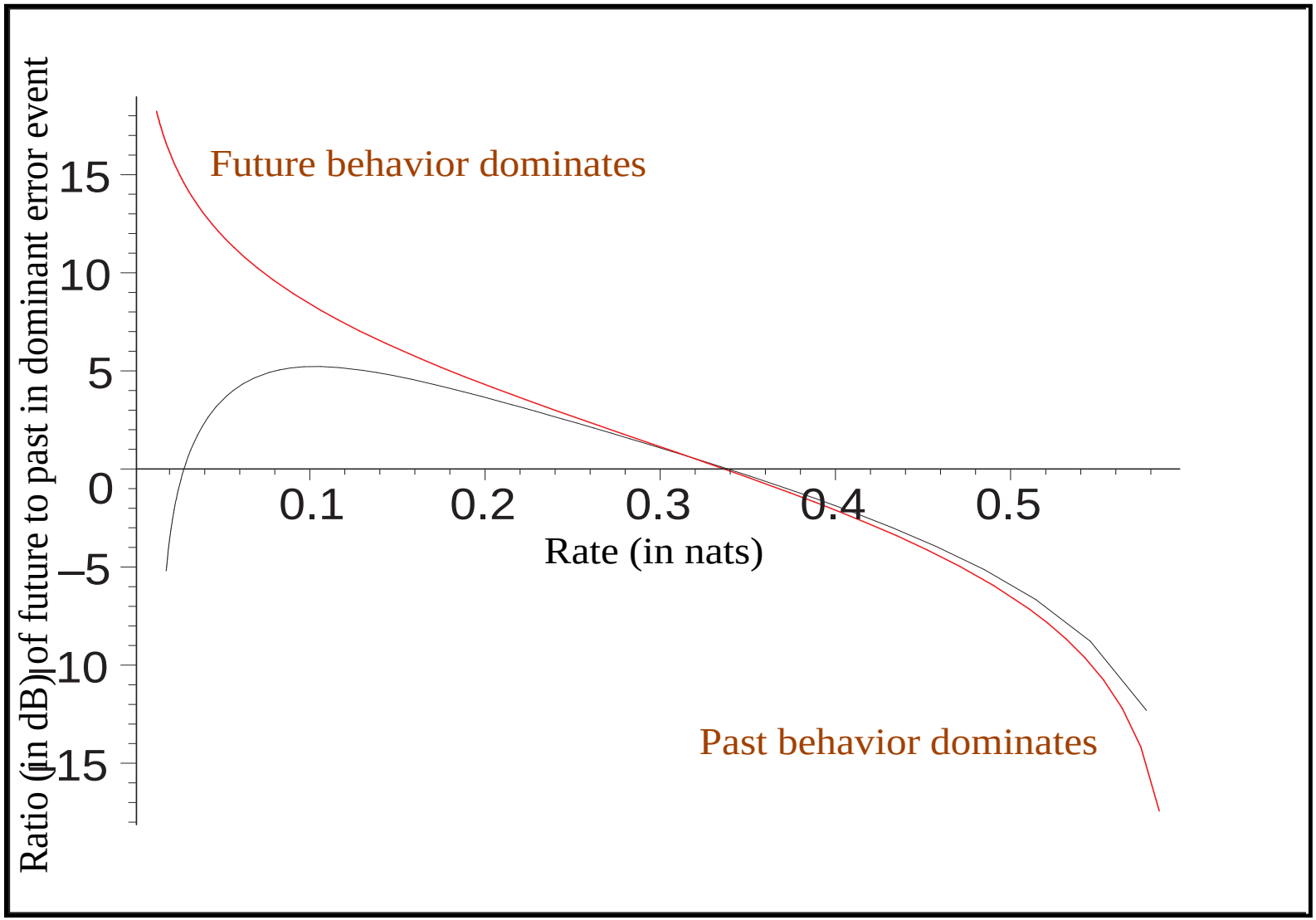


$-\log \delta = E_0(\rho')$ in our context.

Approaching the focusing bound



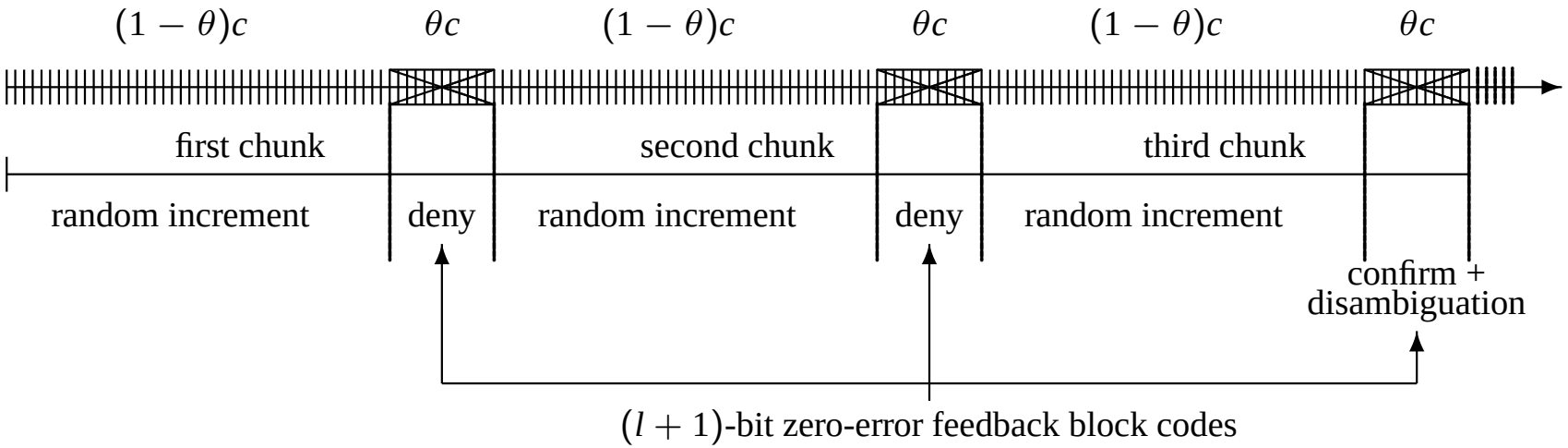
The dominant error events: past vs future



Outline

1. Motivation and background
2. Feedback, delay, and error probability: memoryless channels
   - Block-oriented results
   - Idealized case: fixed-delay with hard deadlines
     - ★ The BEC example
     - ★ Without feedback
     - ★ The focusing bound
     - ★ The source-coding analogy
     - ★ Approaching the focusing bound (with help)
     - ★ No help and universality
   - Idealized case: Fixed-delay with soft deadlines
   - Unreliable/Noisy feedback
3. Feedback and memory
4. Conclusions and open problems

Channels with positive zero-error capacity

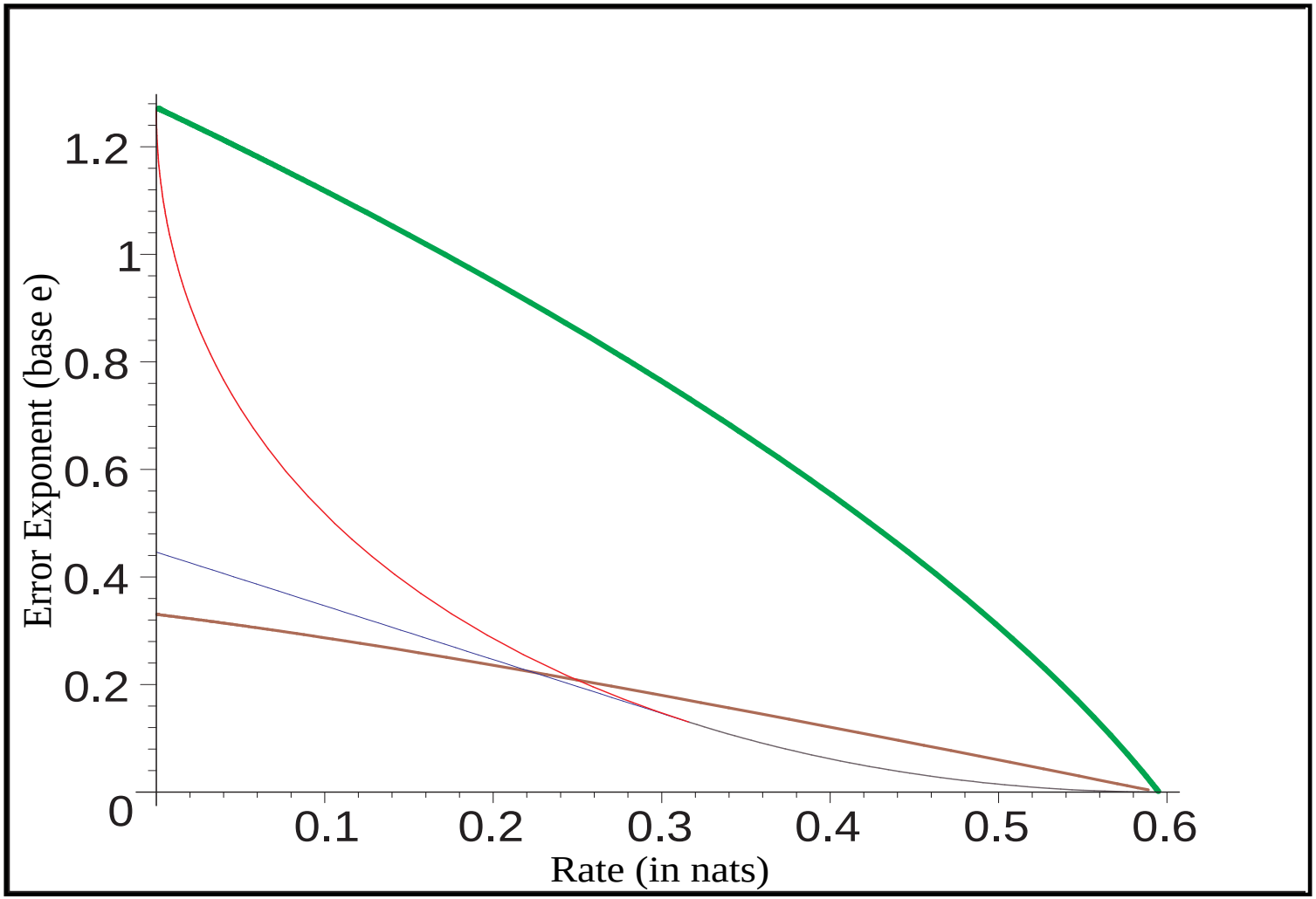


Throw away a fraction θ of channel uses for flow-control overhead

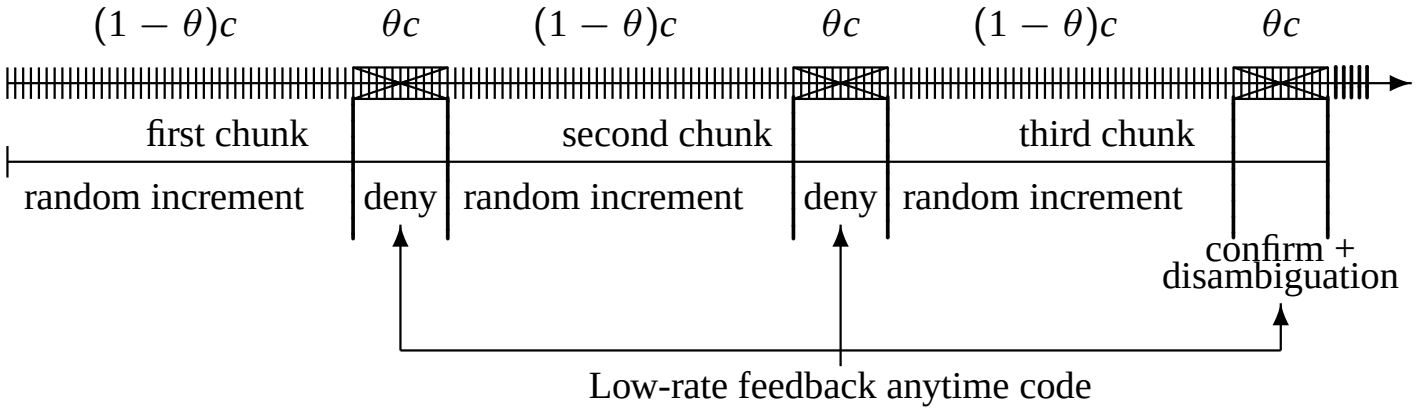
Asymptotically achieves the focusing bound as $\theta \to 0$.

Can do well even without "sugar"

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

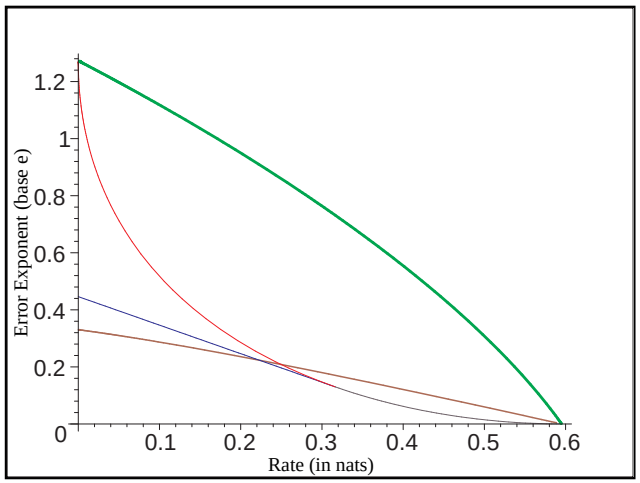


Optimize fraction θ for flow-control



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 $\approx (1-\theta)\frac{E_0(\rho)}{\rho}$



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

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

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

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

Dealing with channel uncertainty

- Compound DMC
  - Fix input distribution.
  - Assume only $I(X;Y) \geq C$ is known.
  - Non-convex set of channels.

Dealing with channel uncertainty

- Compound DMC
  - Fix input distribution.
  - Assume only $I(X;Y) \geq C$ is known.
  - Non-convex set of channels.
- Block-case: MMI decoding achieves $E_r(R)$ for whatever channel is there.

Dealing with channel uncertainty

- Compound DMC
  - Fix input distribution.
  - Assume only $I(X;Y) \geq C$ is known.
  - Non-convex set of channels.
- Block-case: MMI decoding achieves $E_r(R)$ for whatever channel is there.
  - But how big is this guaranteed to be?

Dealing with channel uncertainty

- Compound DMC
  - Fix input distribution.
  - Assume only $I(X;Y) \geq C$ is known.
  - Non-convex set of channels.
- Block-case: MMI decoding achieves $E_r(R)$ for whatever channel is there.
  - But how big is this guaranteed to be?
  - Universal bound (Gallager '71 and Lapidoth/Telatar '98):

$$E_r(R) \geq \frac{(C-R)^2}{8/e^2 + 4[\ln|\mathcal{Y}|]^2}$$

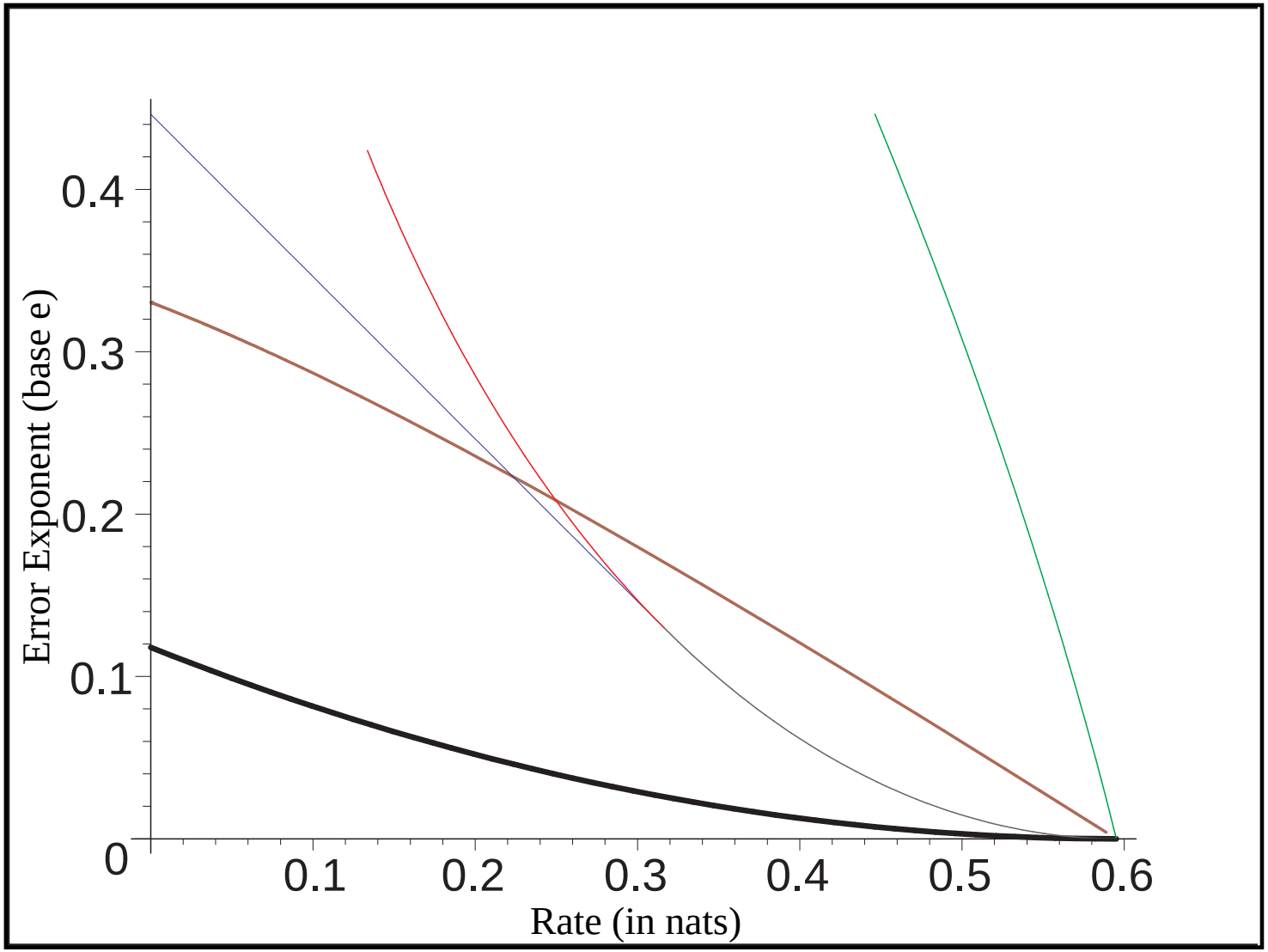
Dealing with channel uncertainty

- Compound DMC
  - Fix input distribution.
  - Assume only $I(X;Y) \geq C$ is known.
  - Non-convex set of channels.
- Block-case: MMI decoding achieves $E_r(R)$ for whatever channel is there.
  - But how big is this guaranteed to be?
  - Universal bound (Gallager '71 and Lapidoth/Telatar '98):

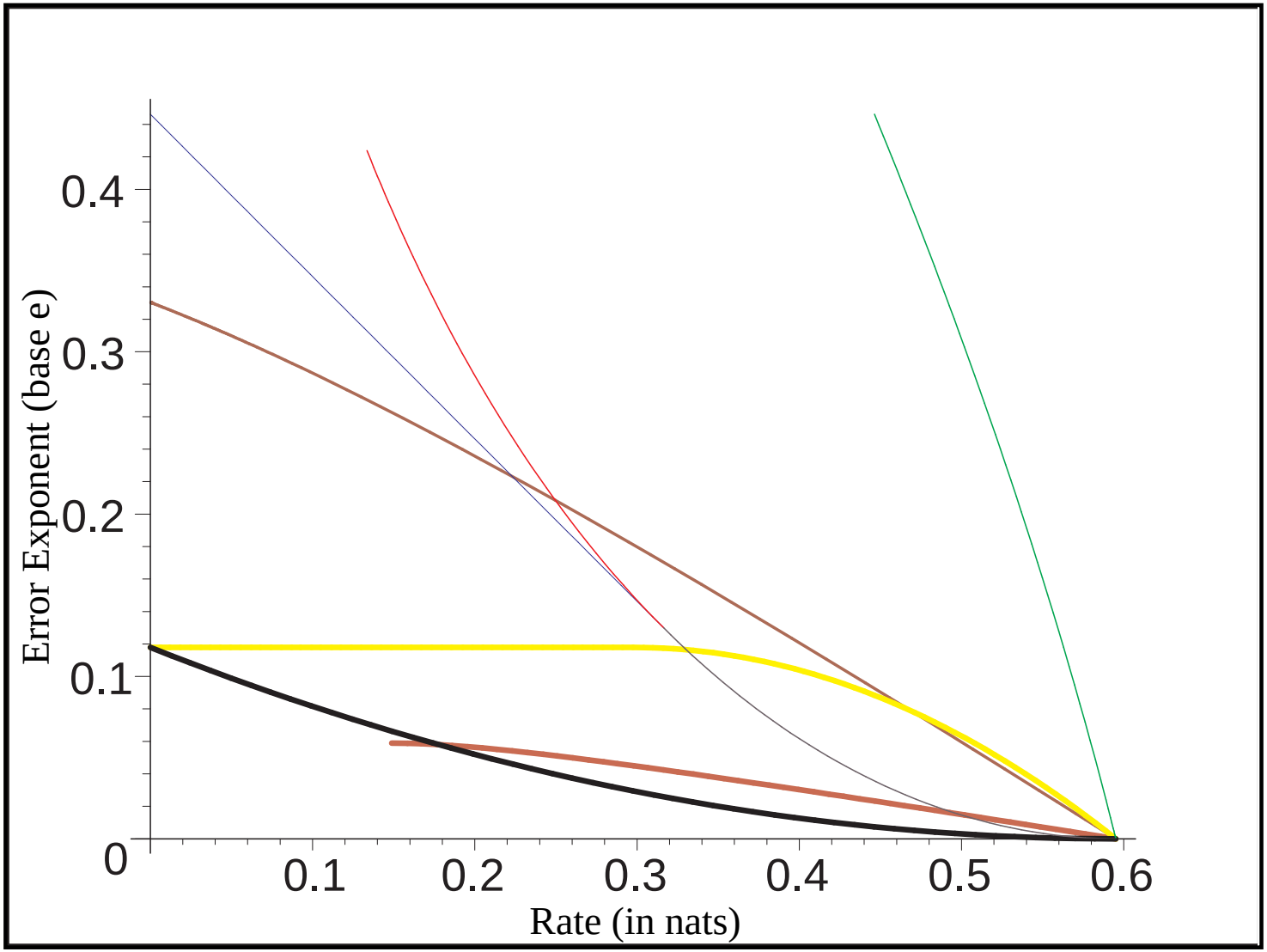
$$E_r(R) \geq \frac{(C-R)^2}{8/e^2 + 4[\ln|\mathcal{Y}|]^2}$$

- Can translate to delay and focusing-bound.

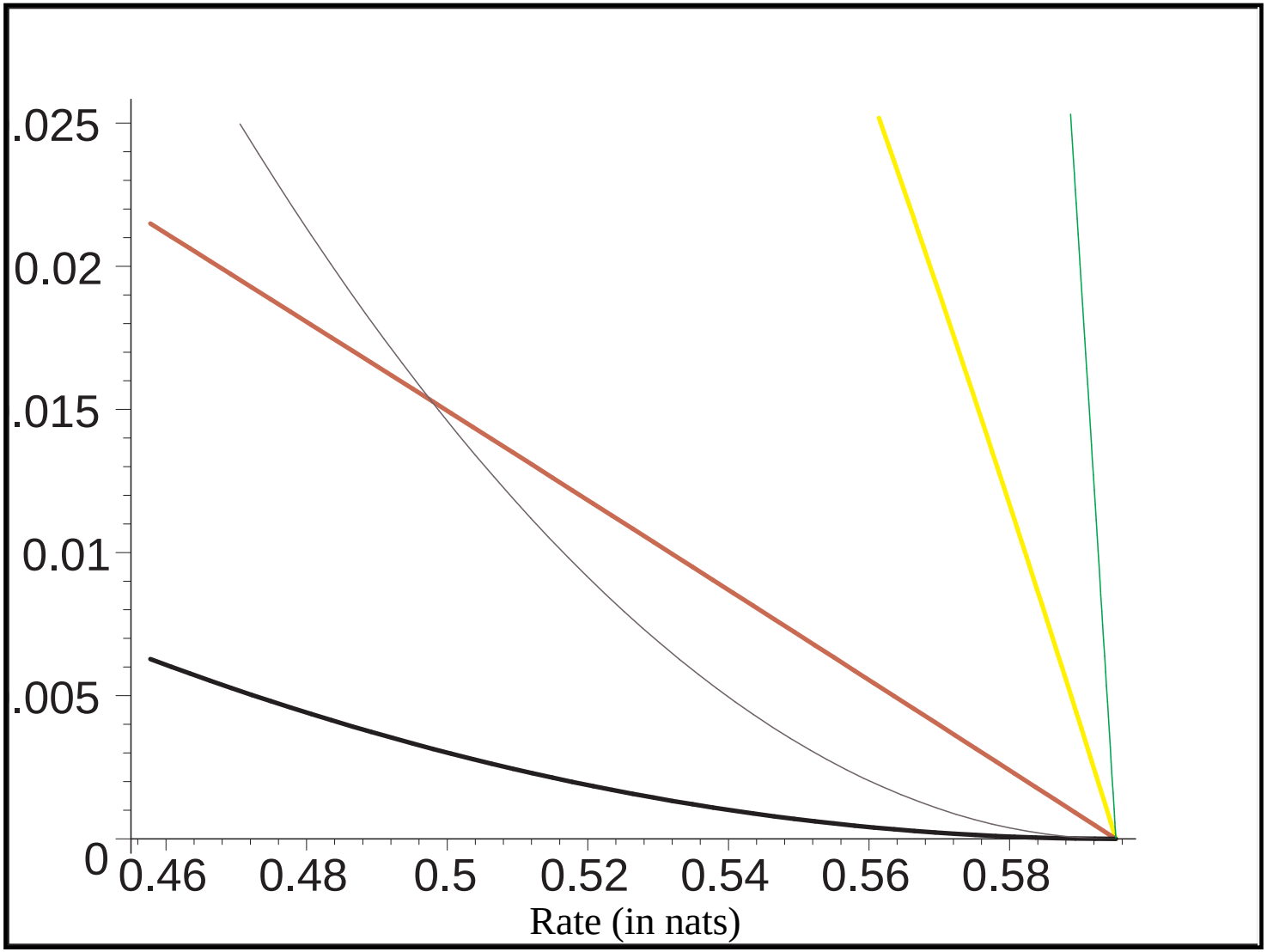
Universal bounds illustrated



Universal bounds illustrated



Universal bounds illustrated



Hard deadline coding final comments

- For known channels, computation per channel use does not depend on probability of error = *infinite computational exponent*

Hard deadline coding final comments

- For known channels, 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.

Hard deadline coding final comments

- For known channels, 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.

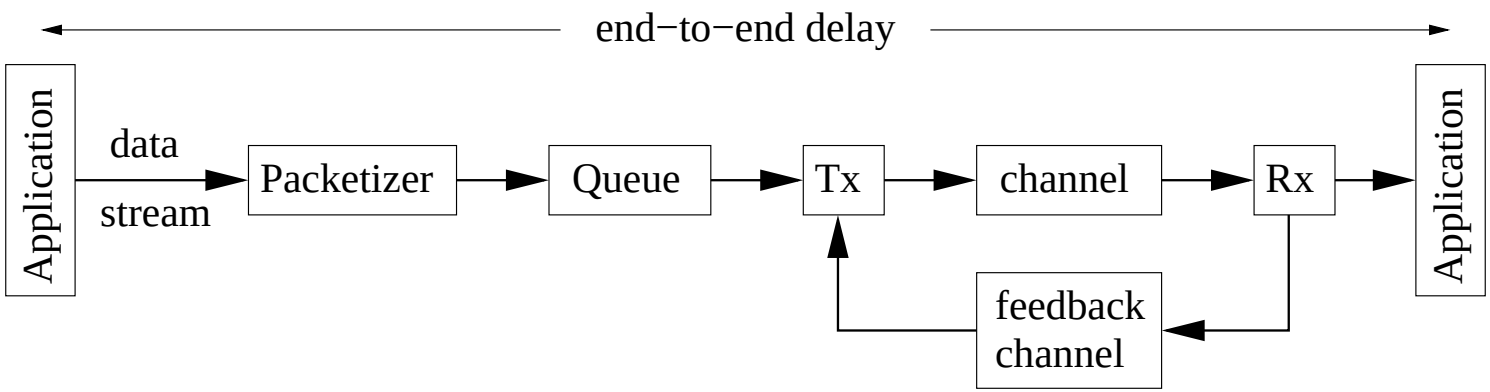
Hard deadline coding final comments

- For known channels, 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.
- Even with unknown channels, get strictly positive slope at capacity.

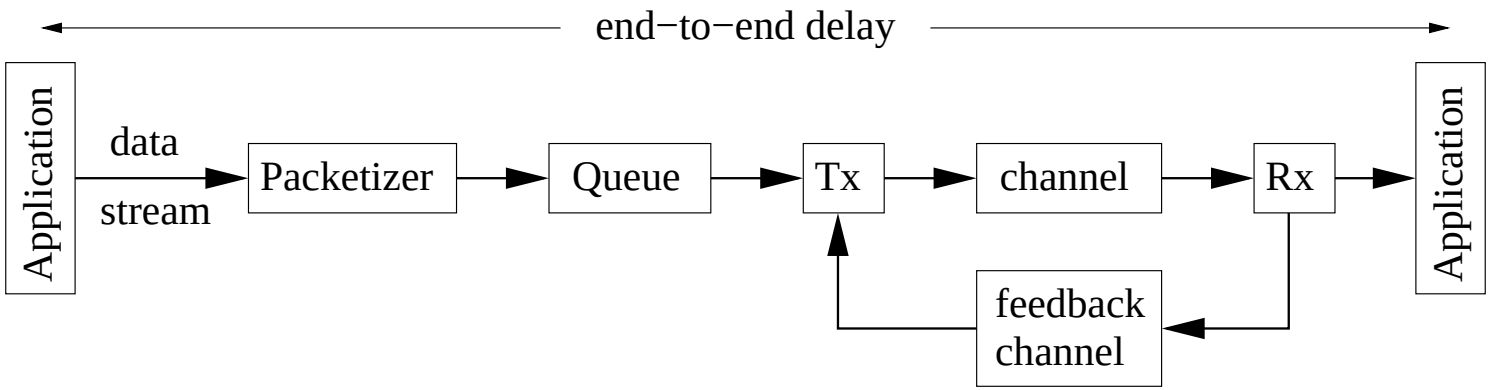
Outline

1. Motivation and background
2. Feedback, delay, and error probability: memoryless channels
   - Block-oriented results
   - Idealized case: fixed-delay with hard deadlines
   - Idealized case: fixed-delay with soft deadlines
     - ★ Kudryashov's scheme
     - ★ Hallucination bound
   - Unreliable/Noisy feedback
3. Feedback and memory
4. Conclusions and open problems

A streaming data perspective on soft deadlines

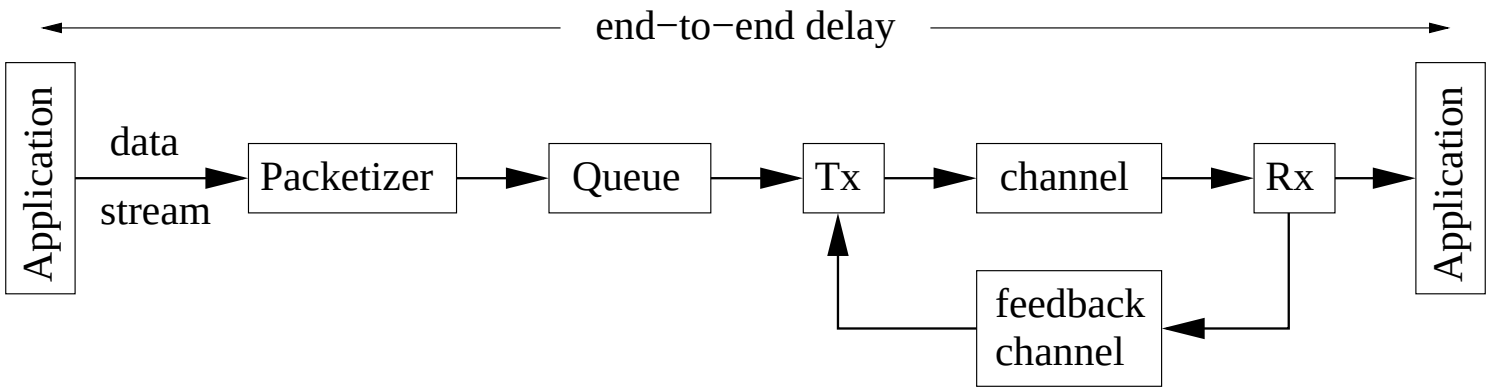


A streaming data perspective on soft deadlines



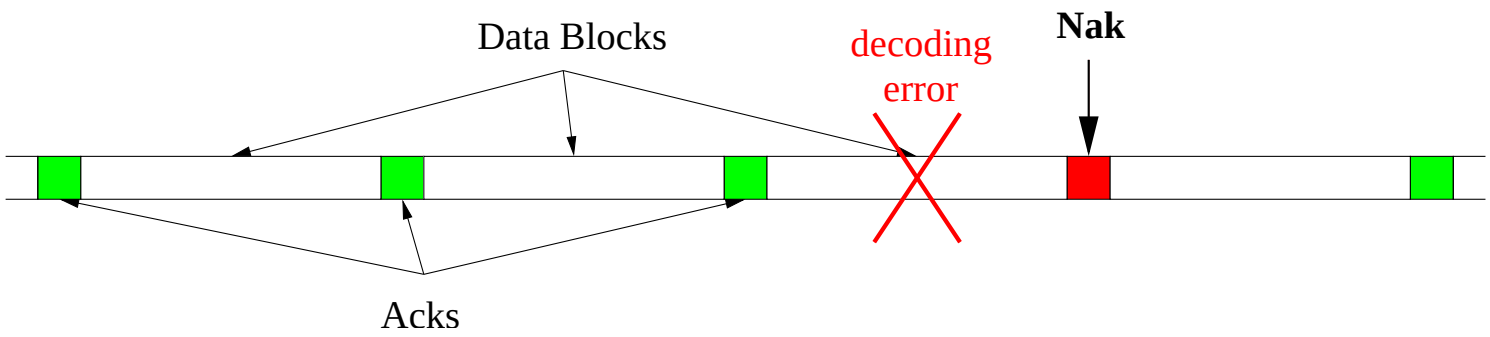
- Queue is optional. Needed if retransmissions are required

A streaming data perspective on soft deadlines



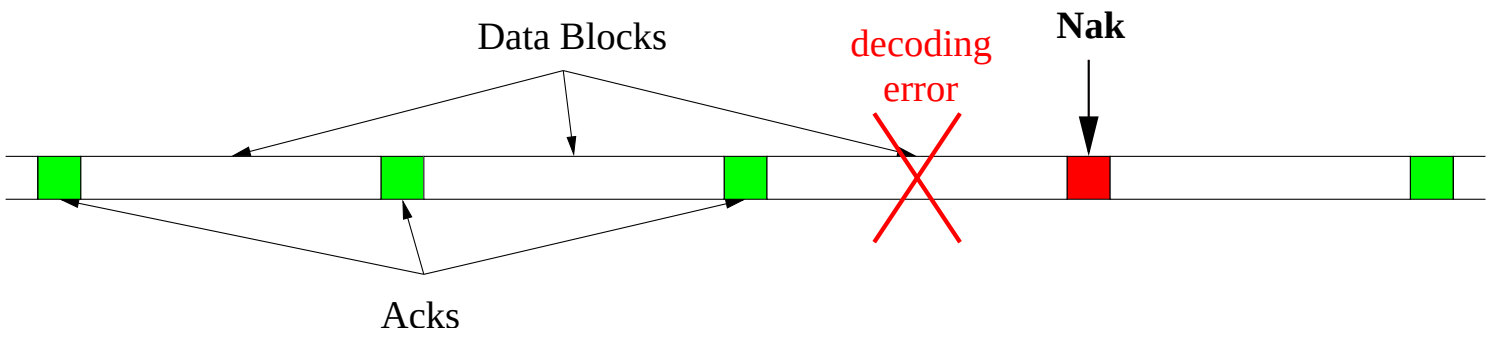
- Queue is optional. Needed if retransmissions are required
- If retransmissions are rare, then **expected** end-to-end delay is dominated by the Tx to Rx delay.

An opportunity presents itself



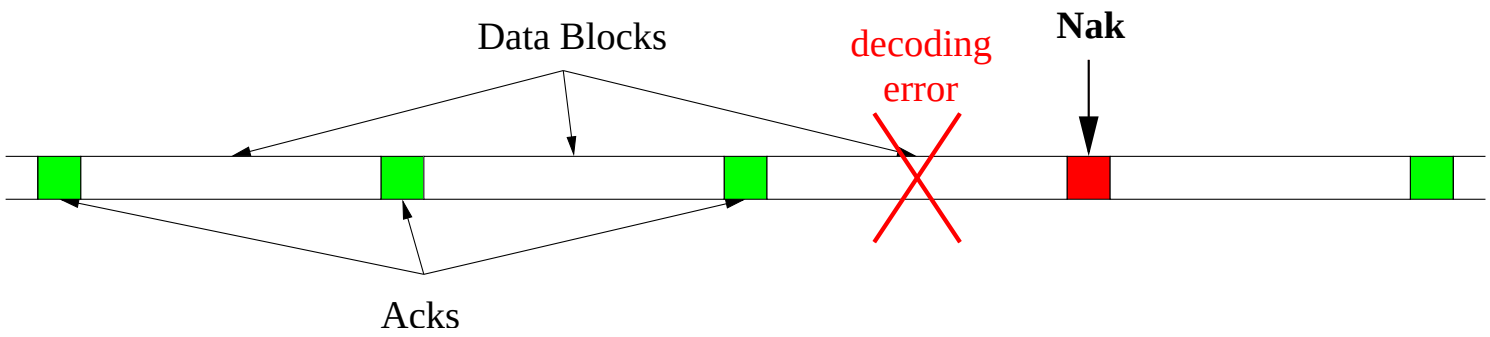
- Erasures are rare so most messages are confirmed.

An opportunity presents itself



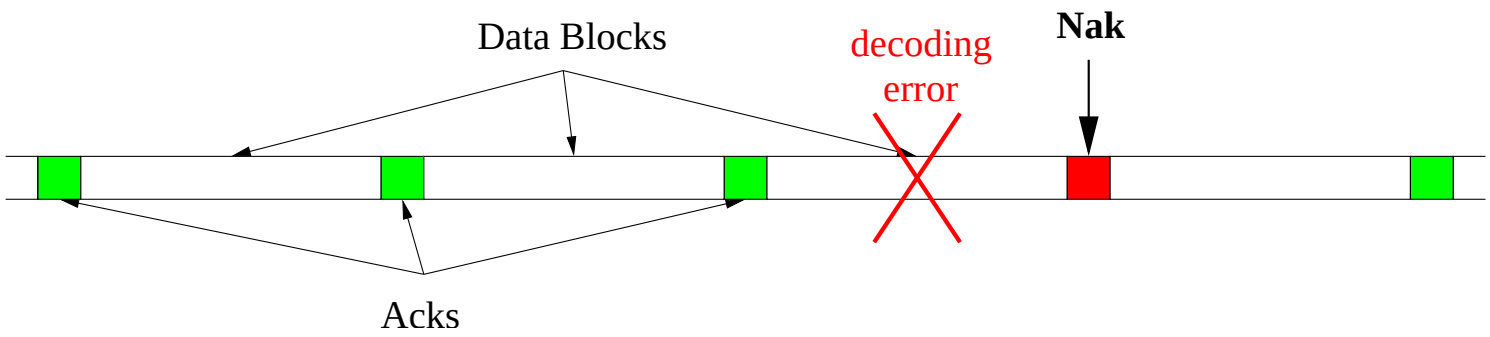
- Erasures are rare so most messages are confirmed.
- We are wasting channel uses.

An opportunity presents itself



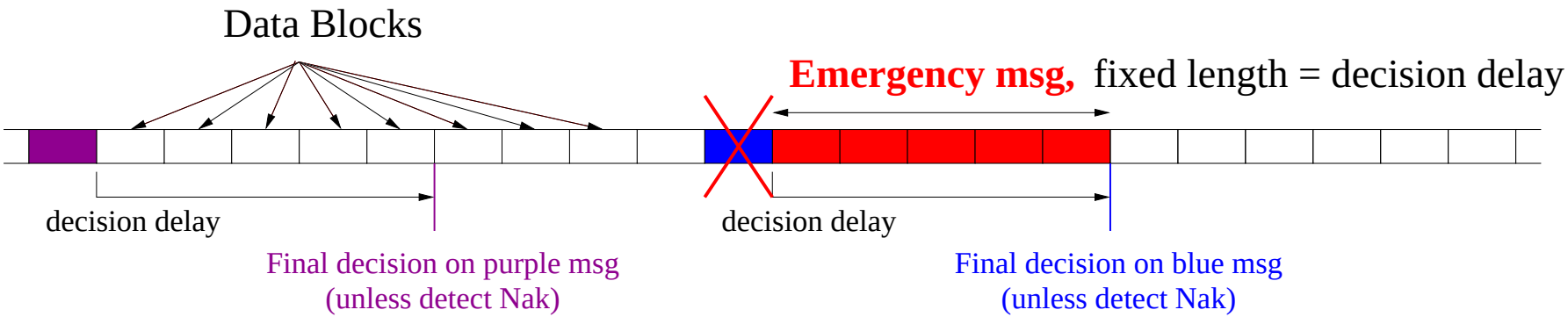
- Erasures are rare so most messages are confirmed.
- We are wasting channel uses.
- What if we only sent NAKs when needed?

An opportunity presents itself



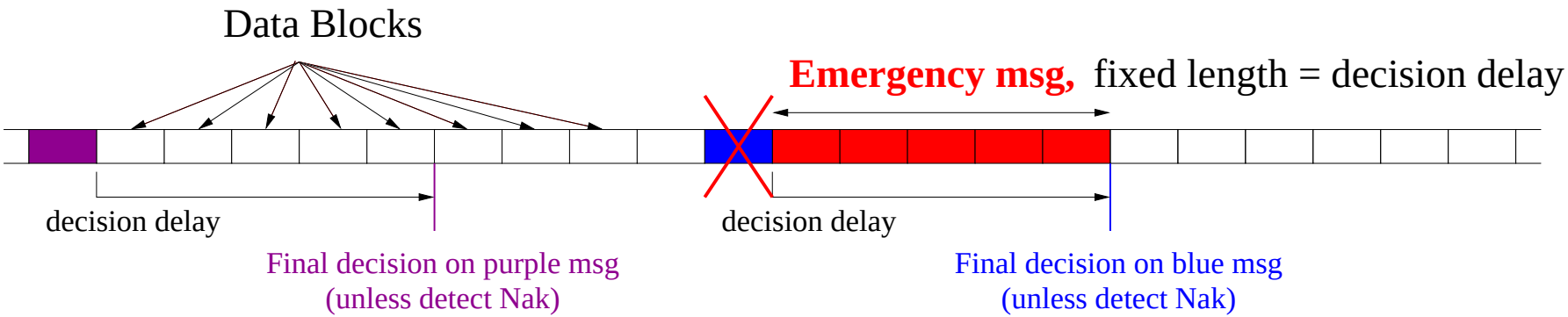
- Erasures are rare so most messages are confirmed.
- We are wasting channel uses.
- What if we only sent NAKs when needed?
- Have a special message for this purpose.

Sliding blocks with collective punishment only (Kudryashov-79)



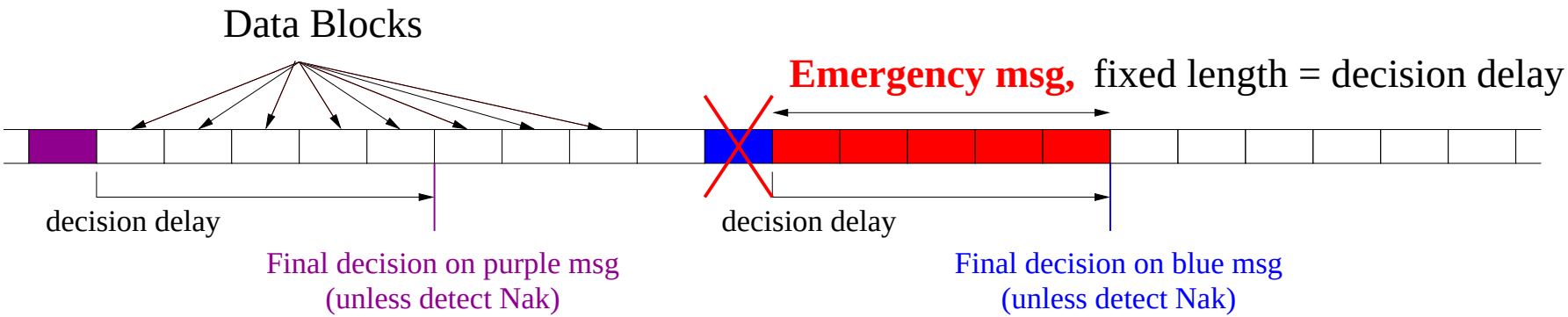
- Make packet size n much smaller than soft deadline d.

Sliding blocks with collective punishment only (Kudryashov-79)



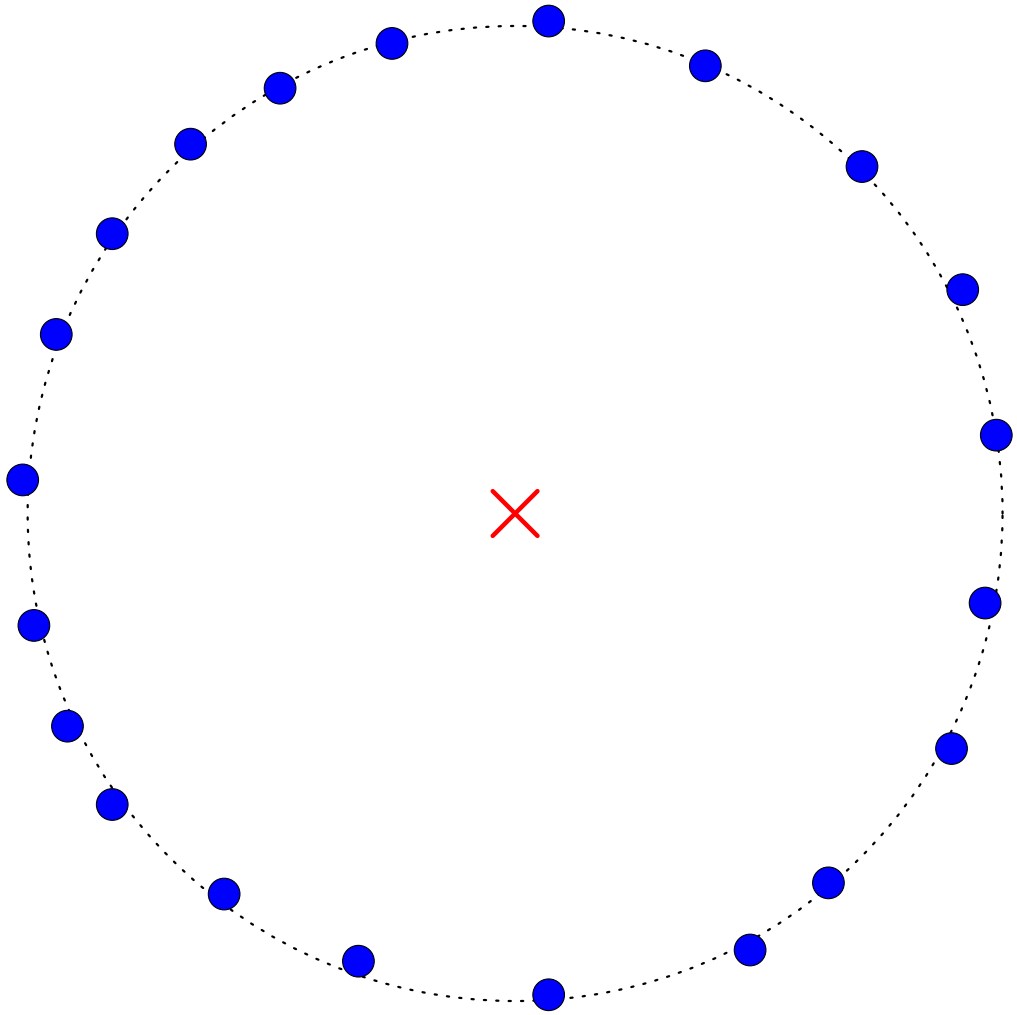
- Make packet size n much smaller than soft deadline d.
- A NAK collectively denies the past $\frac{d}{n} - 1$ packets

Sliding blocks with collective punishment only (Kudryashov-79)

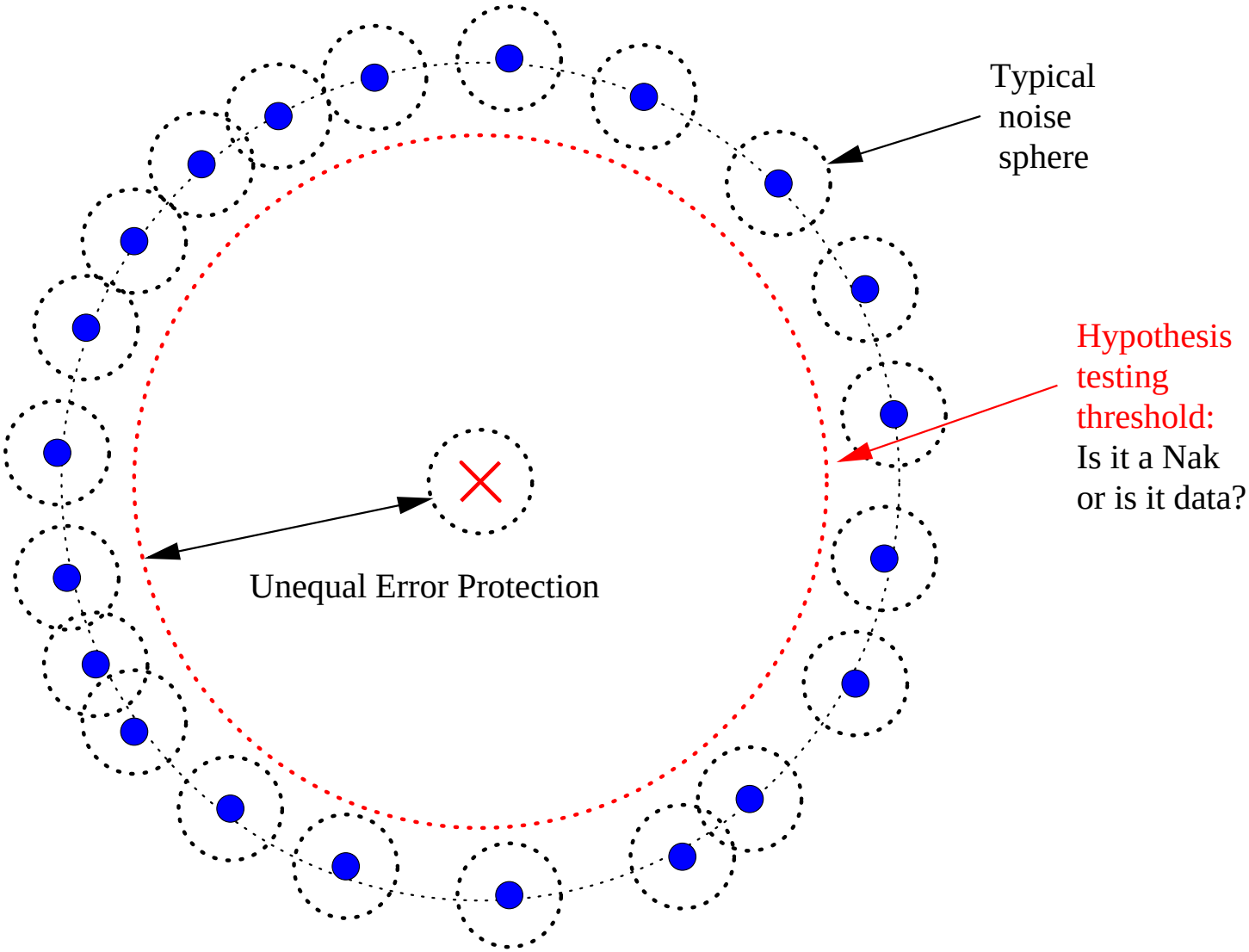


- Make packet size n much smaller than soft deadline d.
- A NAK collectively denies the past $\frac{d}{n} - 1$ packets
- Error only if $\frac{d}{n} - 1$ NAKs are all missed

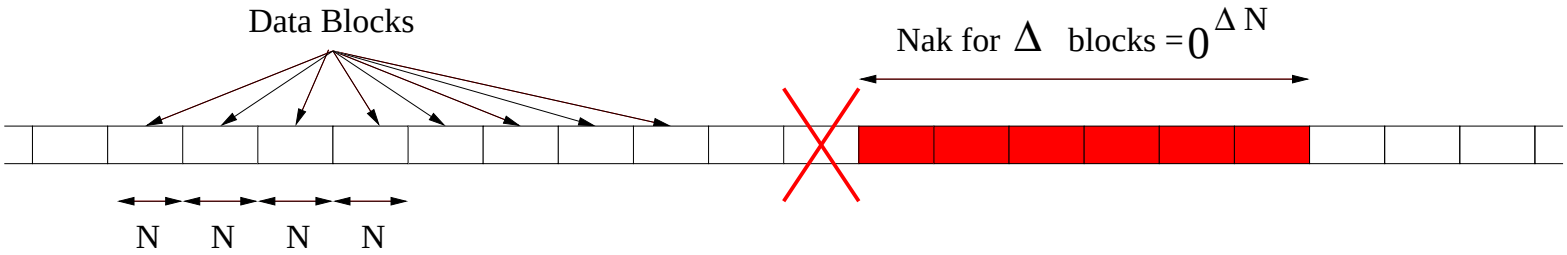
Unequal error protection required in codebook

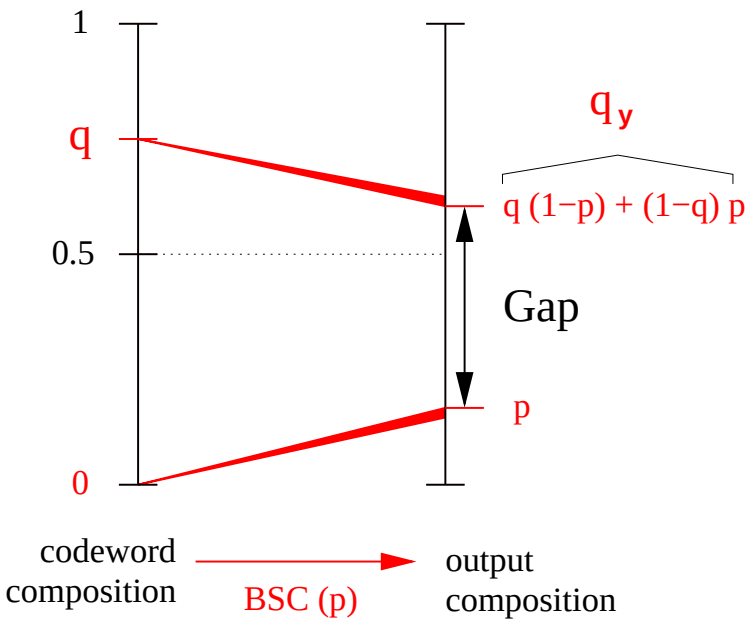


Unequal error protection required in codebook



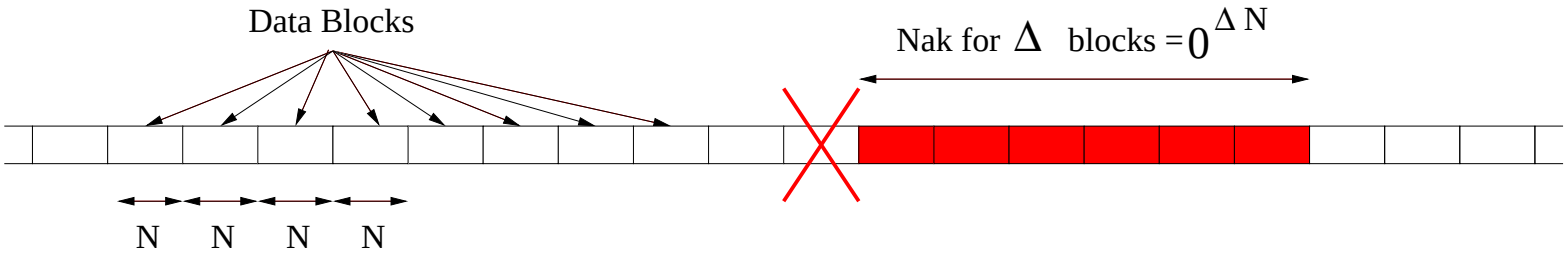
Specialize to BSC case

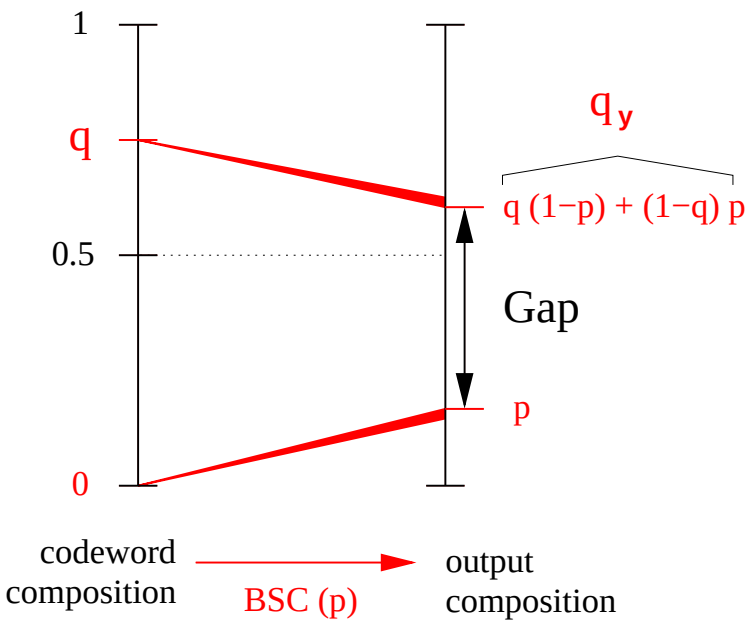




- Use all zero for NAK

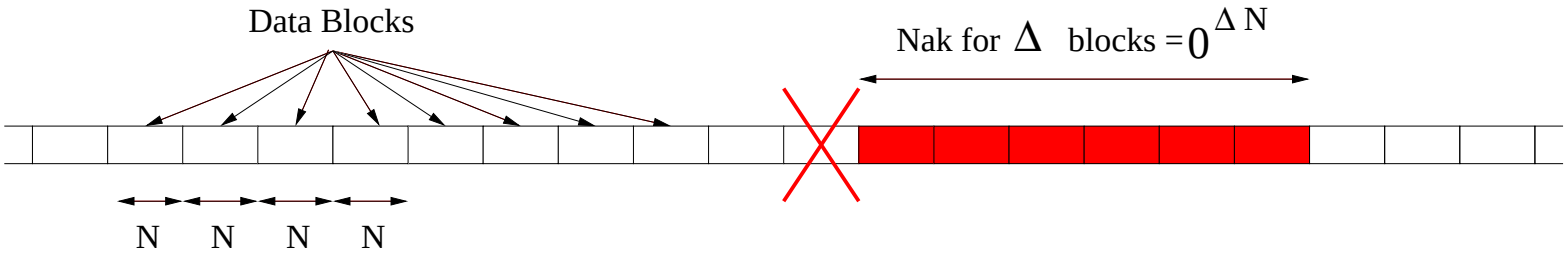
Specialize to BSC case

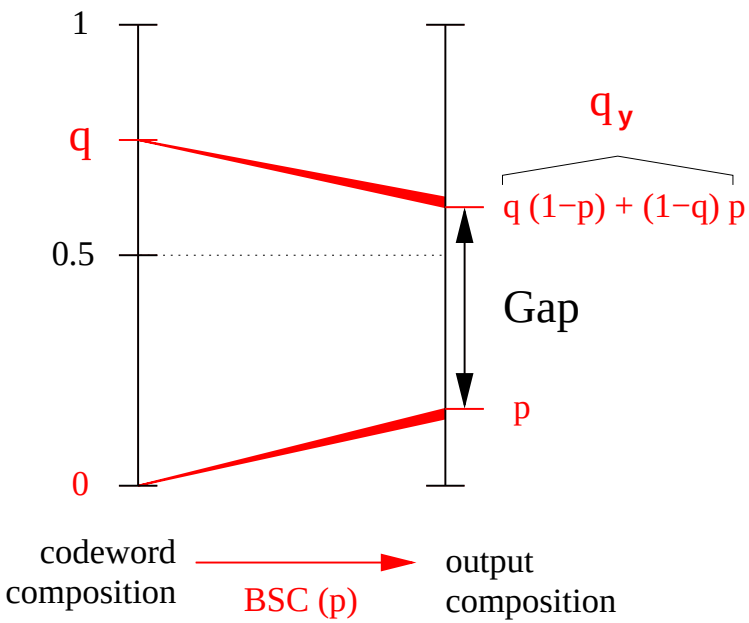




- Use all zero for NAK
- Use composition q code for data: $R < H(q) - H(p)$

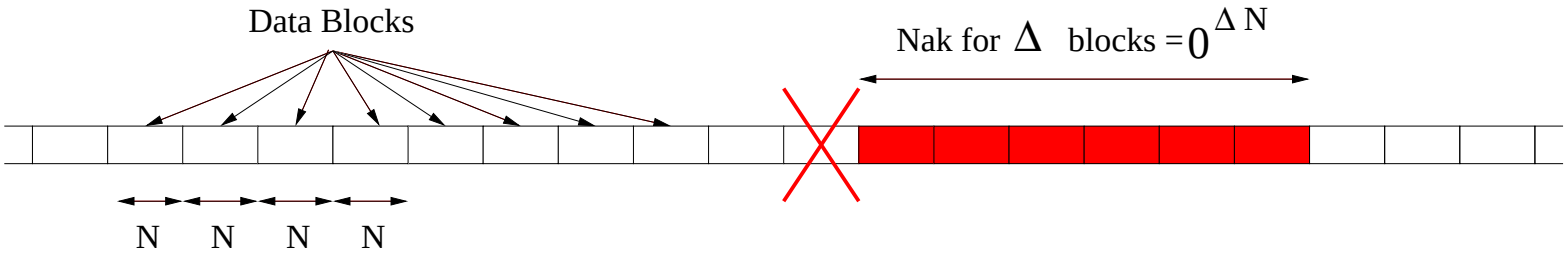
Specialize to BSC case

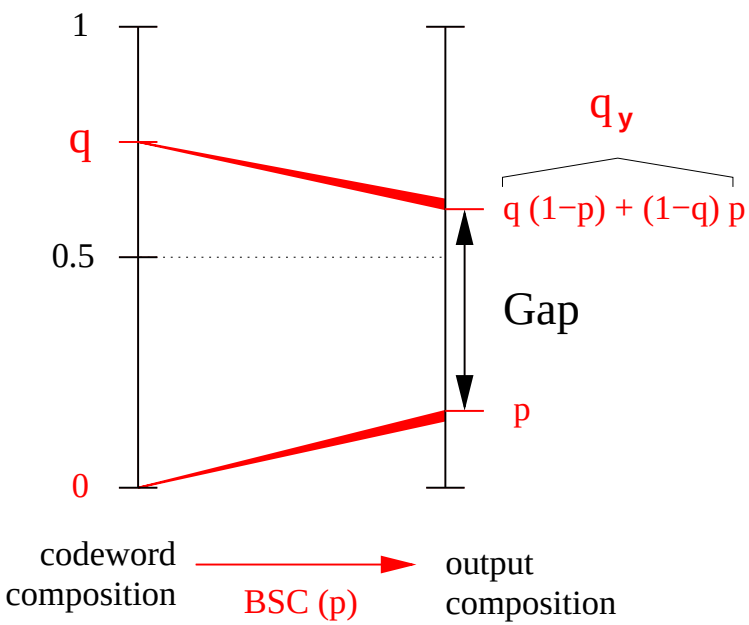




- Use all zero for NAK
- Use composition q code for data: $R < H(q) - H(p)$
- Probability of missed NAK is $2^{-ND(q_y||p)}$

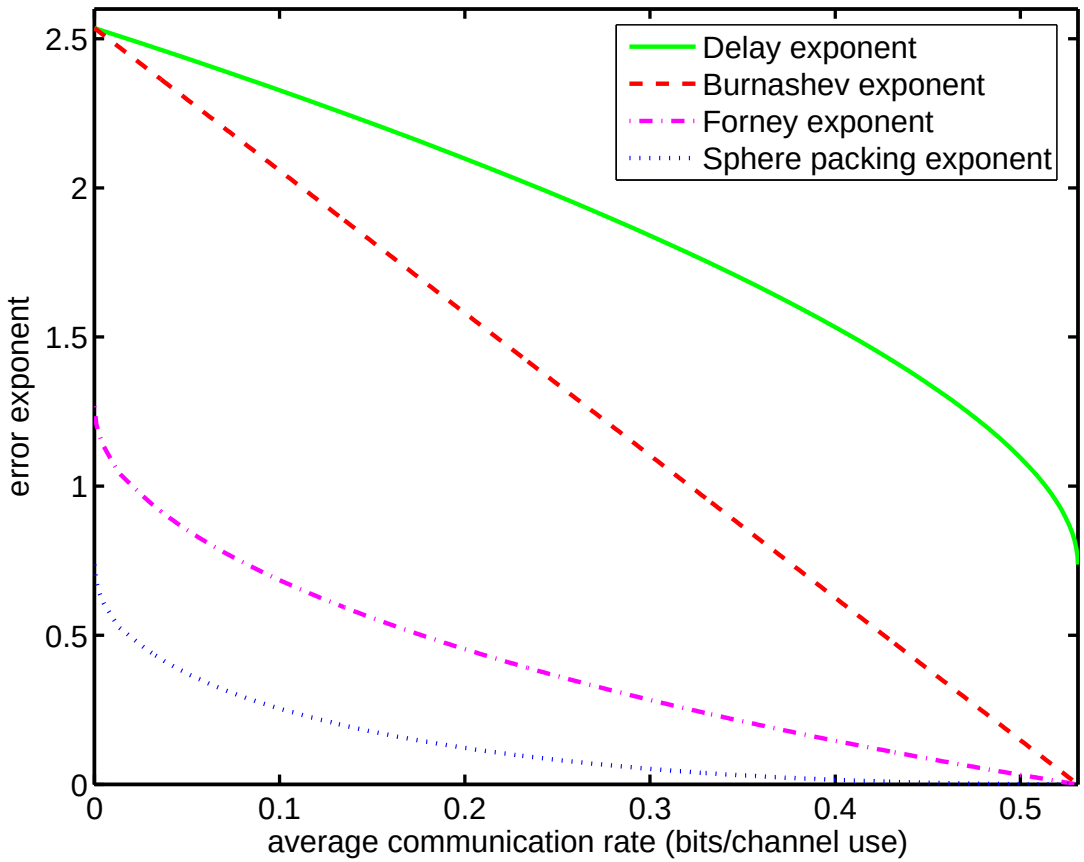
Specialize to BSC case



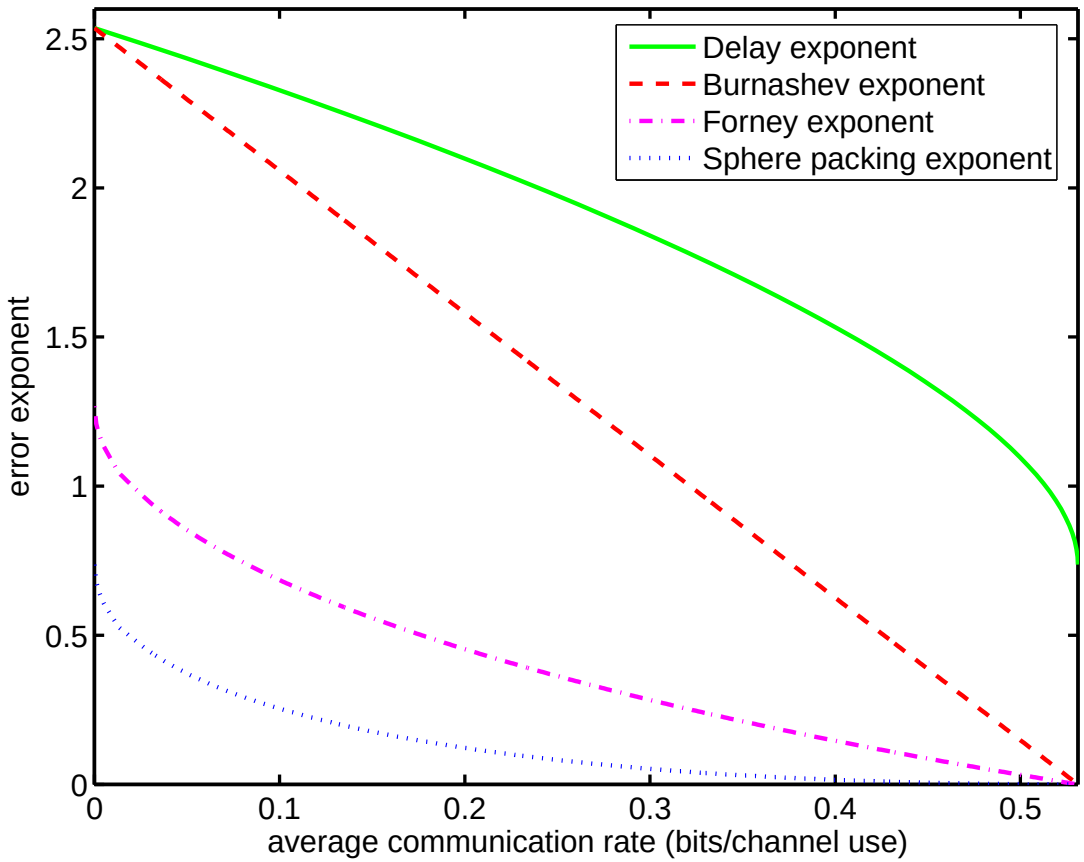


- Use all zero for NAK
- Use composition q code for data: $R < H(q) - H(p)$
- Probability of missed NAK is $2^{-ND(q_y||p)}$
- Get Δ chances: $2^{-\Delta ND(q_y||p)}$

Resulting exponents

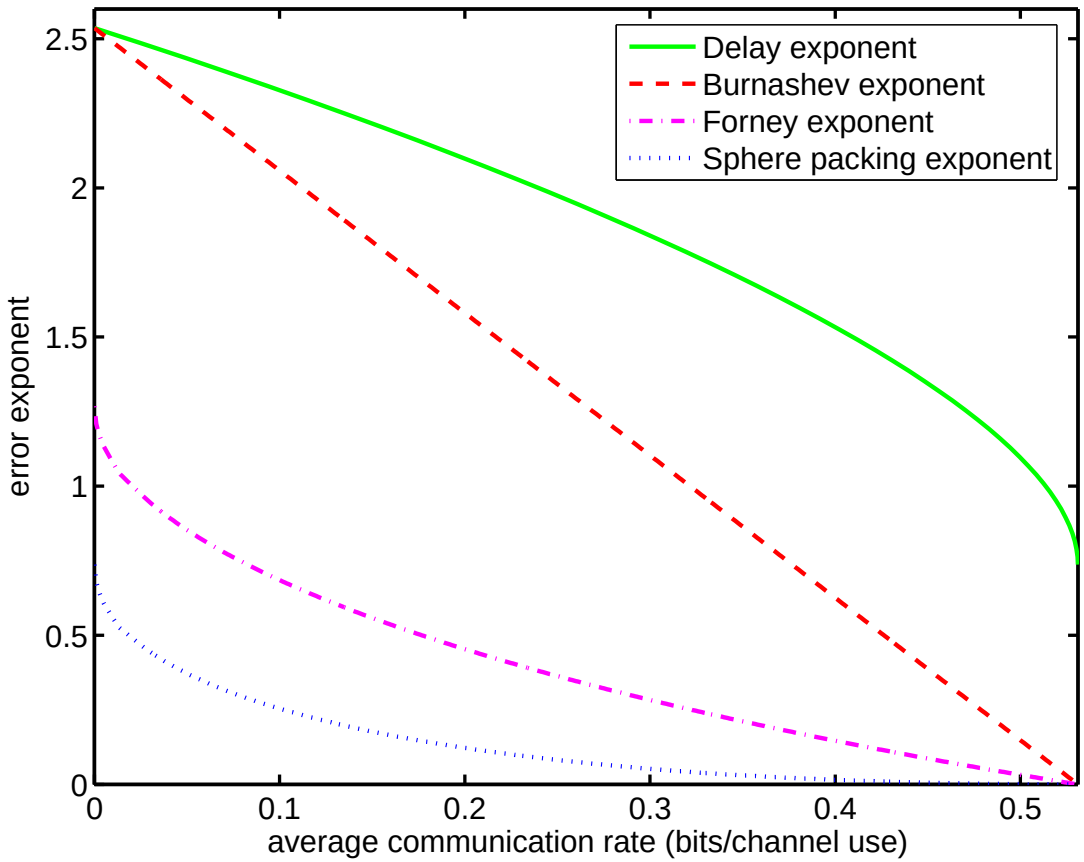


Resulting exponents



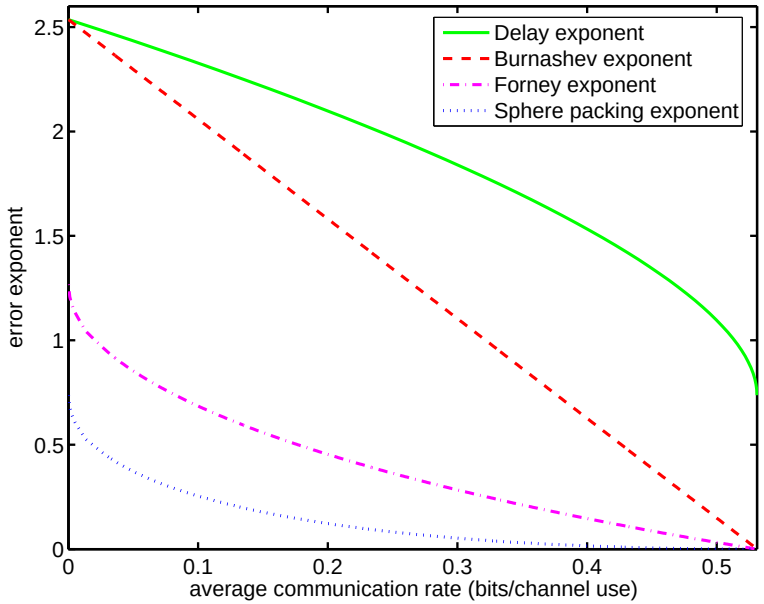
- Positive error exponent $D(\frac{1}{2}||p)$ even near capacity!

Resulting exponents



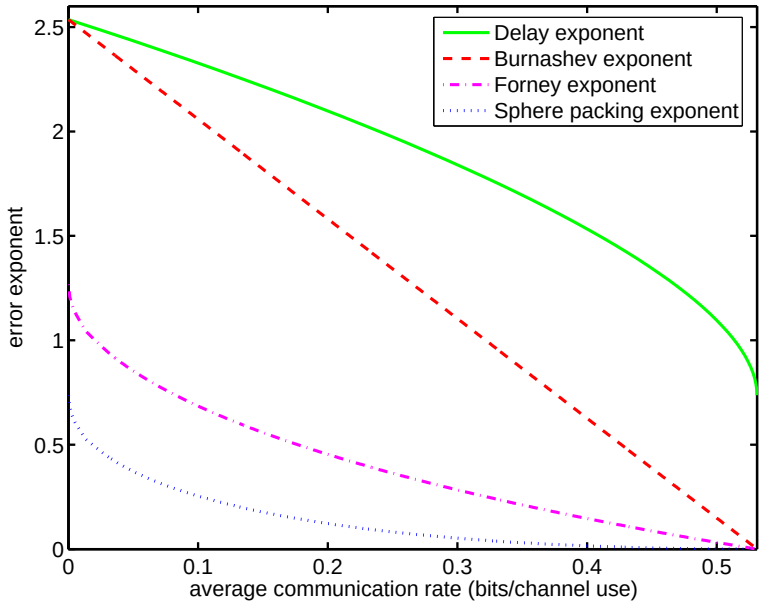
- Positive error exponent $D(\frac{1}{2}||p)$ even near capacity!
- Matches Horstein-63 exponent.

Matches the “hallucination bound”



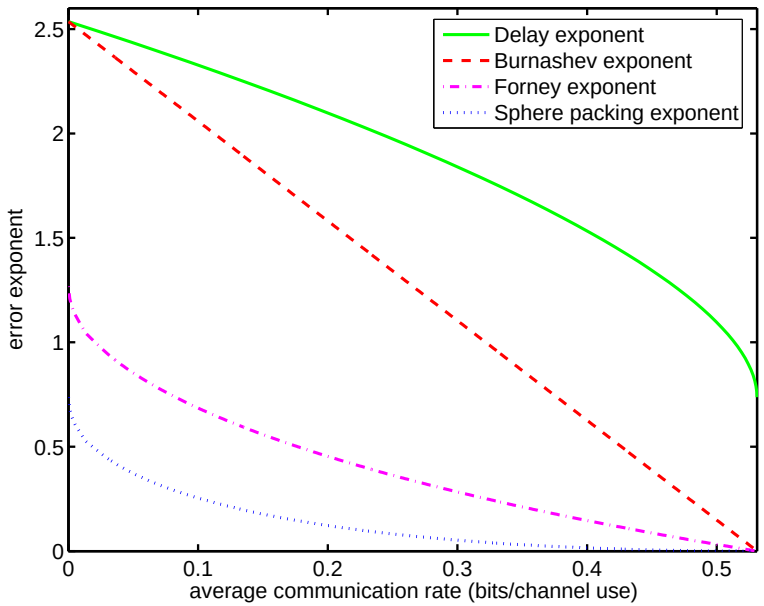
- No converse for Forney, unlike Burnashev and Sphere-packing.

Matches the “hallucination bound”



- No converse for Forney, unlike Burnashev and Sphere-packing.
- What about here?

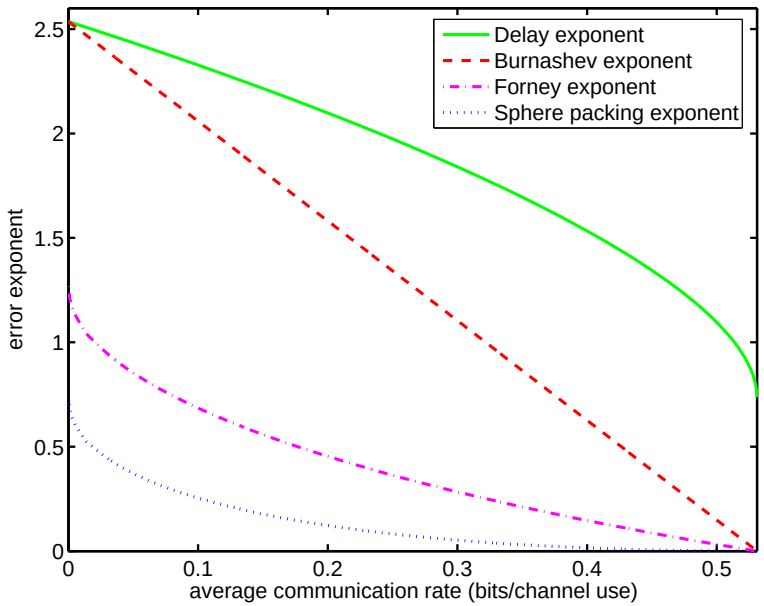
Matches the “hallucination bound”



- Decoding correctly requires a certain “typical” volume of output sequences

- No converse for Forney, unlike Burnashev and Sphere-packing.
- What about here?

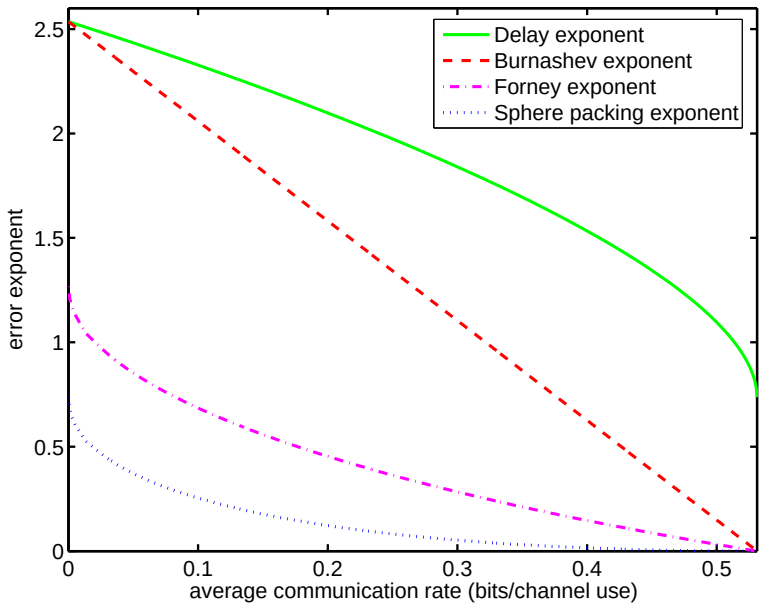
Matches the “hallucination bound”



- Decoding correctly requires a certain “typical” volume of output sequences
- If the channel forces you to hallucinate for d time-steps, you are doomed.

- No converse for Forney, unlike Burnashev and Sphere-packing.
- What about here?

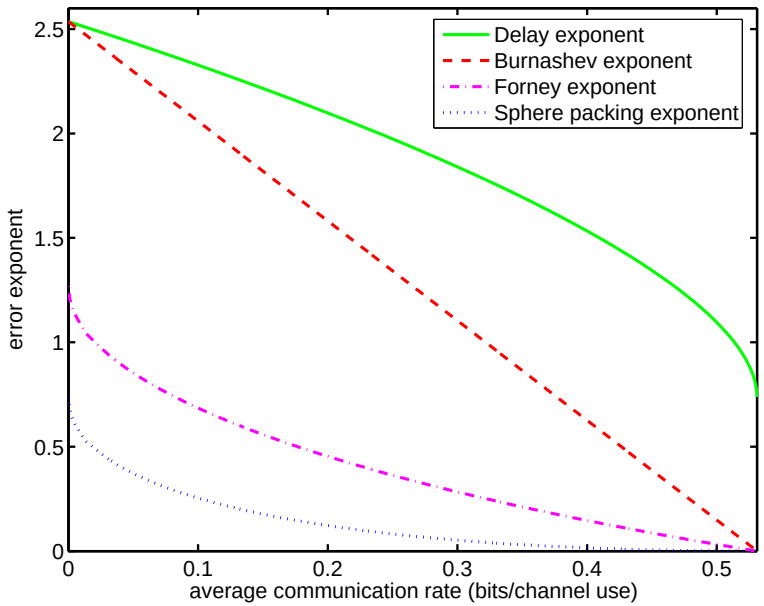
Matches the “hallucination bound”



- No converse for Forney, unlike Burnashev and Sphere-packing.
- What about here?

- Decoding correctly requires a certain “typical” volume of output sequences
- If the channel forces you to hallucinate for d time-steps, you are doomed.
- With feedback, you can choose the channel input to minimize this probability.

Matches the “hallucination bound”



- No converse for Forney, unlike Burnashev and Sphere-packing.
- What about here?

- Decoding correctly requires a certain “typical” volume of output sequences
- If the channel forces you to hallucinate for d time-steps, you are doomed.
- With feedback, you can choose the channel input to minimize this probability.
- Matches achievability.

Outline

1. Motivation and background
2. Feedback, delay, and error probability: memoryless channels
   - Idealized cases with “magical” feedback
     - ★ Block-oriented results
     - ★ Fixed-delay with hard deadlines
     - ★ Fixed-delay with soft deadlines
   - Unreliable/Noisy feedback
     - ★ Schalkwijk/Kailath scheme
     - ★ Packet-erasure channels with unreliable feedback
     - ★ Noisy channels with soft deadlines
3. Feedback and memory
4. Conclusions and open problems

Flashback and Warning: 1973 (Bob Lucky)

"Feedback communications was an area of intense activity in 1968...A number of authors had shown constructive, even simple, schemes using noiseless feedback to achieve Shannon-like behavior...The situation in 1973 is dramatically different...The subject itself seems to be a burned out case...

In extending the simple noiseless feedback model to allow for more realistic situations, such as noisy feedback channels, bandlimited channels, and peak power constraints, theorists discovered a certain "brittleness" or sensitivity in their previous results..."

Flashback and Warning: 1973 (Bob Lucky)

"Feedback communications was an area of intense activity in 1968...A number of authors had shown constructive, even simple, schemes using noiseless feedback to achieve Shannon-like behavior...The situation in 1973 is dramatically different...The subject itself seems to be a burned out case...

In extending the simple noiseless feedback model to allow for more realistic situations, such as noisy feedback channels, bandlimited channels, and peak power constraints, theorists discovered a certain "brittleness" or sensitivity in their previous results..."

The goal: show "robustness" of gains due to feedback.

What goes wrong with the S-K scheme?

- Wyner-68 showed that the double-exponential depends crucially on the *average* nature of the power-constraint.

What goes wrong with the S-K scheme?

- Wyner-68 showed that the double-exponential depends crucially on the *average* nature of the power-constraint.
- Passive noisy feedback
  - Intuition
    - ★ Linear in the feedback noise.
    - ★ Shows up in the final estimation error as a term that does not vanish with n.
    - ★ "Smudges out" the message, regardless of rate.

What goes wrong with the S-K scheme?

- Wyner-68 showed that the double-exponential depends crucially on the *average* nature of the power-constraint.
- Passive noisy feedback
  - Intuition
    - ★ Linear in the feedback noise.
    - ★ Shows up in the final estimation error as a term that does not vanish with n.
    - ★ "Smudges out" the message, regardless of rate.
  - Proof: (Kim, Lapidoth, Weissman '06)
    - ★ Linear strategies must achieve double-exponentials.
    - ★ Exponential chance the noisy feedback behaves like it is disconnected.

What goes wrong with the S-K scheme?

- Wyner-68 showed that the double-exponential depends crucially on the *average* nature of the power-constraint.
- Passive noisy feedback
  - Intuition
    - ★ Linear in the feedback noise.
    - ★ Shows up in the final estimation error as a term that does not vanish with n.
    - ★ "Smudges out" the message, regardless of rate.
  - Proof: (Kim, Lapidoth, Weissman '06)
    - ★ Linear strategies must achieve double-exponentials.
    - ★ Exponential chance the noisy feedback behaves like it is disconnected.
- Active feedback with finite-support noise (Weissman, Martins '06)
  - Quantize feedback and send with zero-error.
  - Quantization noise known at the decoder.

What goes wrong with the S-K scheme?

- Wyner-68 showed that the double-exponential depends crucially on the *average* nature of the power-constraint.
- Passive noisy feedback
  - Intuition
    - ★ Linear in the feedback noise.
    - ★ Shows up in the final estimation error as a term that does not vanish with n.
    - ★ "Smudges out" the message, regardless of rate.
  - Proof: (Kim, Lapidoth, Weissman '06)
    - ★ Linear strategies must achieve double-exponentials.
    - ★ Exponential chance the noisy feedback behaves like it is disconnected.
- Active feedback with finite-support noise (Weissman, Martins '06)
  - Quantize feedback and send with zero-error.
  - Quantization noise known at the decoder.
  - Linearity implies effect can be subtracted at the end.

What goes wrong with the S-K scheme?

- Wyner-68 showed that the double-exponential depends crucially on the *average* nature of the power-constraint.
- Passive noisy feedback
  - Intuition
    - ★ Linear in the feedback noise.
    - ★ Shows up in the final estimation error as a term that does not vanish with n.
    - ★ "Smudges out" the message, regardless of rate.
  - Proof: (Kim, Lapidoth, Weissman '06)
    - ★ Linear strategies must achieve double-exponentials.
    - ★ Exponential chance the noisy feedback behaves like it is disconnected.
- Active feedback with finite-support noise (Weissman, Martins '06)
  - Quantize feedback and send with zero-error.
  - Quantization noise known at the decoder.
  - Linearity implies effect can be subtracted at the end.
  - Increases average power slightly.

Outline

1. Motivation and background
2. Feedback, delay, and error probability: memoryless channels
   - Idealized cases with “magical” feedback
     - ★ Block-oriented results
     - ★ Fixed-delay with hard deadlines
     - ★ Fixed-delay with soft deadlines
   - Unreliable/Noisy feedback
     - ★ Schalkwijk/Kailath scheme
     - ★ Packet-erasure channels with unreliable feedback
     - ★ Noisy channels with soft deadlines
3. Feedback and memory
4. Conclusions and open problems

Packet-erasure channel

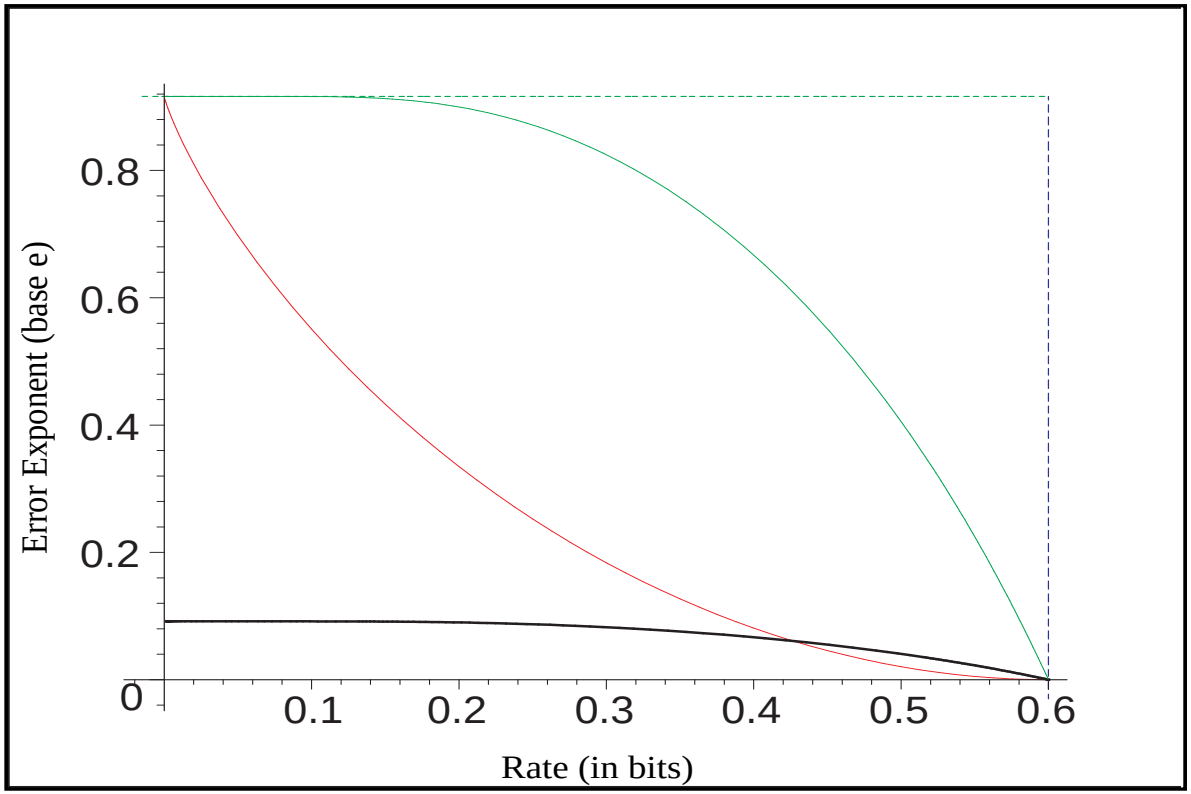
- Forward packets either delivered or erased
- Feedback limitations:
  - Delayed but noiseless
  - Over a separate low-rate packet-erasure channel
  - Over a shared packet-erasure channel

Example: short round-trip-time k

First approach: treat as k parallel unit-delay channels

Example: short round-trip-time k

First approach: treat as k parallel unit-delay channels



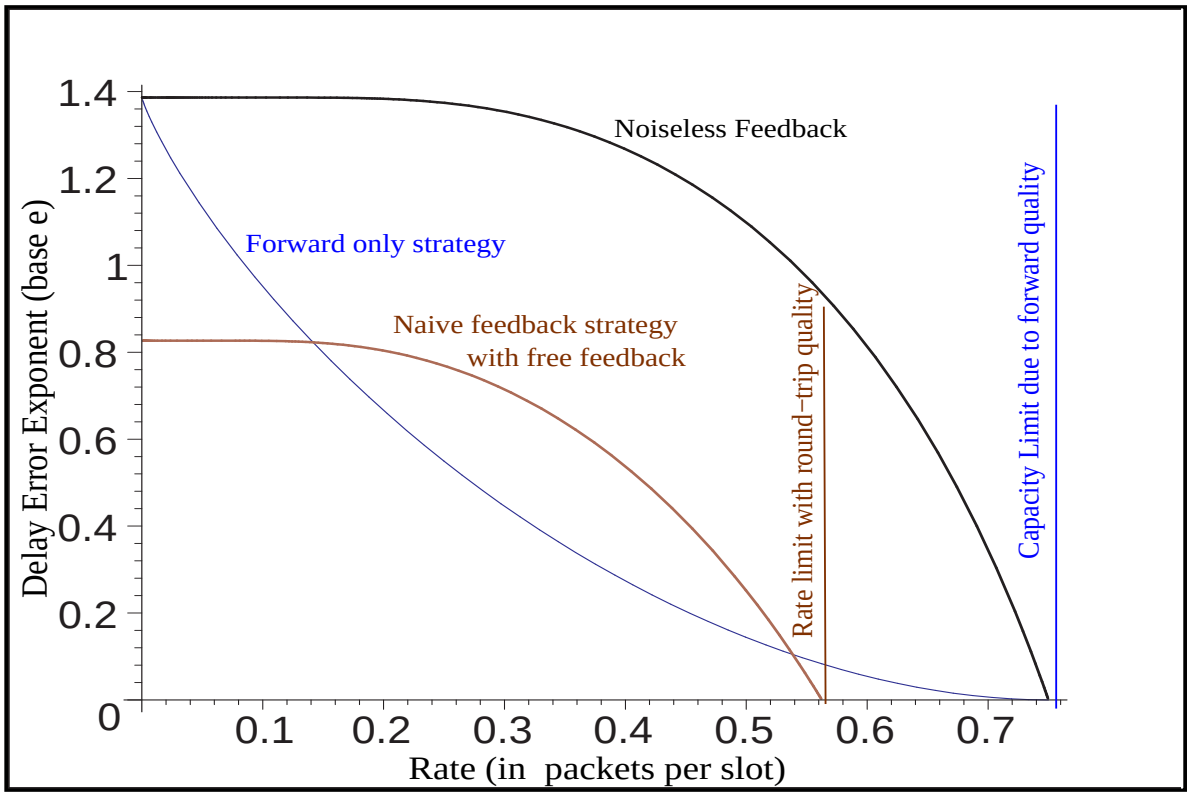
Serious penalty for "dividing the channel"

Example: unreliable feedback

First approach: consider a lost feedback as negative feedback

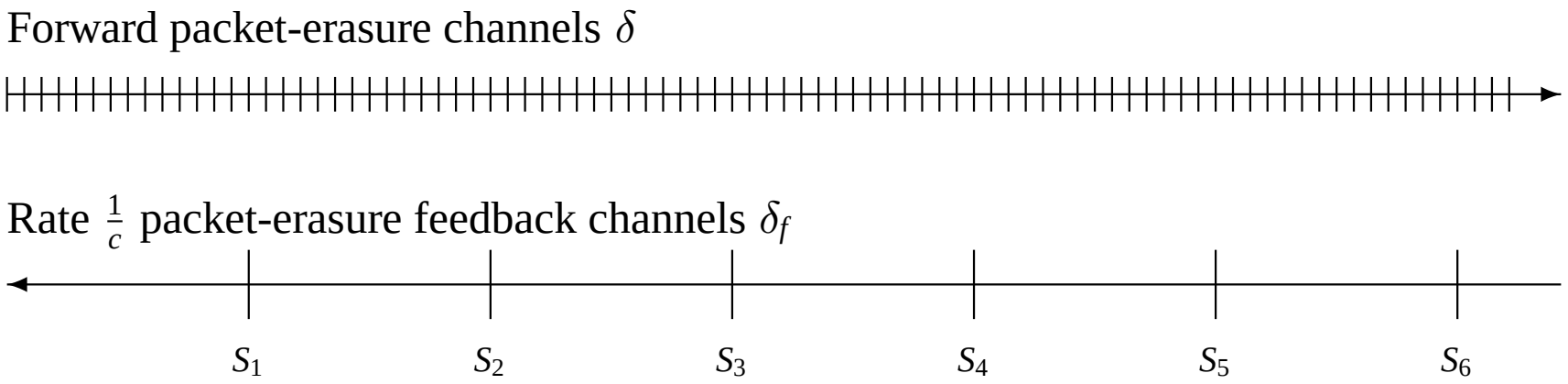
Example: unreliable feedback

First approach: consider a lost feedback as negative feedback



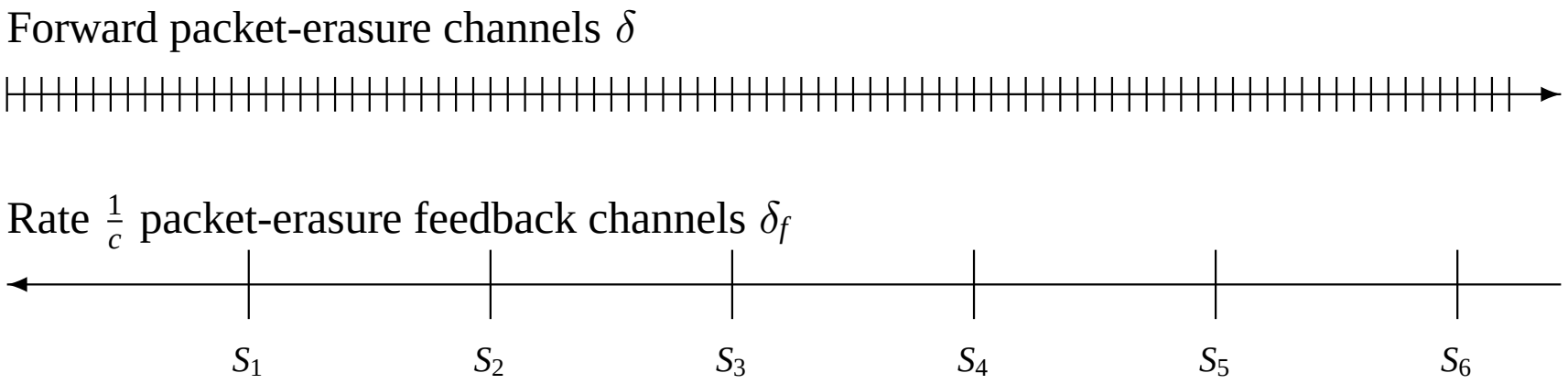
Capacity penalty for treating both losses symmetrically.

Unreliable feedback picture



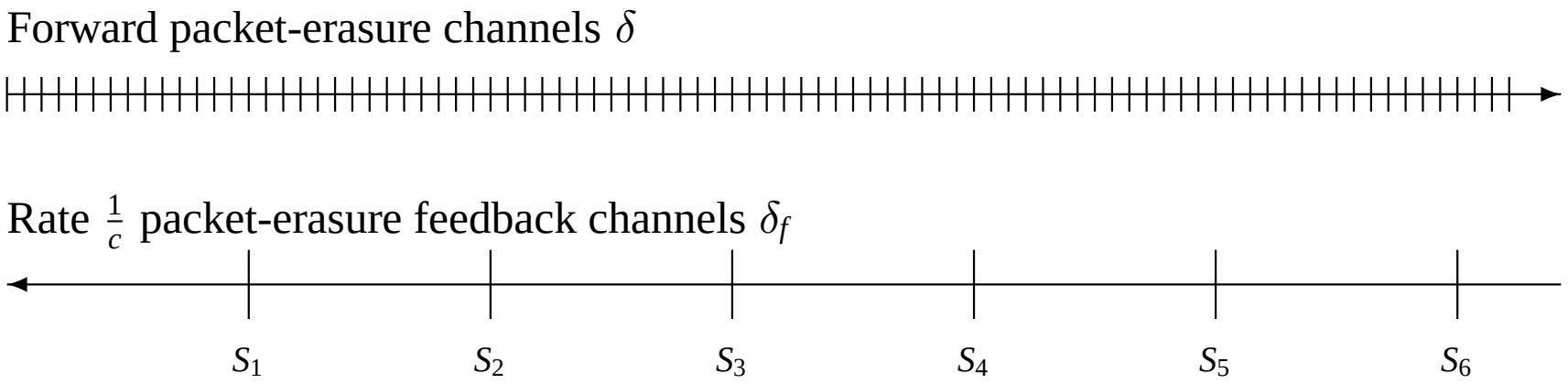
- *Both* channels drop packets randomly.

Unreliable feedback picture



- *Both* channels drop packets randomly.
- Assume packets large enough so that "header bits" are free.

Unreliable feedback picture



- *Both* channels drop packets randomly.
- Assume packets large enough so that "header bits" are free.
- Assume target latency $d \gg c, k$.

Adapting incremental redundancy hybrid ARQ

- Gentler retransmission:

Adapting incremental redundancy hybrid ARQ

- Gentler retransmission:
  1. Group packets into blocks of size nR. ($c \ll n \ll d$)

Adapting incremental redundancy hybrid ARQ

- Gentler retransmission:
  1. Group packets into blocks of size nR. ($c \ll n \ll d$)
  2. Hold blocks in a FIFO queue

Adapting incremental redundancy hybrid ARQ

- Gentler retransmission:
  1. Group packets into blocks of size nR. ($c \ll n \ll d$)
  2. Hold blocks in a FIFO queue
  3. Service using an ∞-length random codebook. (rateless code)

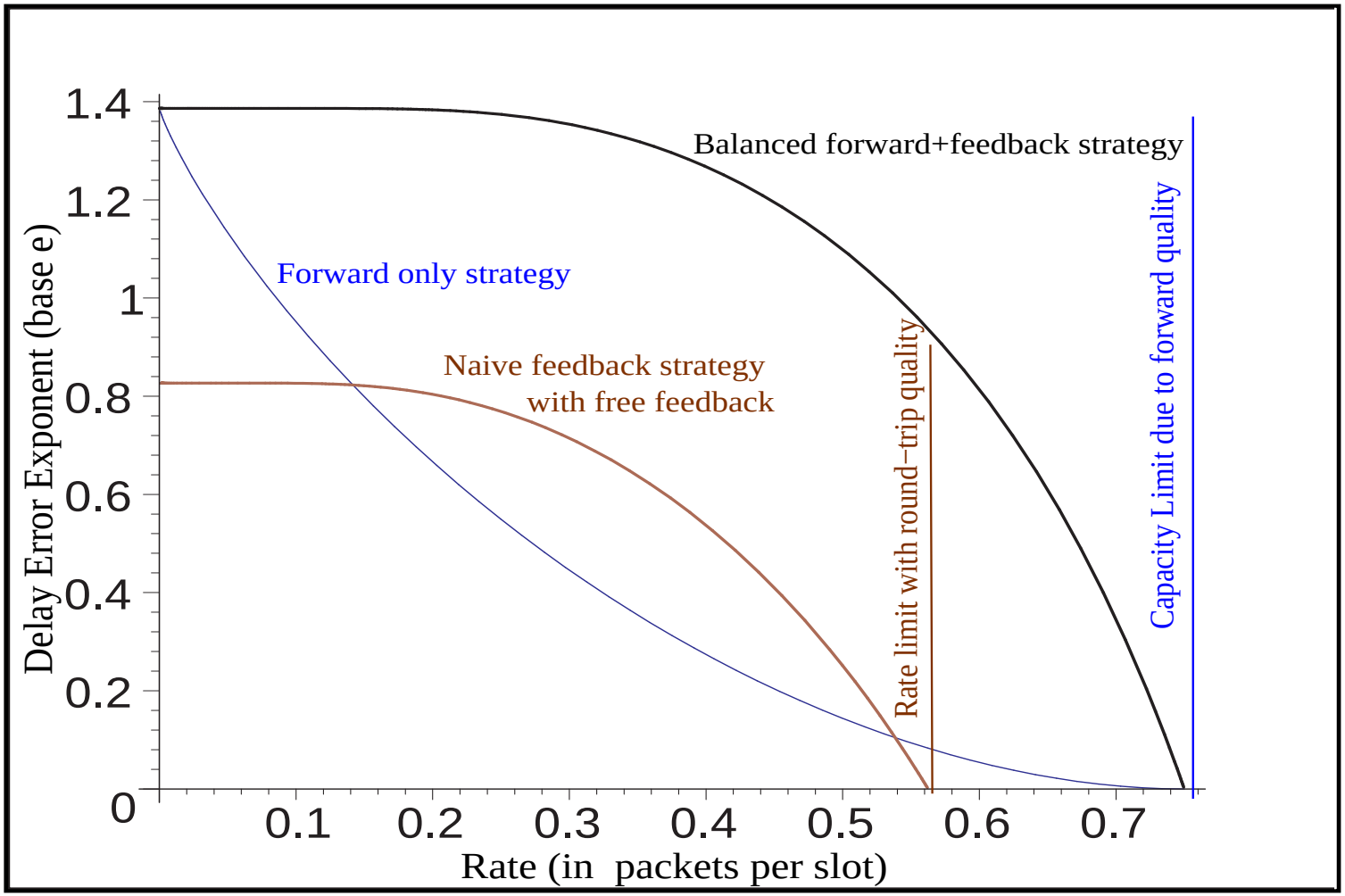
Adapting incremental redundancy hybrid ARQ

- Gentler retransmission:
  1. Group packets into blocks of size nR. ($c \ll n \ll d$)
  2. Hold blocks in a FIFO queue
  3. Service using an ∞-length random codebook. (rateless code)
  4. Feedback "ACK" when we can decode. (repetition code)

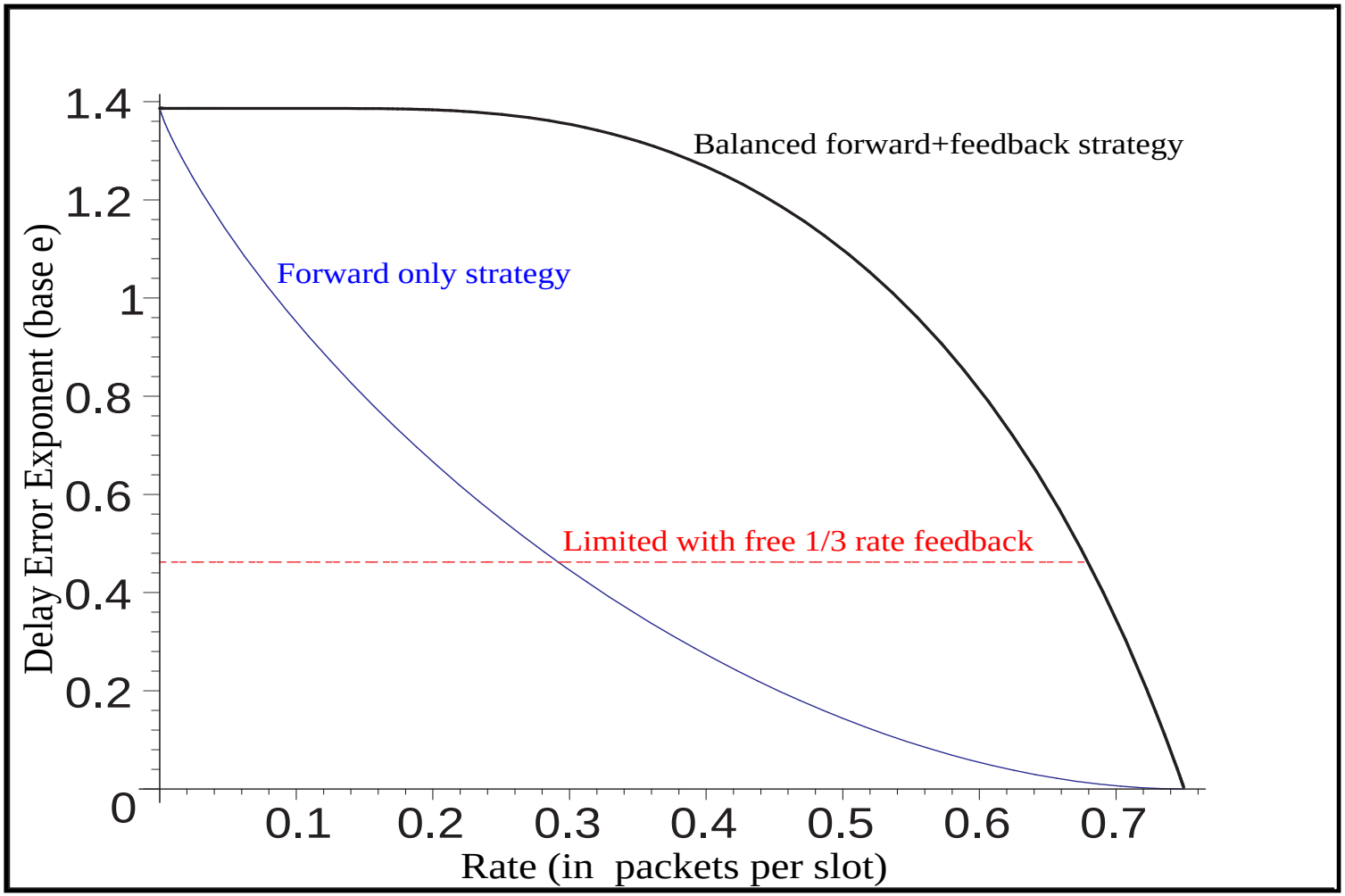
Adapting incremental redundancy hybrid ARQ

- Gentler retransmission:
  1. Group packets into blocks of size nR. ($c \ll n \ll d$)
  2. Hold blocks in a FIFO queue
  3. Service using an ∞-length random codebook. (rateless code)
  4. Feedback "ACK" when we can decode. (repetition code)
- Add one header bit to forward packets (Massey)
  - 0: first message block
  - 1: second message block
  - 0: third message block
  - $\vdots$

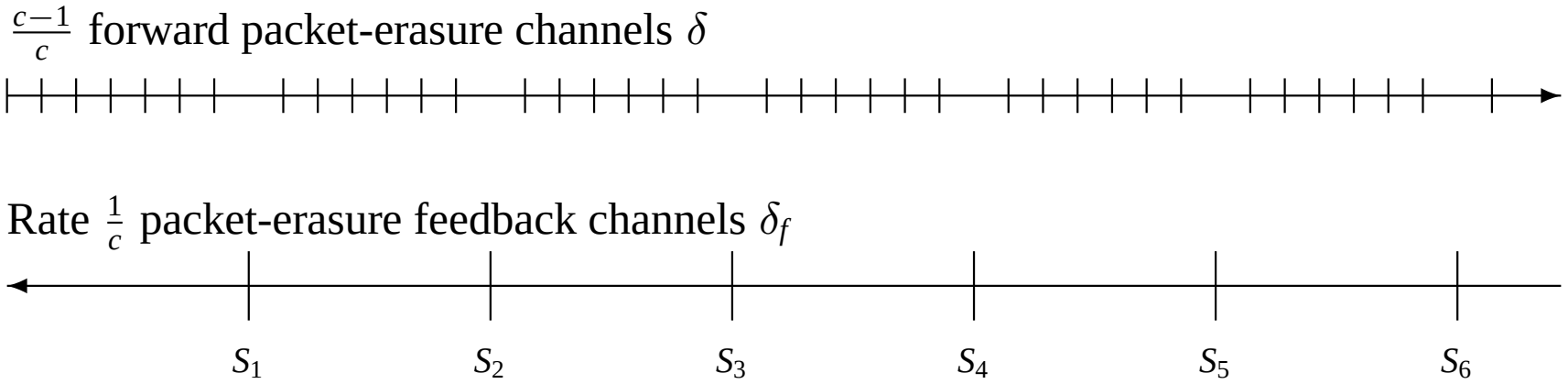
Performance with unreliable feedback channel



Performance with unreliable feedback channel

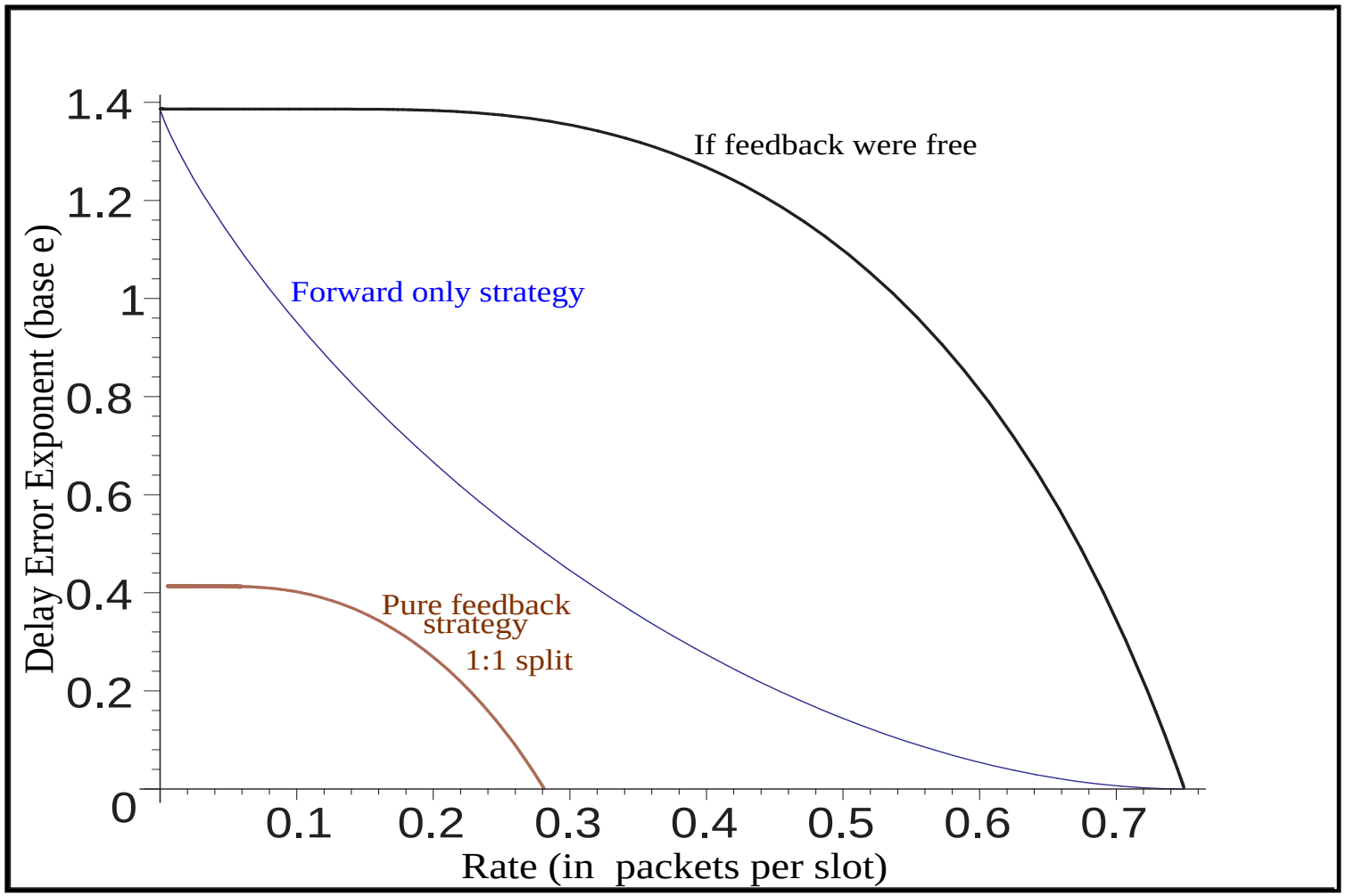


Shared channel picture

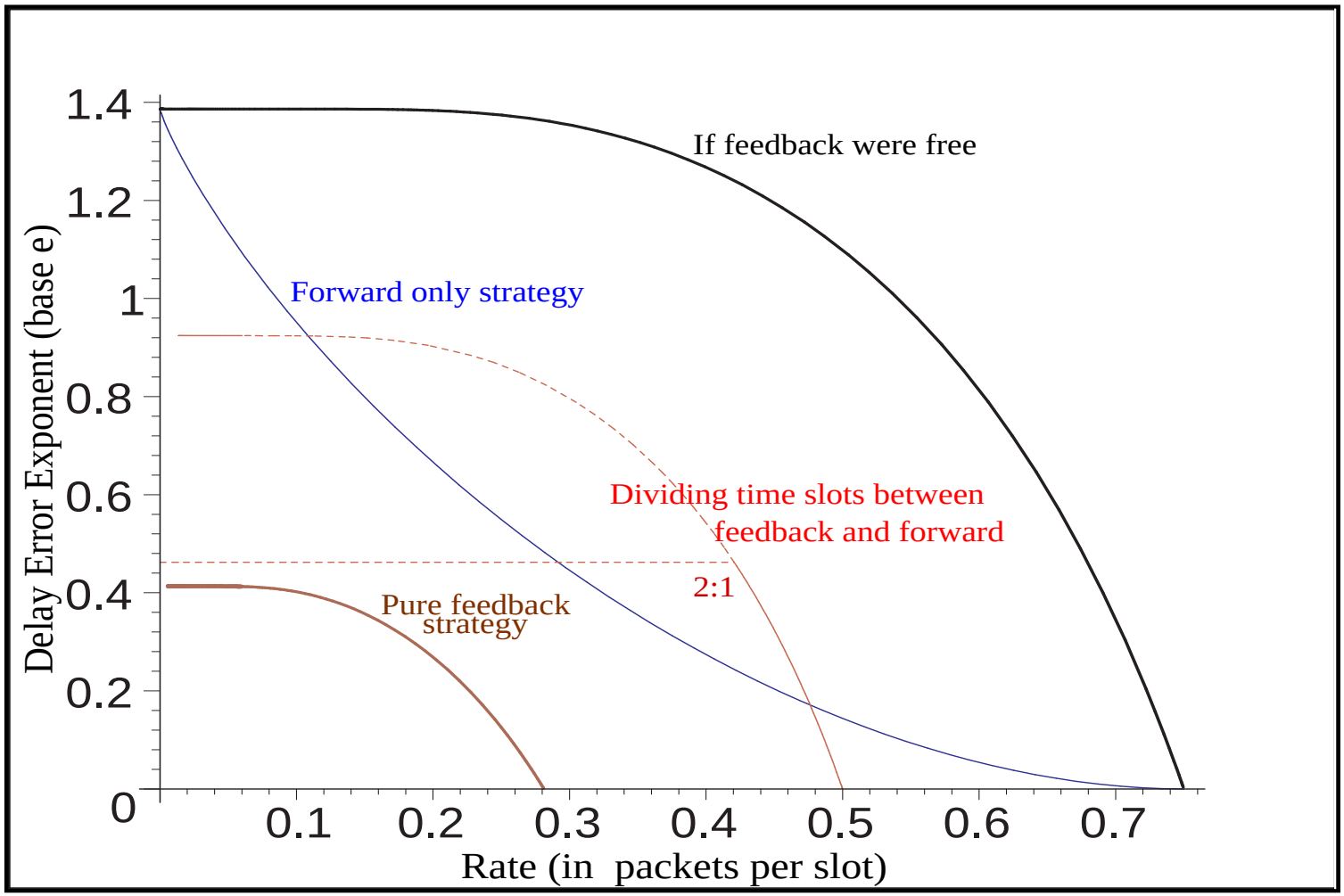


- *Both* users share a single physical channel
- Half-duplex constraint: only one can use at a time
- Assume we must schedule them in advance
- Same erasure probability in both directions

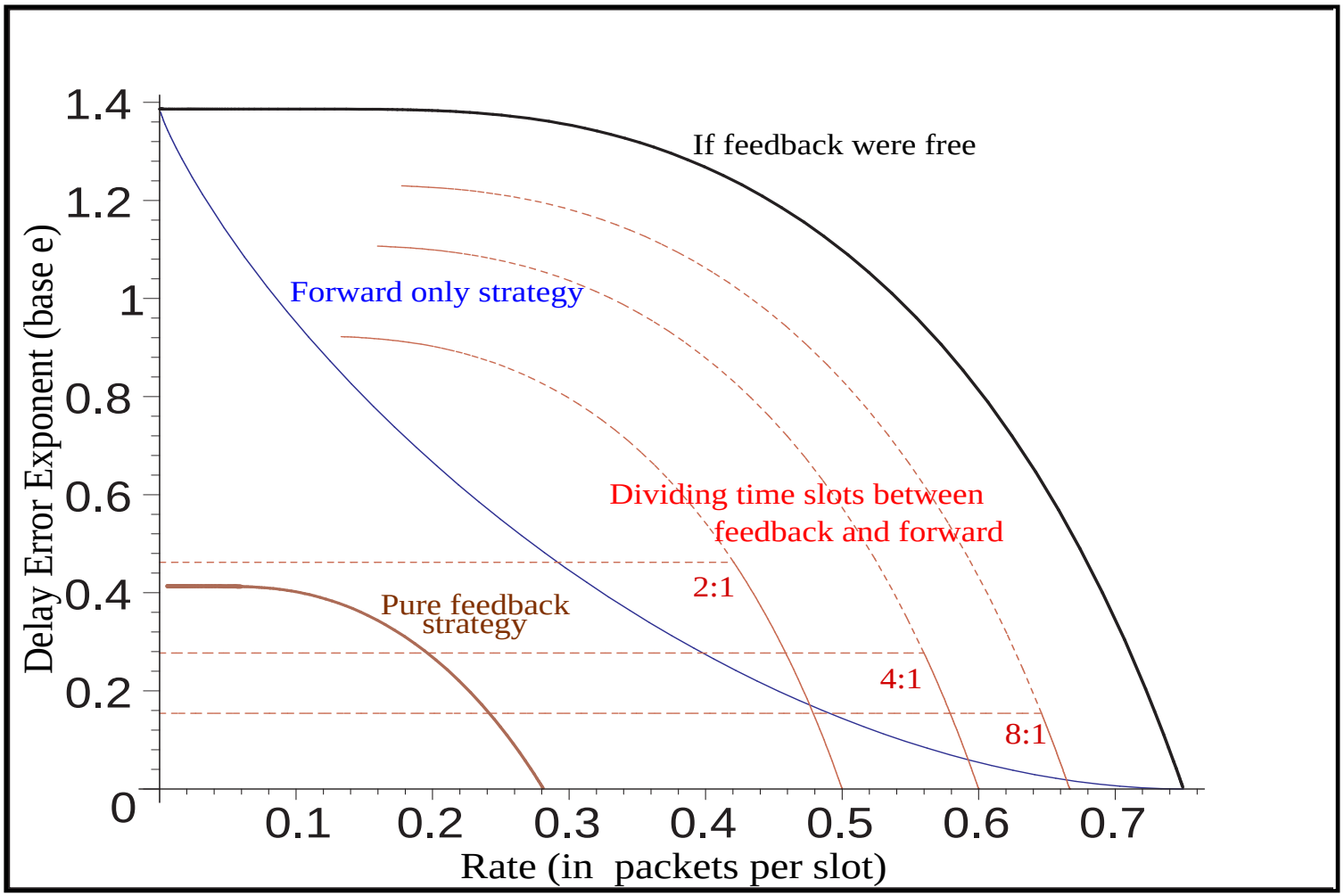
Shared unreliable channel used for feedback



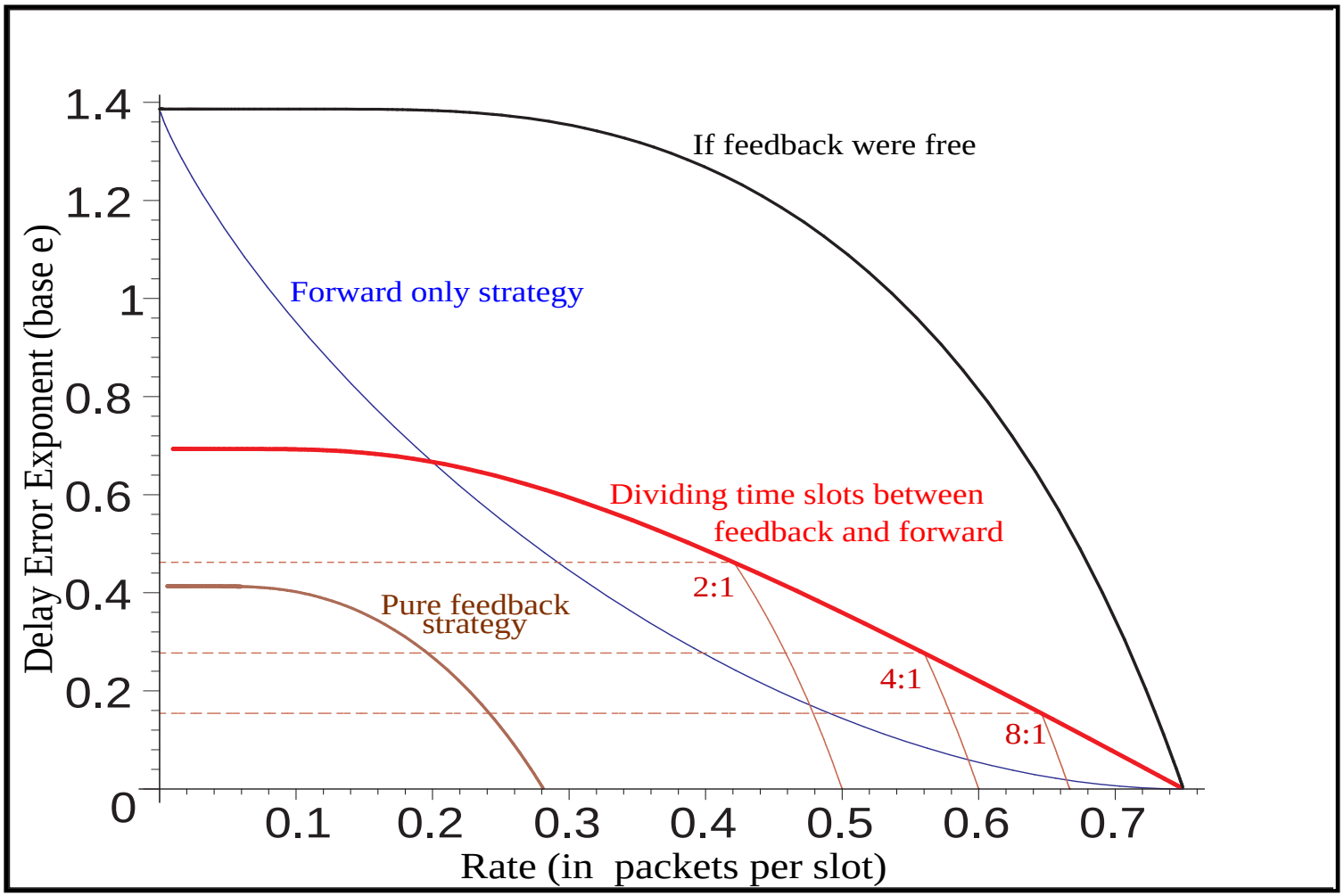
Shared unreliable channel used for feedback



Shared unreliable channel used for feedback



Shared unreliable channel used for feedback



Like focusing bound using $E_0'(\rho) = (-\ln(\delta_f)^{-1} + E_0^{-1}(\rho))^{-1}$

Summary of results:

- Perfect feedback performance as long as $\frac{-1}{c} \log \delta_f$ is high enough.

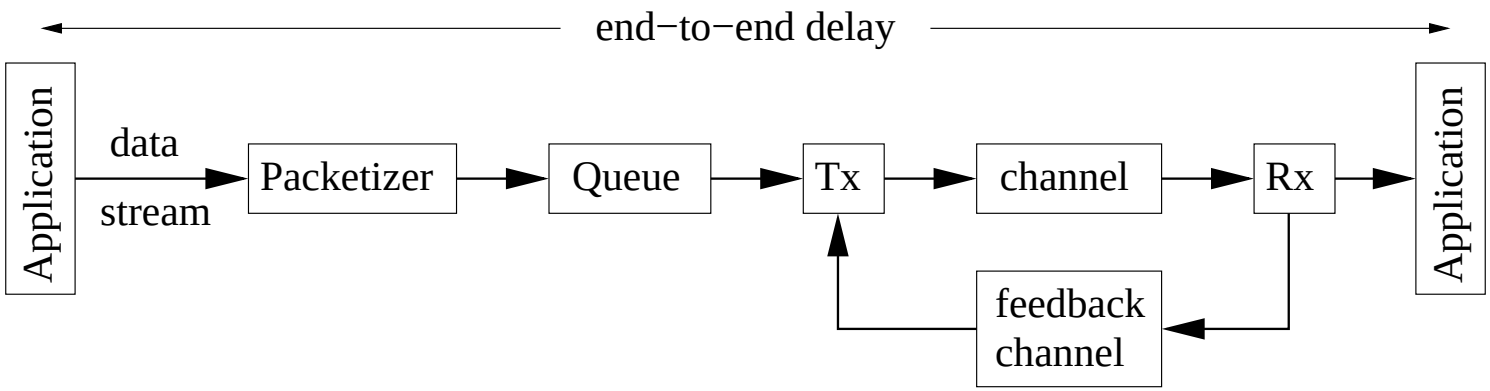
Summary of results:

- Perfect feedback performance as long as $\frac{-1}{c} \log \delta_f$ is high enough.
- It is worth allocating slots for feedback *even if this means taking them away from the forward channel.*

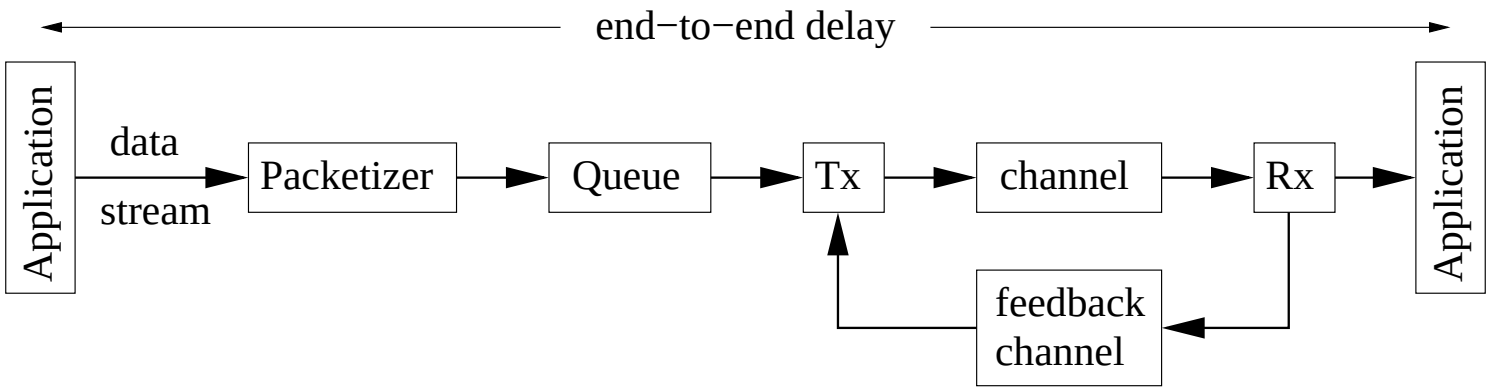
Outline

1. Motivation and background
2. Feedback, delay, and error probability: memoryless channels
   - Idealized cases with “magical” feedback
     - ★ Block-oriented results
     - ★ Fixed-delay with hard deadlines
     - ★ Fixed-delay with soft deadlines
   - Unreliable/Noisy feedback
     - ★ Schalkwijk/Kailath scheme
     - ★ Packet-erasure channels with unreliable feedback
     - ★ Noisy channels with soft deadlines
3. Feedback and memory
4. Conclusions and open problems

Noisy feedback: two distinct issues

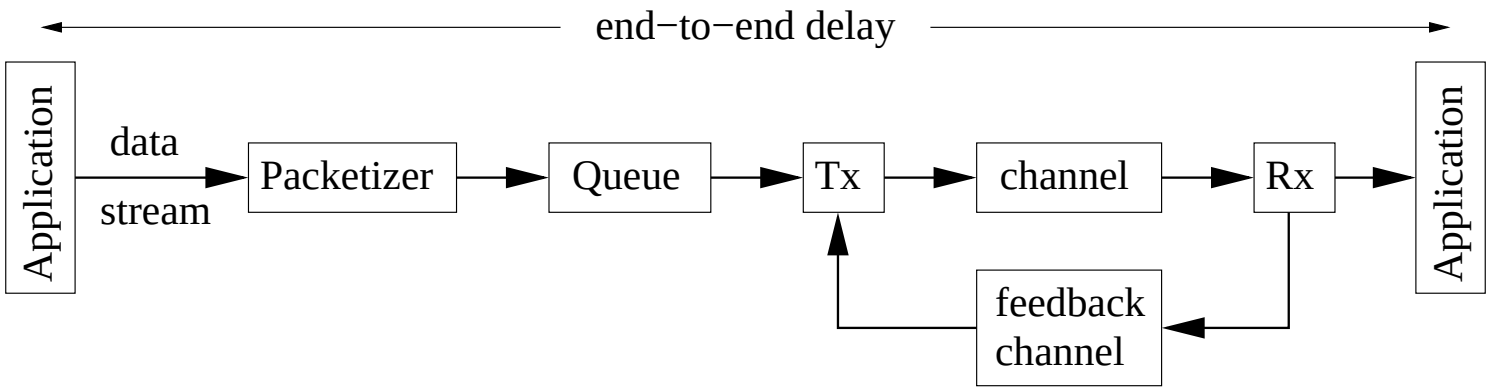


Noisy feedback: two distinct issues



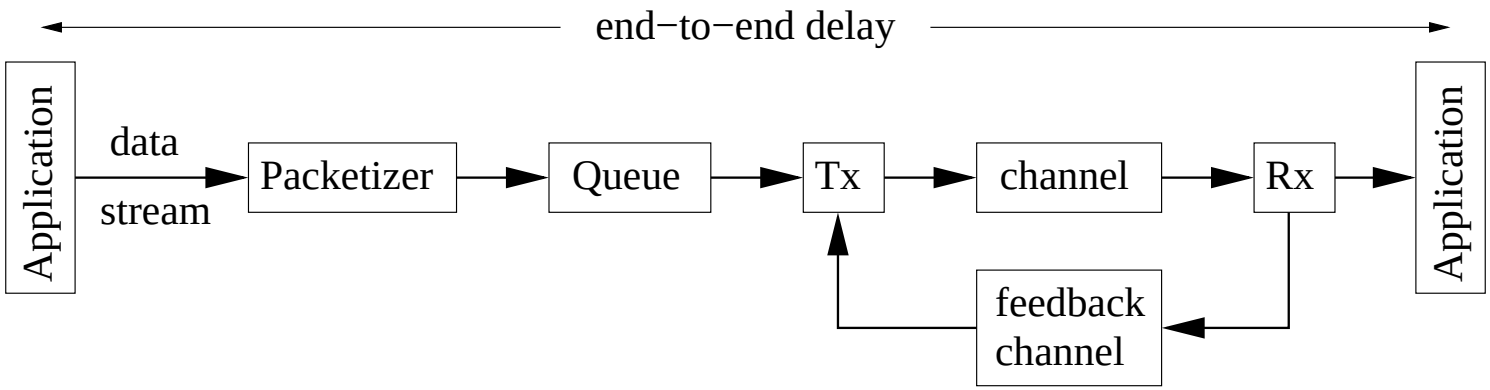
- Retransmission control: maintaining synchronization

Noisy feedback: two distinct issues



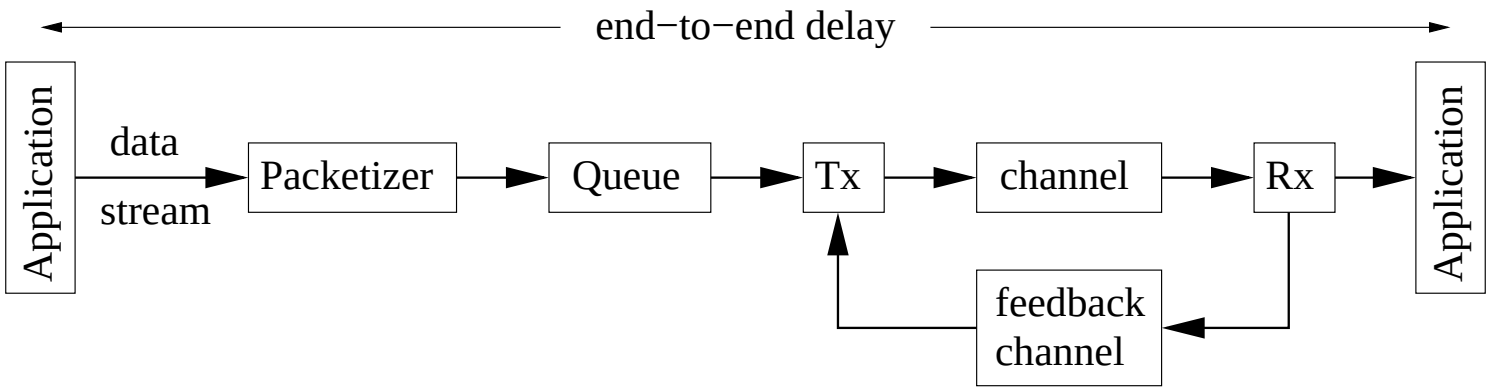
- Retransmission control: maintaining synchronization
  - Can model the state of the receiver as a random walk with drift.

Noisy feedback: two distinct issues



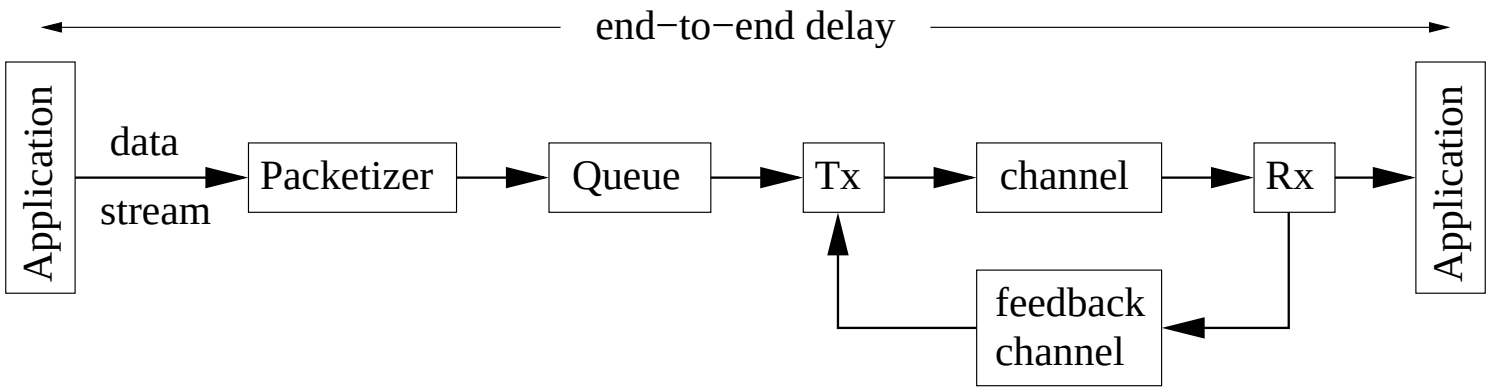
- Retransmission control: maintaining synchronization
  - Can model the state of the receiver as a random walk with drift.
  - Unstable process, but it can be tracked using an *anytime* code.

Noisy feedback: two distinct issues



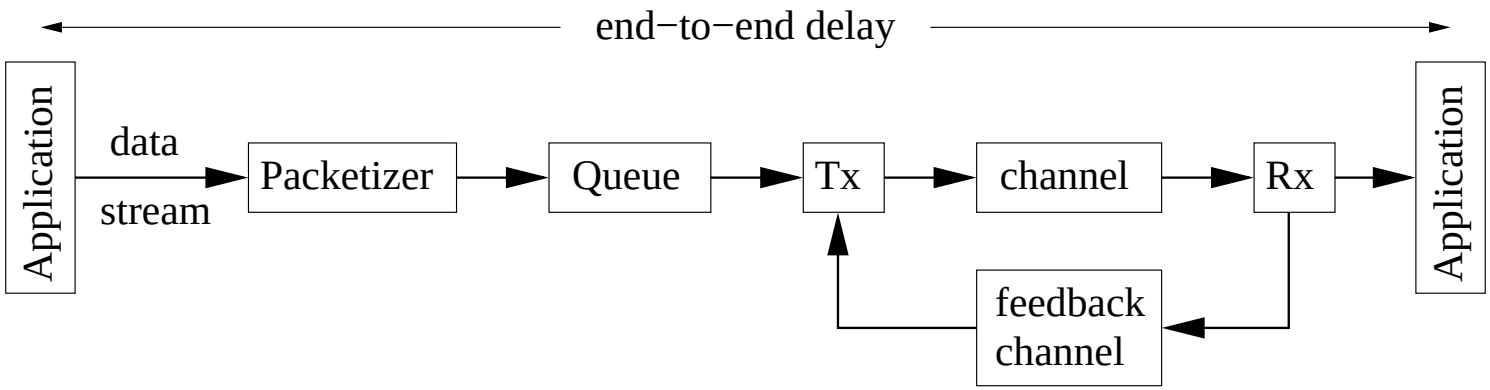
- Retransmission control: maintaining synchronization
  - Can model the state of the receiver as a random walk with drift.
  - Unstable process, but it can be tracked using an *anytime* code.
  - Since retransmissions are rare, can afford extra latency to allow state estimates to converge.

Noisy feedback: two distinct issues



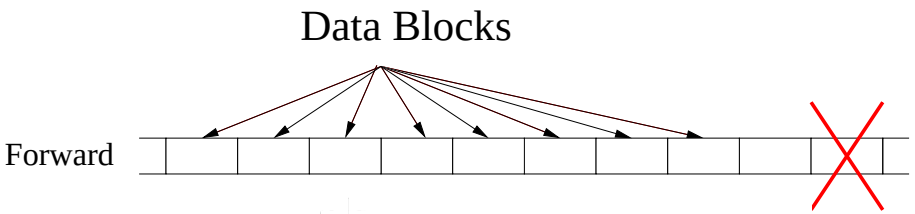
- Retransmission control: maintaining synchronization
  - Can model the state of the receiver as a random walk with drift.
  - Unstable process, but it can be tracked using an *anytime* code.
  - Since retransmissions are rare, can afford extra latency to allow state estimates to converge.
  - Synchronization rate is less than 1 bit per packet.

Noisy feedback: two distinct issues



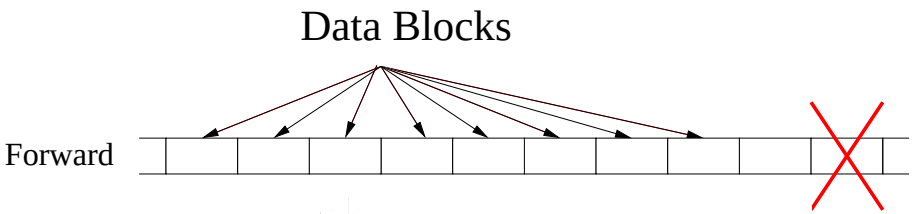
- Retransmission control: maintaining synchronization
  - Can model the state of the receiver as a random walk with drift.
  - Unstable process, but it can be tracked using an *anytime* code.
  - Since retransmissions are rare, can afford extra latency to allow state estimates to converge.
  - Synchronization rate is less than 1 bit per packet.
- How to NAK?

What is needed to NAK?



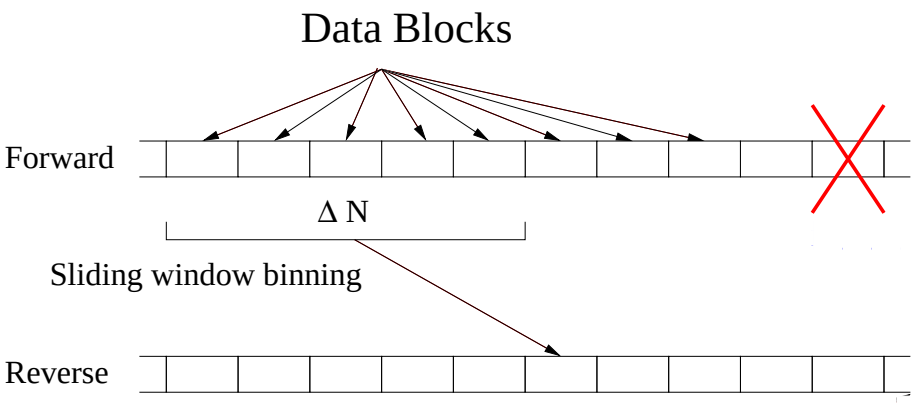
- Must know if any errors in the sliding window.

What is needed to NAK?



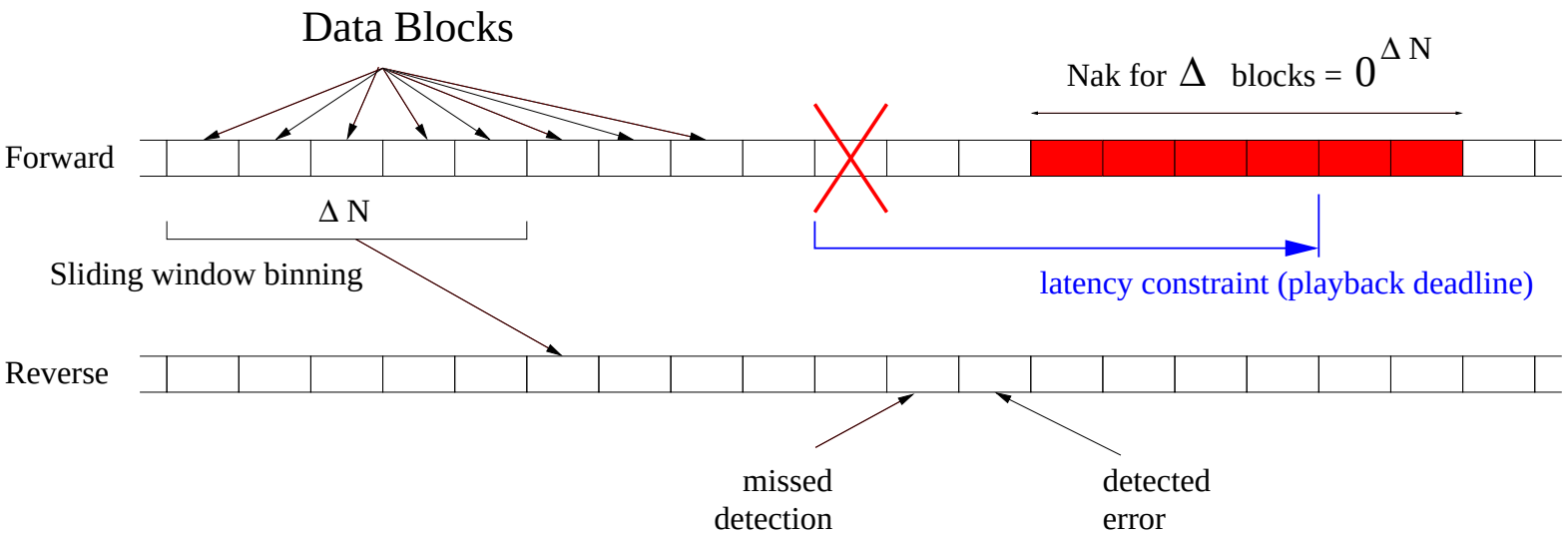
- Must know if any errors in the sliding window.
- Identification problem since encoder knows what it wants to hear.

What is needed to NAK?



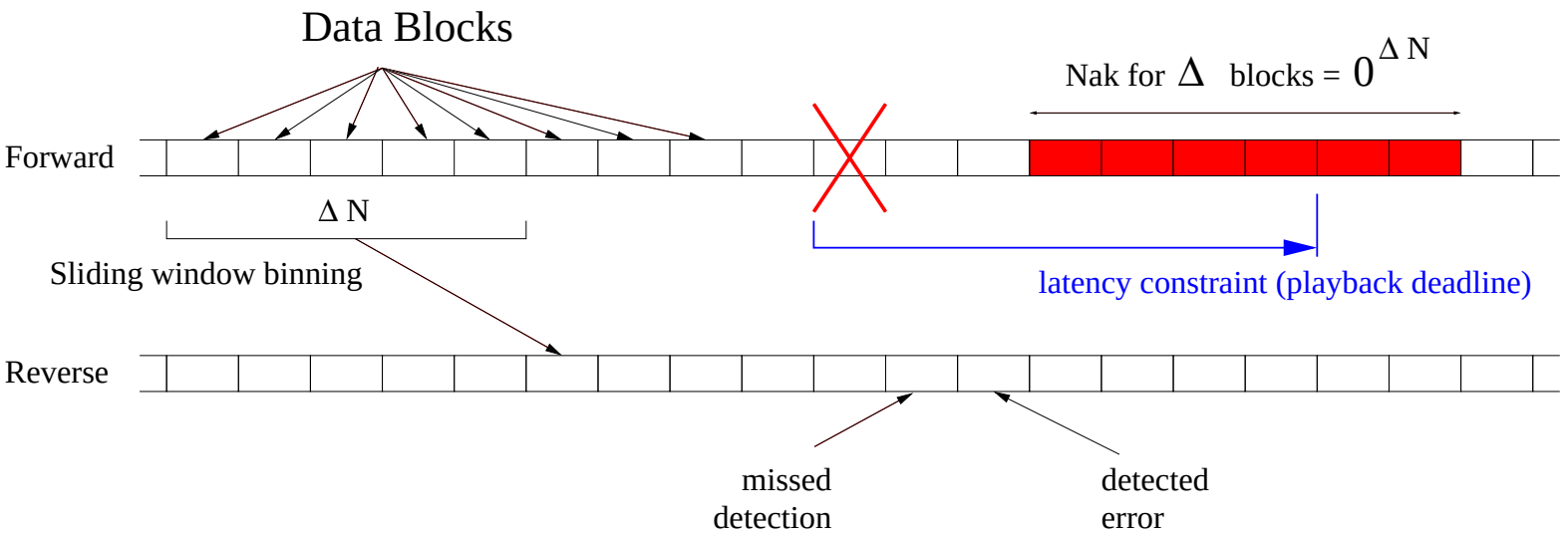
- Must know if any errors in the sliding window.
- Identification problem since encoder knows what it wants to hear.
- Use a random hash of entire sliding window.

What is needed to NAK?



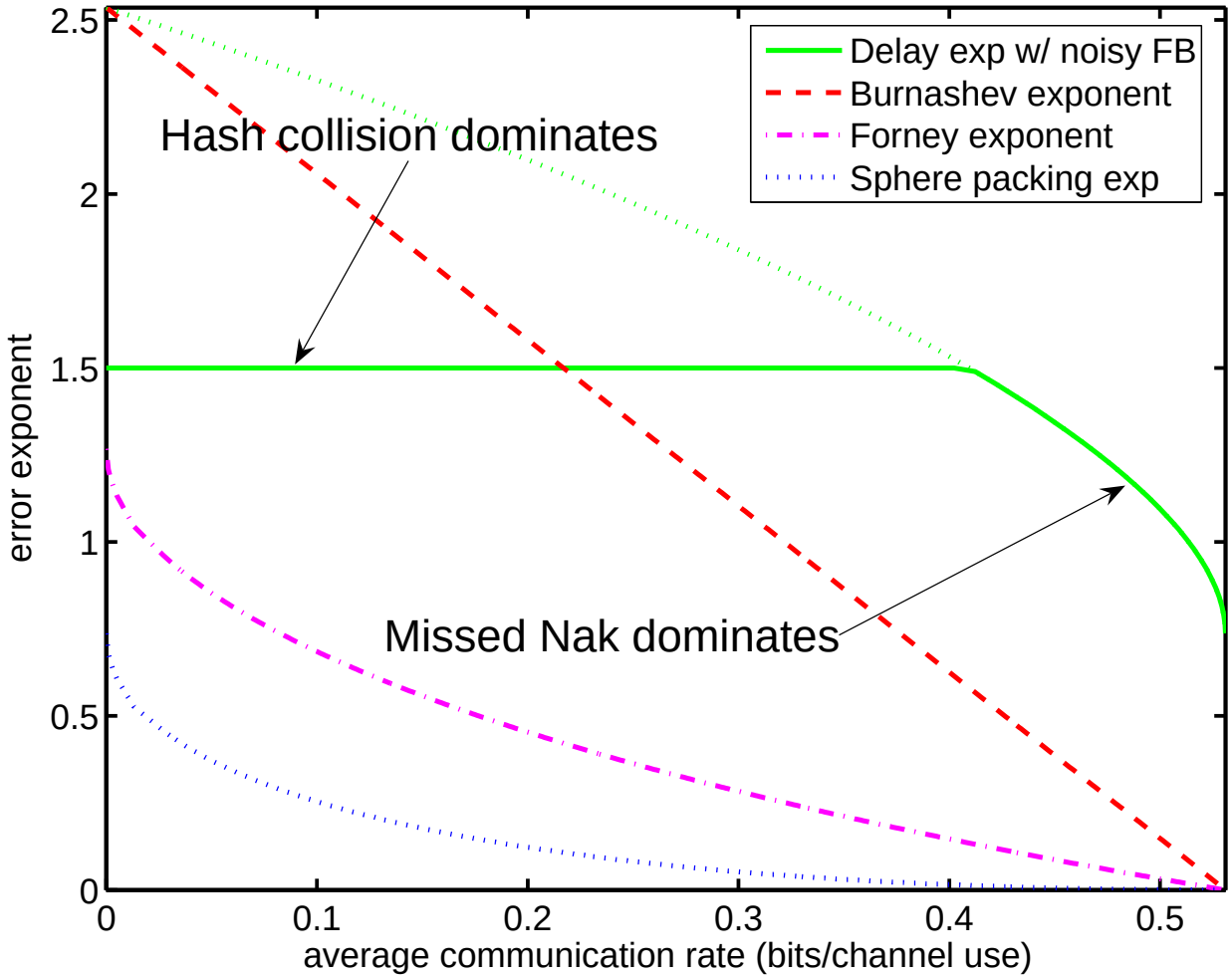
- Must know if any errors in the sliding window.
- Identification problem since encoder knows what it wants to hear.
- Use a random hash of entire sliding window.
- Compare Pr(hash collision) with Pr(missed NAK)

What is needed to NAK?



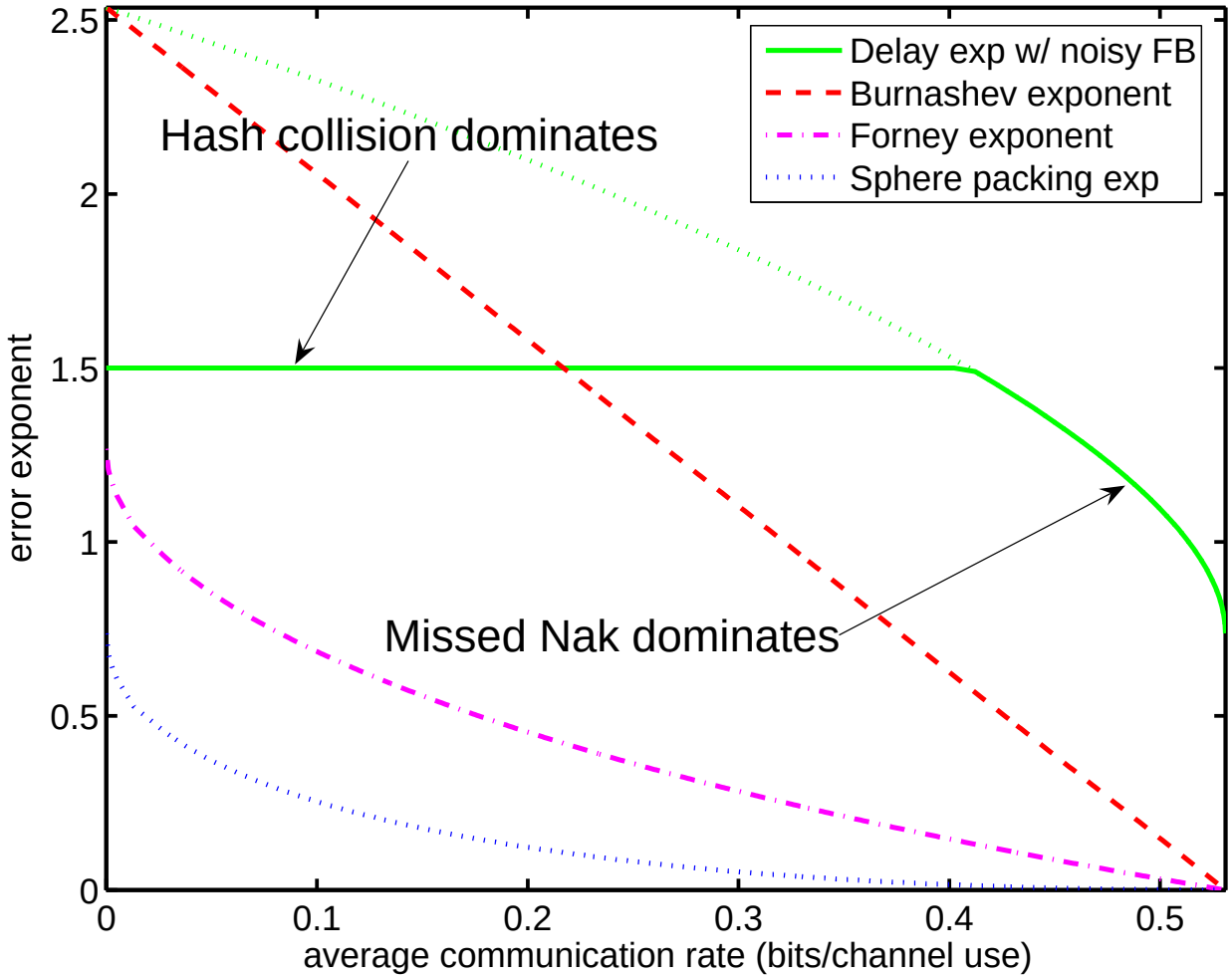
- Must know if any errors in the sliding window.
- Identification problem since encoder knows what it wants to hear.
- Use a random hash of entire sliding window.
- Compare Pr(hash collision) with Pr(missed NAK)
- If $C_{fb} > D(q_y||p)$, no loss in net exponent!

Feedback capacity acts as a ceiling to reliability



- **Feedback reliability gains are robust to noisy feedback.**

Feedback capacity acts as a ceiling to reliability



- **Feedback reliability gains are robust to noisy feedback.**
- Open problem: does this also hold in the general hard deadline case?

Summary of reliability for *symmetric* channels

	Hard Deadlines			**Soft Deadlines** (undetected errors only)		
	Bound	*Achievable*	*Robust*	*Bound*	*Achievable*	*Robust*
Block	Sphere-packing	Yes	trivial	Burnashev	Yes	Mild*
No FB	Sphere-packing	Yes	trivial	Partial(Telatar)	Forney	Yes*
Delay	**Focusing***	**Partial***	**Partial***	**Hallucination***	Yes	**Yes***
No FB	Sphere-packing	Yes	trivial	Unknown	straight-line	Yes*

- * entries are our contributions with **bold** for those in this talk.
- Trivial robustness is when feedback is not used at all.
- Partial achievability of the focusing bound is:
  - Tight for erasure channels and all DMCs with $C_{0,f} > 0$
  - Robust for packet erasure channels with erasure feedback
  - Better than sphere-packing bound for essentially all channels at high rate.

Summary of reliability for *symmetric* channels

	Hard Deadlines			**Soft Deadlines** (undetected errors only)		
	Bound	*Achievable*	*Robust*	*Bound*	*Achievable*	*Robust*
Block	Sphere-packing	Yes	trivial	Burnashev	Yes	Mild*
No FB	Sphere-packing	Yes	trivial	Partial(Telatar)	Forney	Yes*
Delay	**Focusing***	**Partial***	**Partial***	**Hallucination***	Yes	**Yes***
No FB	Sphere-packing	Yes	trivial	Unknown	straight-line	Yes*

- * entries are our contributions with **bold** for those in this talk.
- Trivial robustness is when feedback is not used at all.
- Partial achievability of the focusing bound is:
  - Tight for erasure channels and all DMCs with $C_{0,f} > 0$
  - Robust for packet erasure channels with erasure feedback
  - Better than sphere-packing bound for essentially all channels at high rate.
- For *asymmetric* channels, Haroutunian vs sphere-packing style gaps exist between bounds and achievable codes everywhere except *Block*, *No-FB*.

Summary of reliability for *symmetric* channels

	Hard Deadlines			**Soft Deadlines** (undetected errors only)		
	Bound	*Achievable*	*Robust*	*Bound*	*Achievable*	*Robust*
Block	Sphere-packing	Yes	trivial	Burnashev	Yes	Mild*
No FB	Sphere-packing	Yes	trivial	Partial(Telatar)	Forney	Yes*
Delay	**Focusing***	**Partial***	**Partial***	**Hallucination***	Yes	**Yes***
No FB	Sphere-packing	Yes	trivial	Unknown	straight-line	Yes*

- * entries are our contributions with **bold** for those in this talk.
- Trivial robustness is when feedback is not used at all.
- Partial achievability of the focusing bound is:
  - Tight for erasure channels and all DMCs with $C_{0,f} > 0$
  - Robust for packet erasure channels with erasure feedback
  - Better than sphere-packing bound for essentially all channels at high rate.
- For *asymmetric* channels, Haroutunian vs sphere-packing style gaps exist between bounds and achievable codes everywhere except *Block*, *No-FB*.
- Ultimately, there should be a continuum between perfect feedback and no feedback, as well as between hard and soft deadlines.