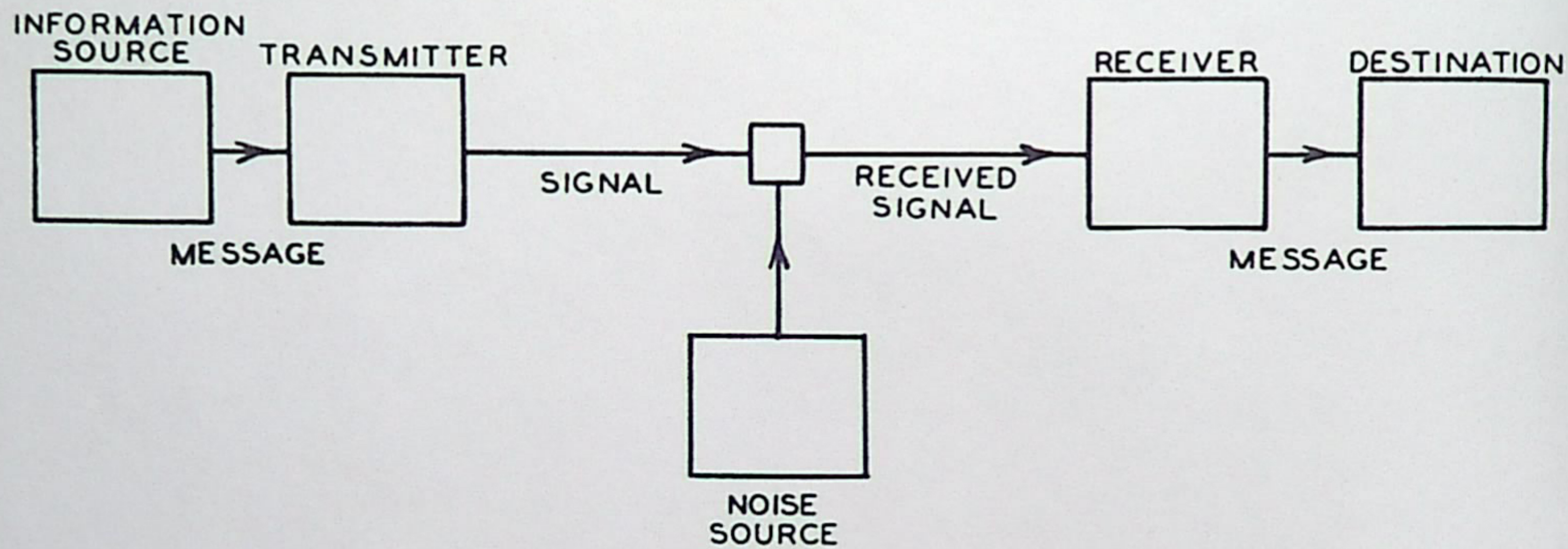
What Hockey Teams and Foraging Animals Can Teach Us About Feedback Communication

Part I: A Tutorial on Feedback

Aaron Wagner
Cornell University

The “Coat of Arms”

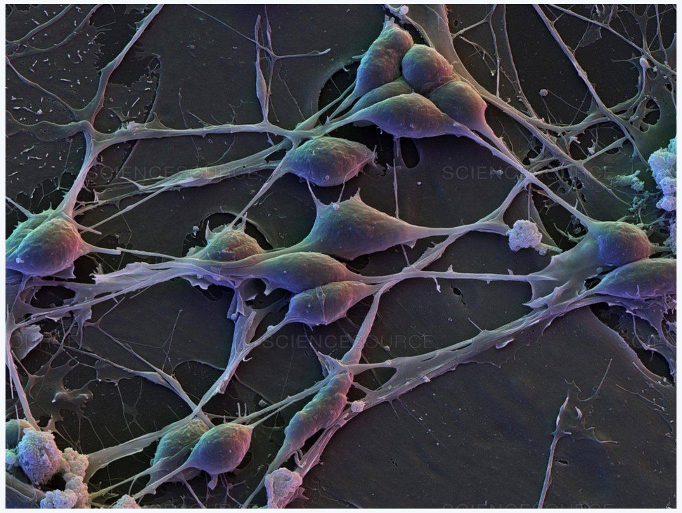


The Information Theorist’s Coat of Arms

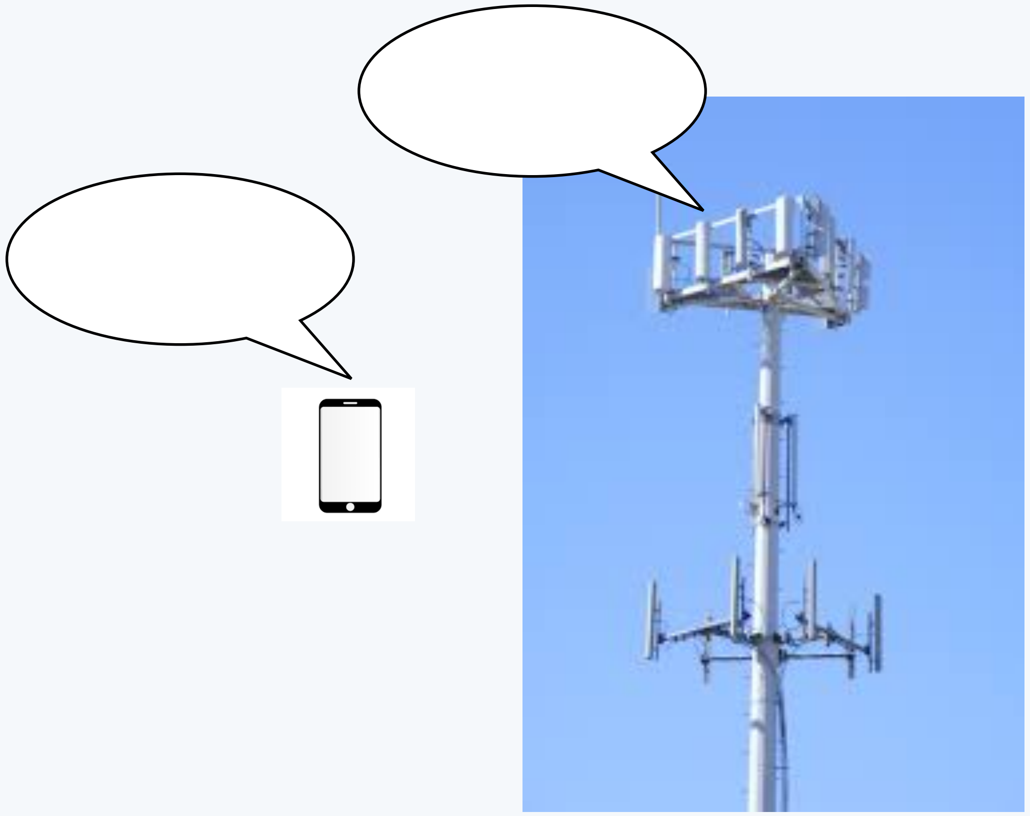
Shannon’s canonical block diagram of the one-way communication system. (Reproduced with permission from “A Mathematical Theory of Communication,” C. E. Shannon, *Bell System Technical Journal*, October 1948.)

[source: *Key Papers in the Development of Information Theory*]

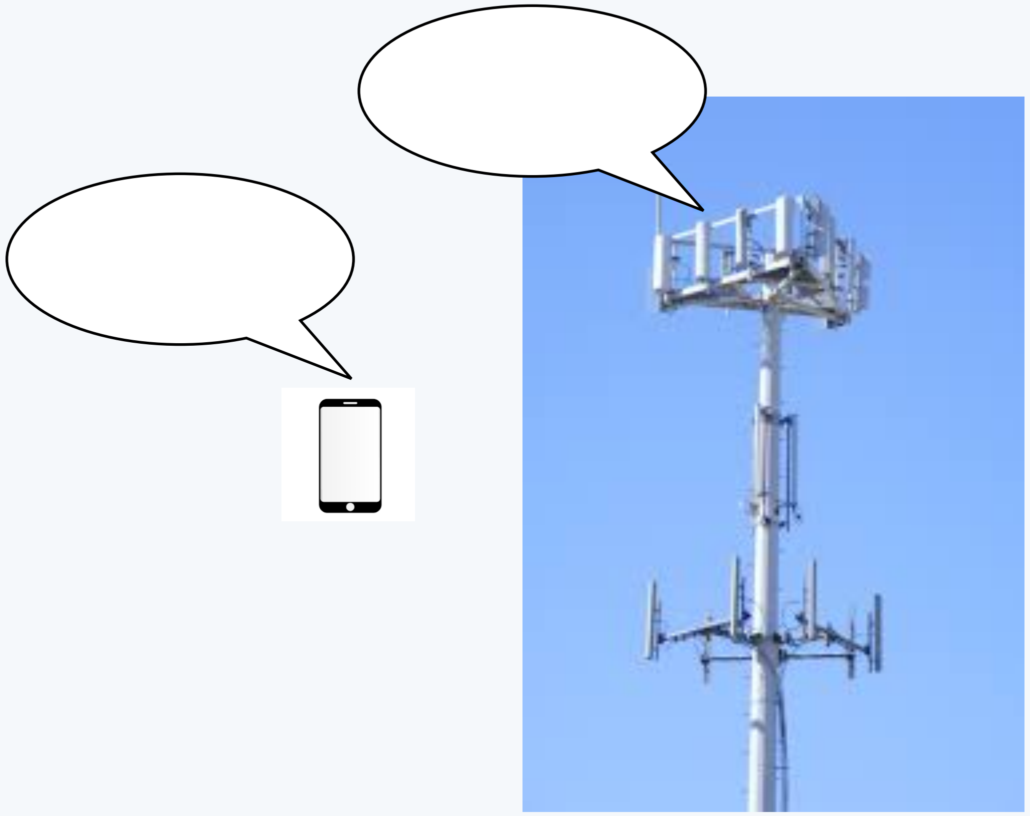
Communication *Without* Feedback is the Exception

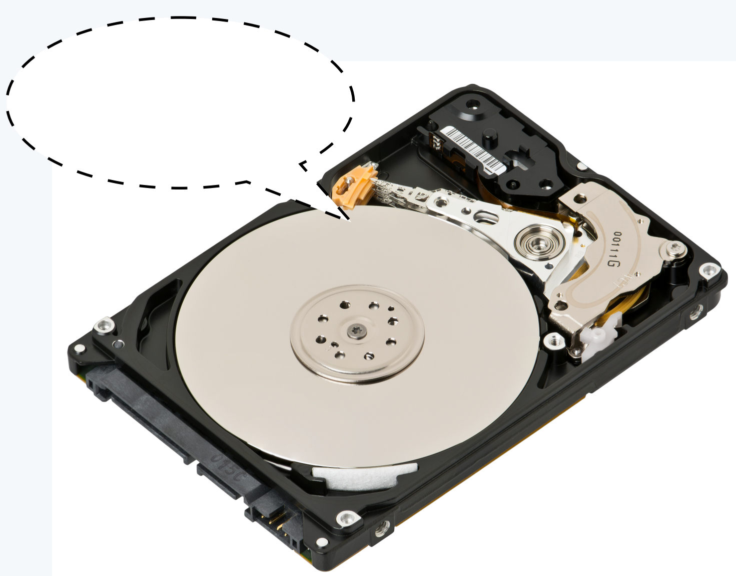


Communication *Without* Feedback is the Exception

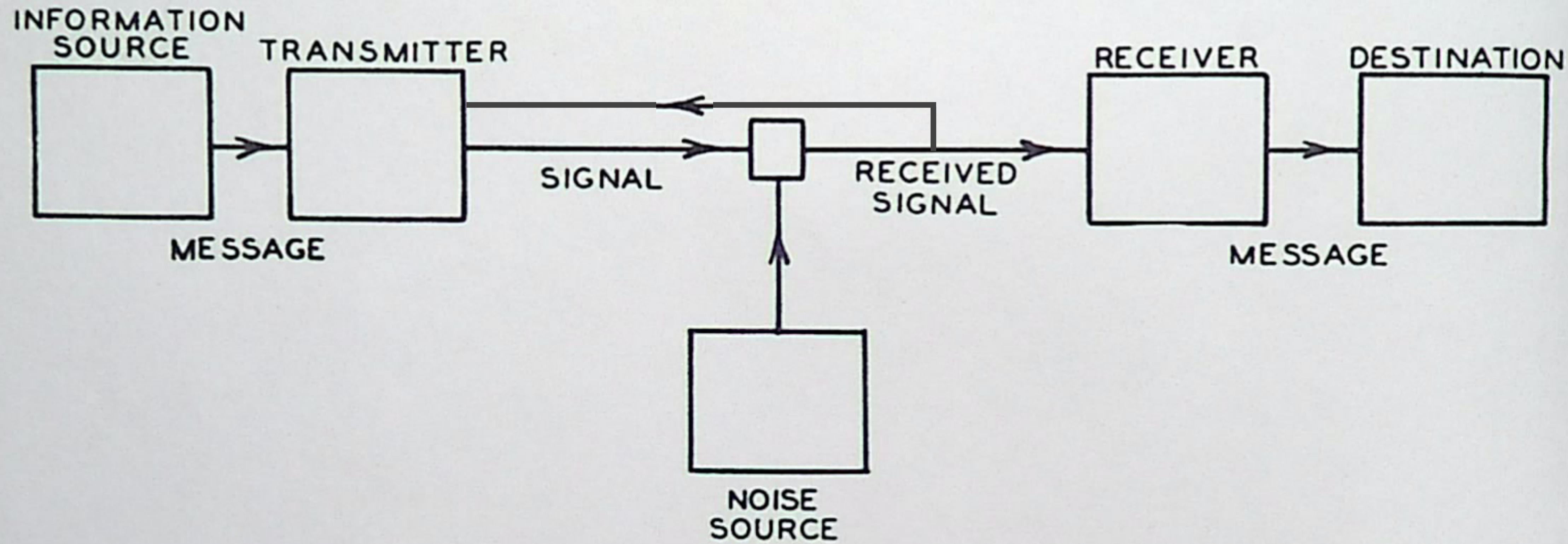


Communication *Without* Feedback is the Exception





A Doctored Coat of Arms

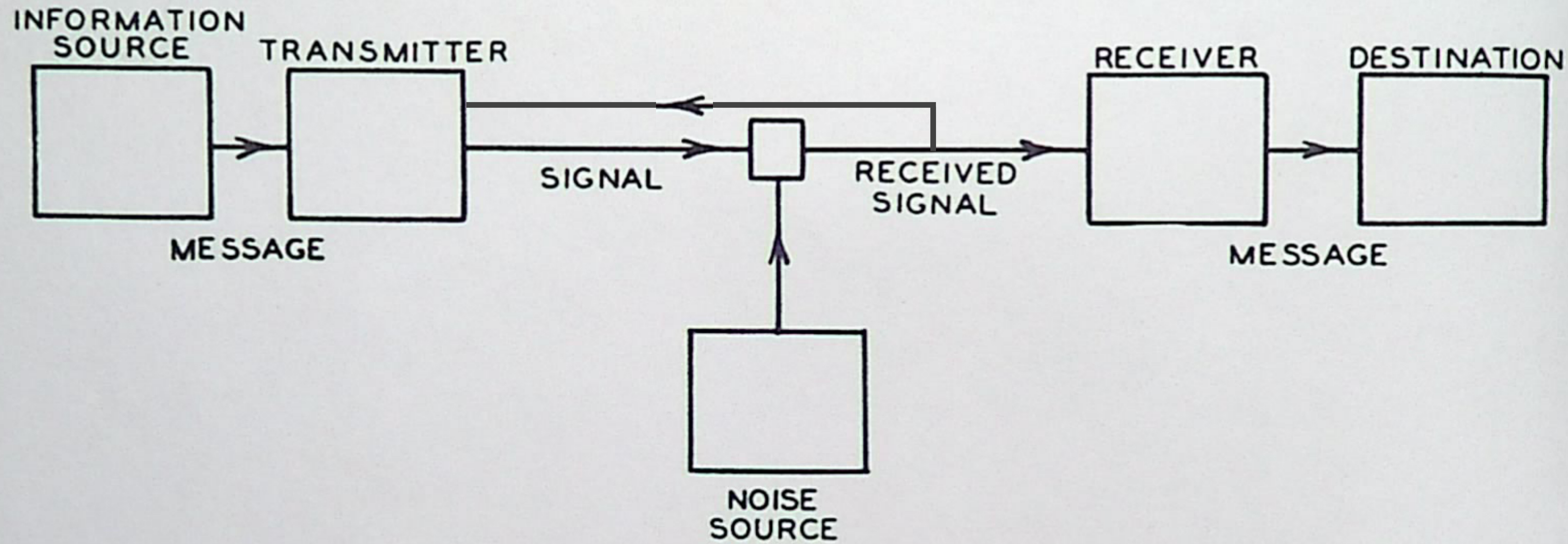


The Information Theorist's Coat of Arms

Shannon's canonical block diagram of the one-way communication system. (Reproduced with permission from "A Mathematical Theory of Communication," C. E. Shannon, *Bell System Technical Journal*, October 1948.)

[source: *Key Papers in the Development of Information Theory*]

A Doctored Coat of Arms



The Information Theorist's Coat of Arms

Shannon's canonical block diagram of the one-way communication system. (Reproduced with permission from "A Mathematical Theory of Communication," C. E. Shannon, *Bell System Technical Journal*, October 1948.)

[source: *Key Papers in the Development of Information Theory*]

[We only consider ideal feedback in this talk]

Mechanisms

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
  - [Other contexts: networks, control over noisy channels, streaming codes, complexity-constrained coding....]

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

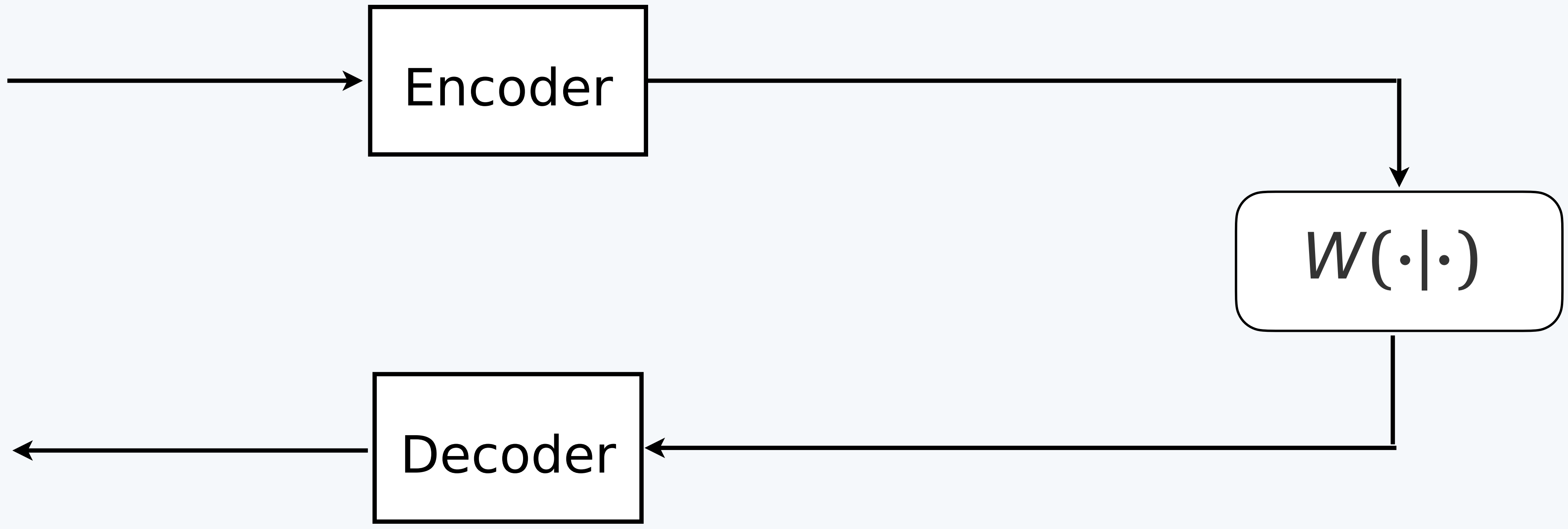
  - [Other contexts: networks, control over noisy channels, streaming codes, complexity-constrained coding....]

Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)

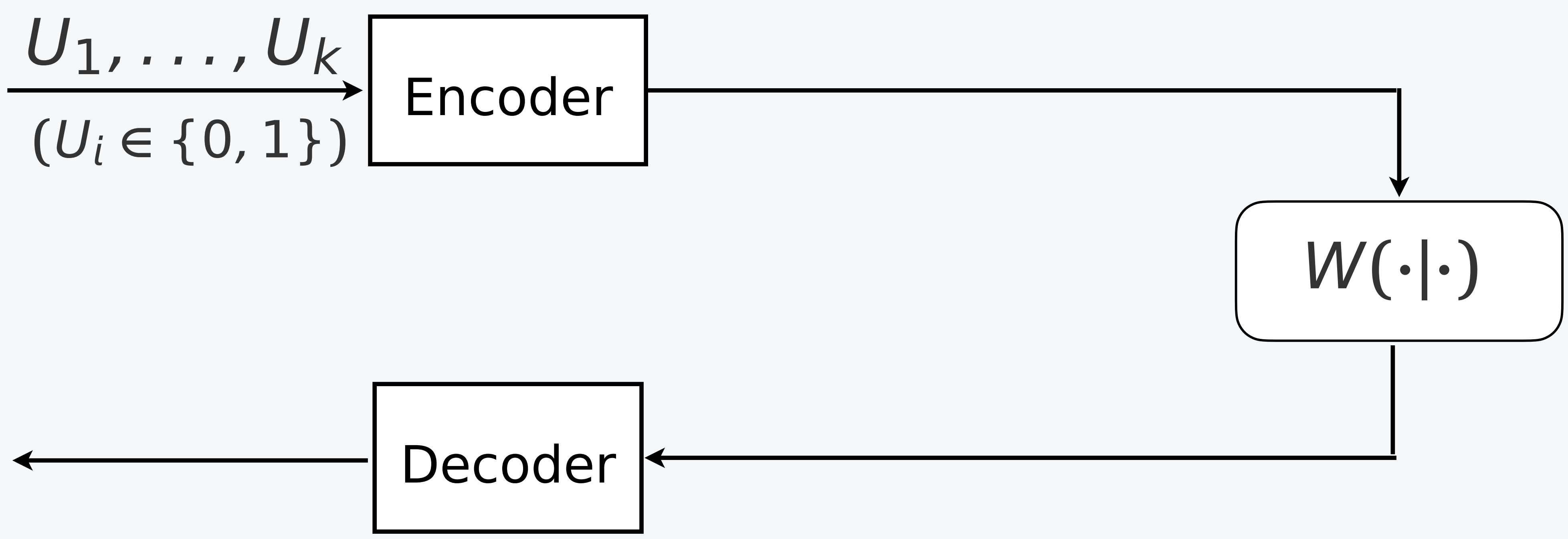
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



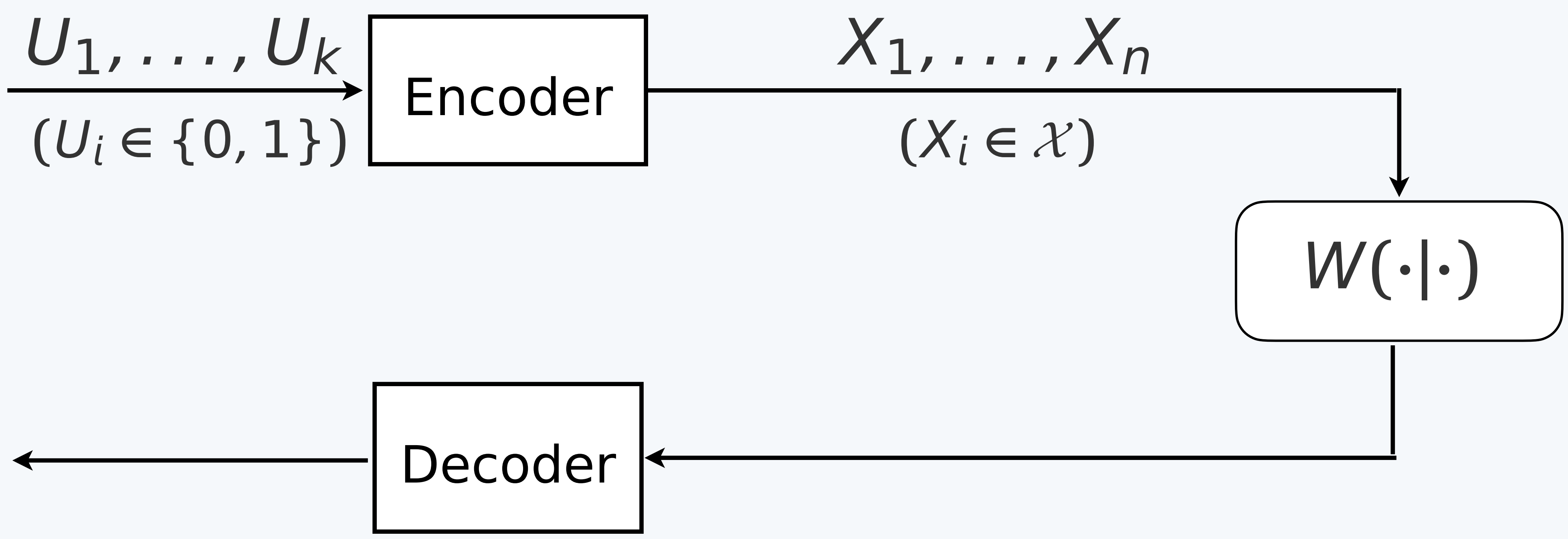
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



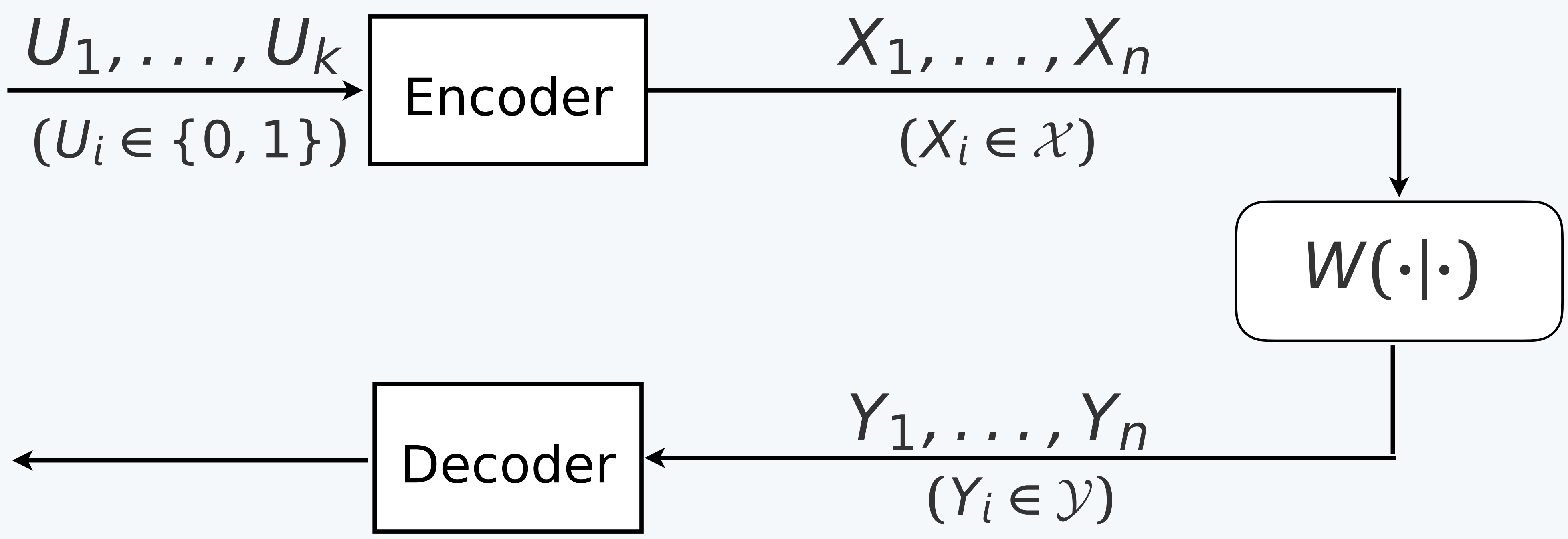
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



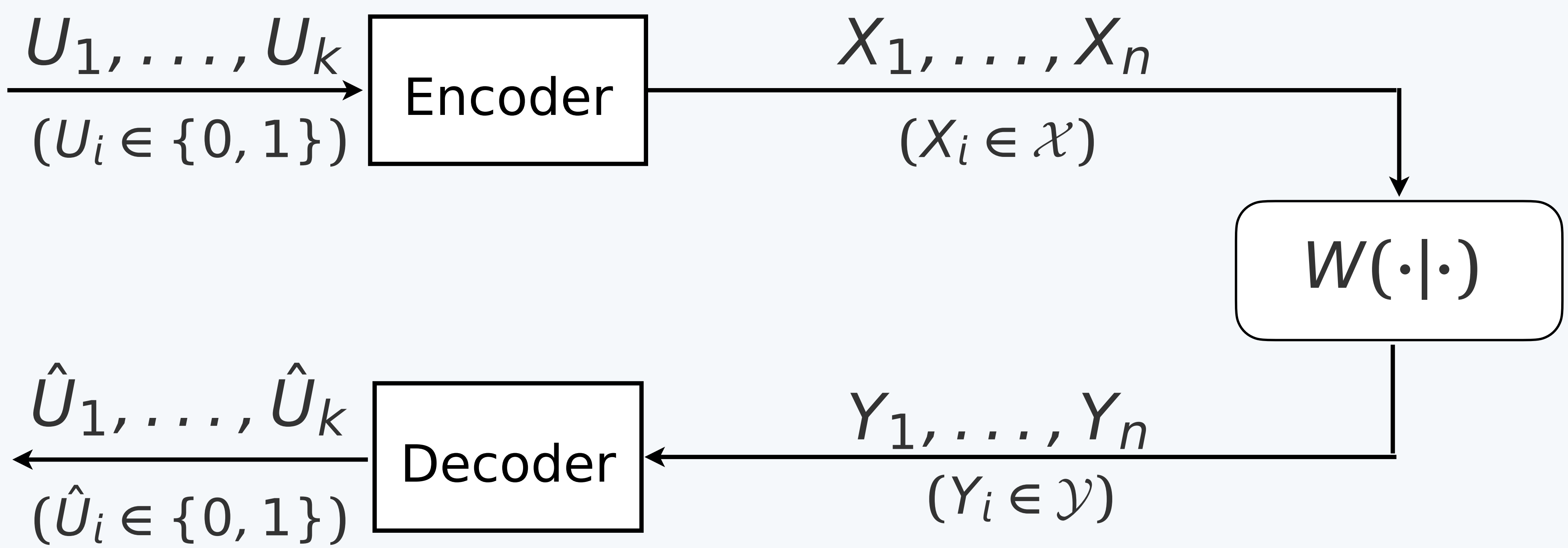
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



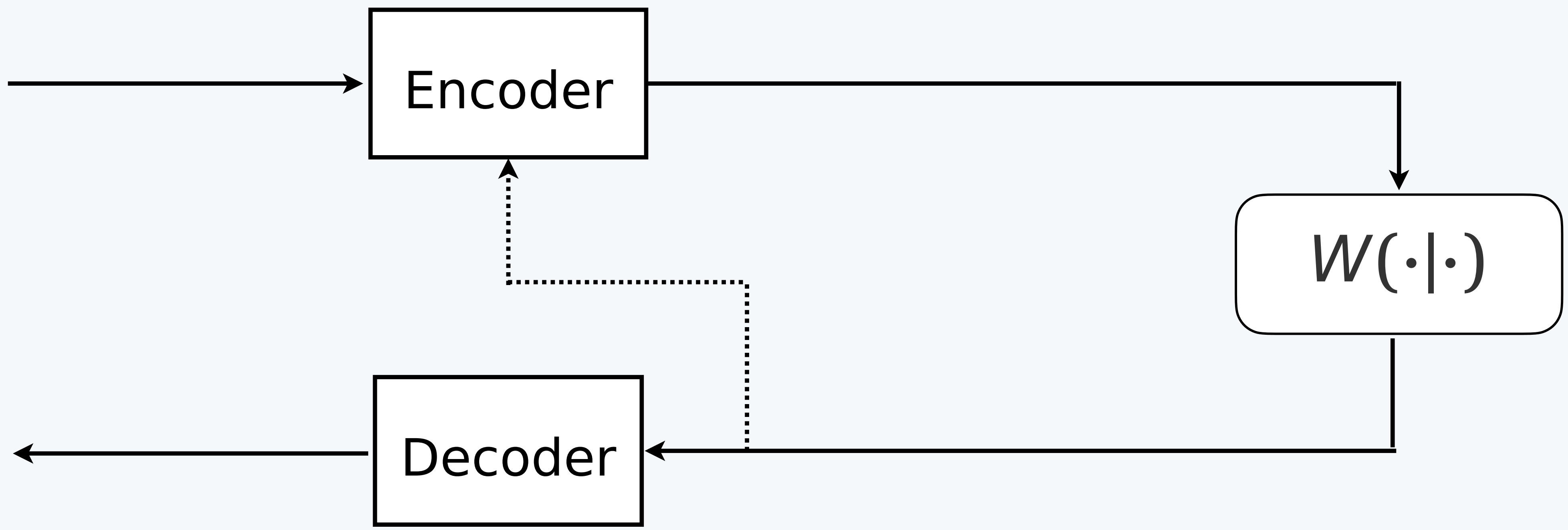
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



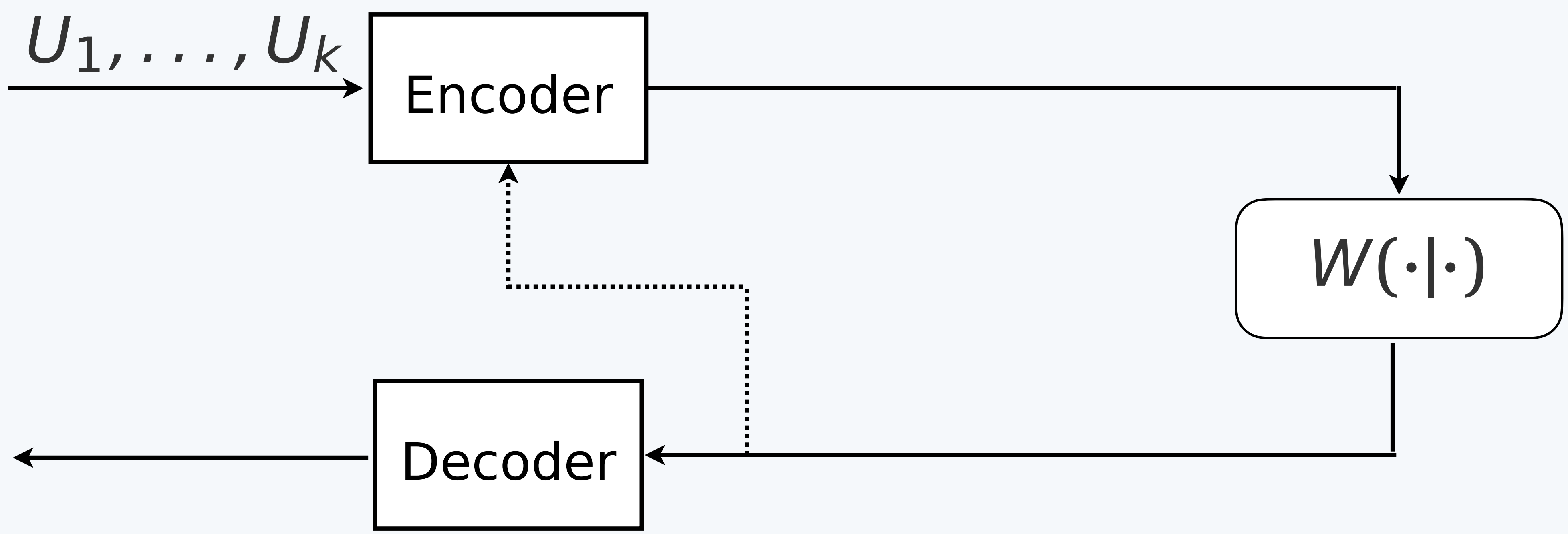
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



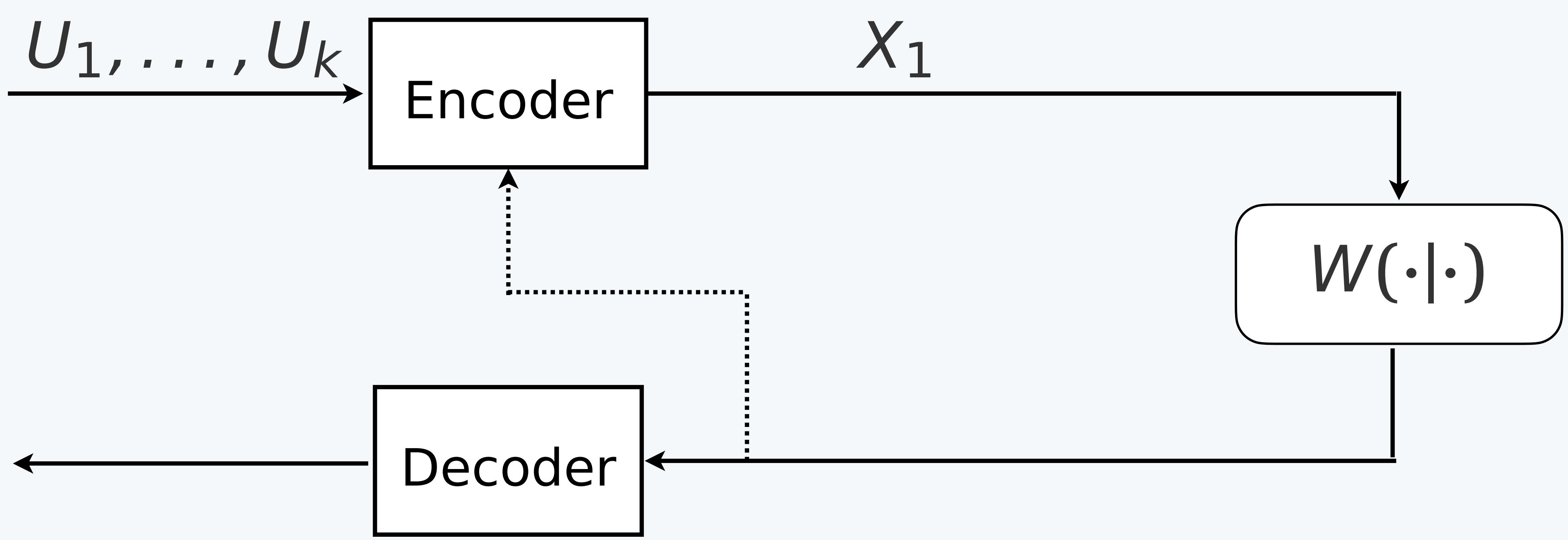
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



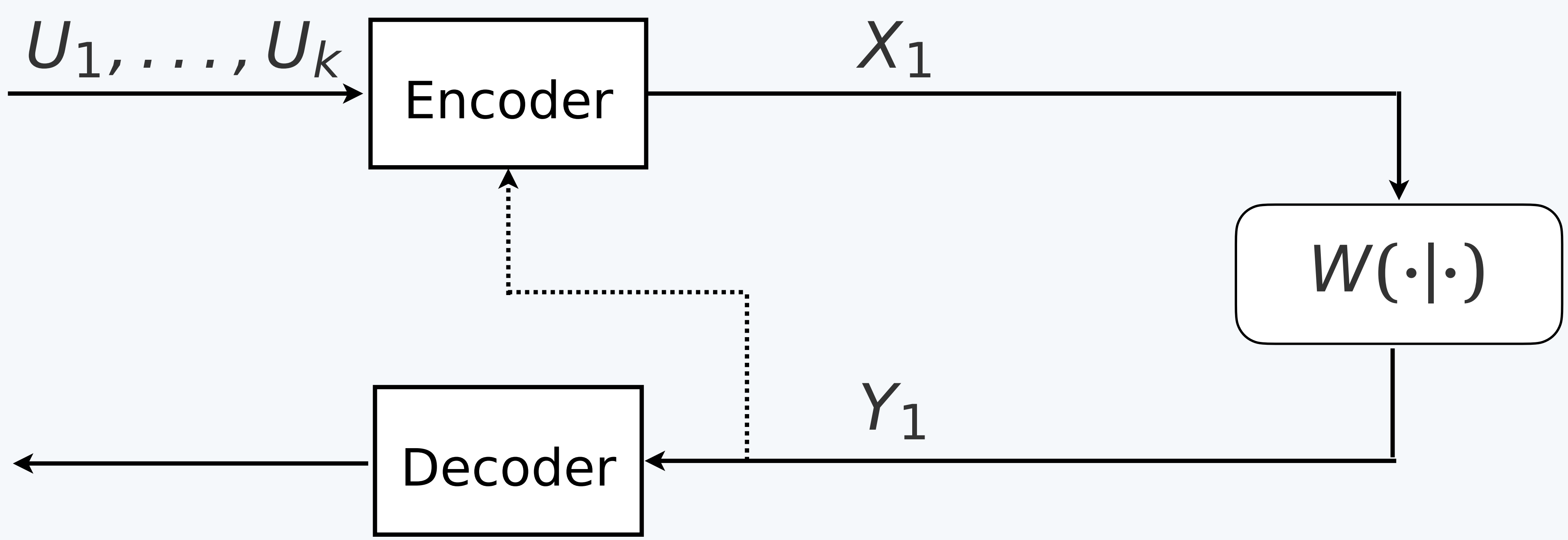
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



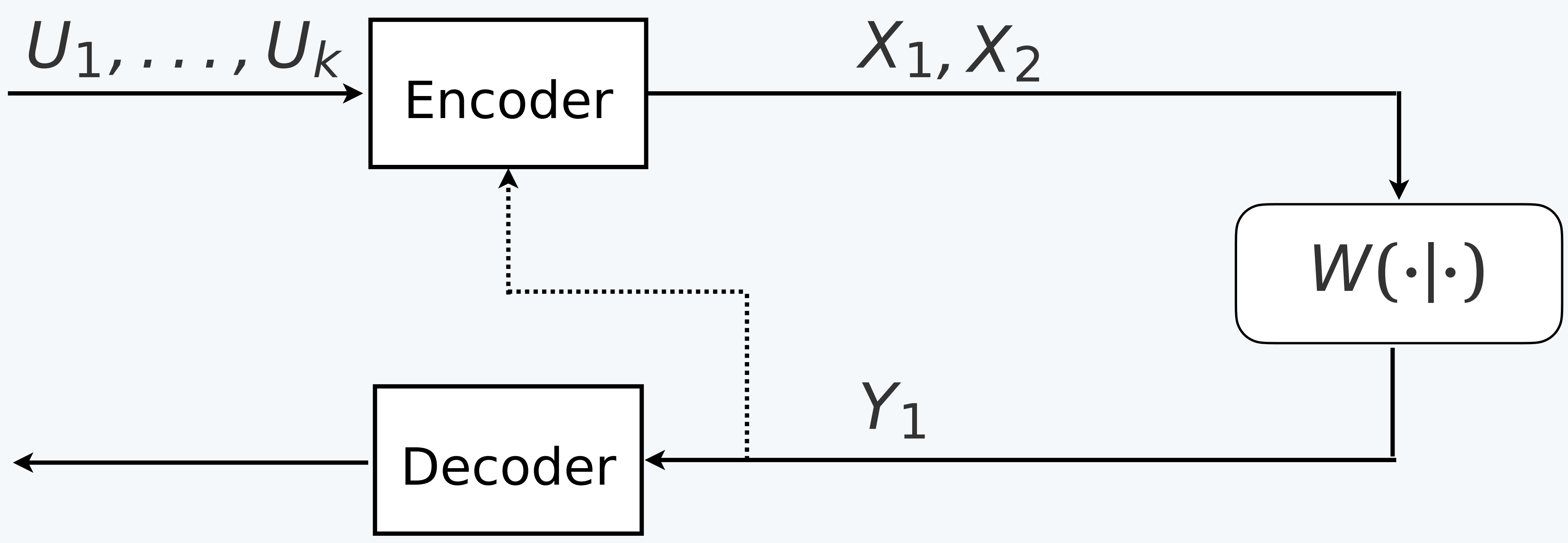
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



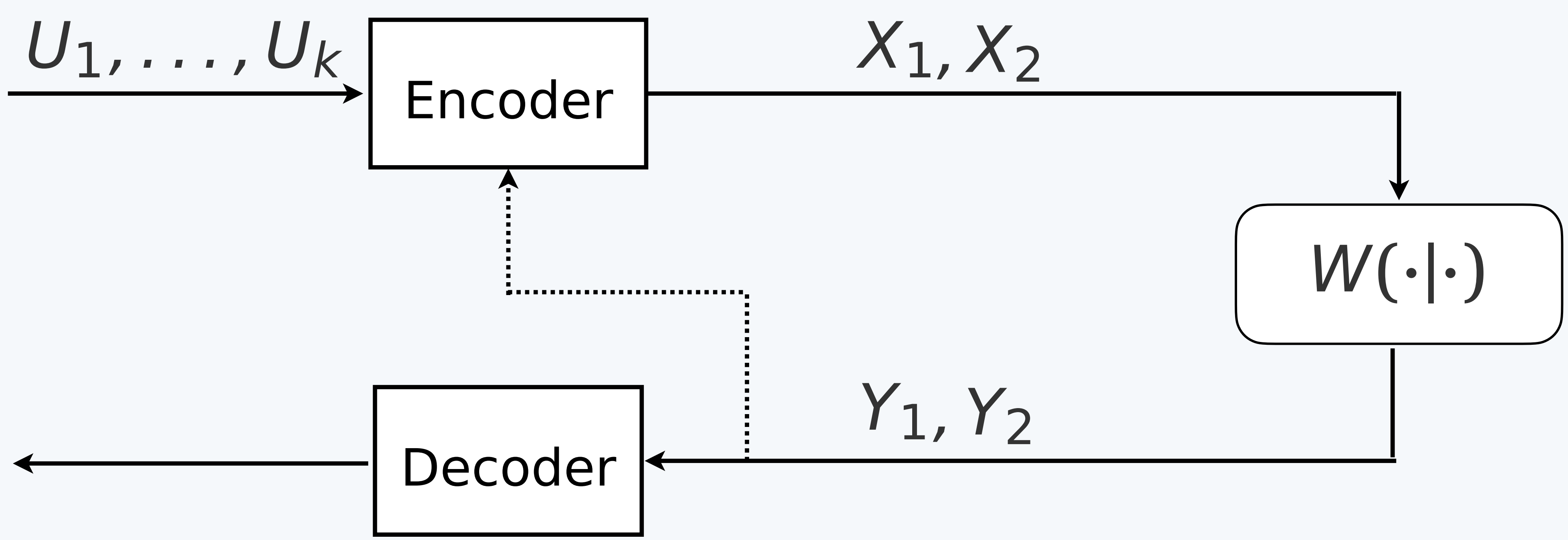
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



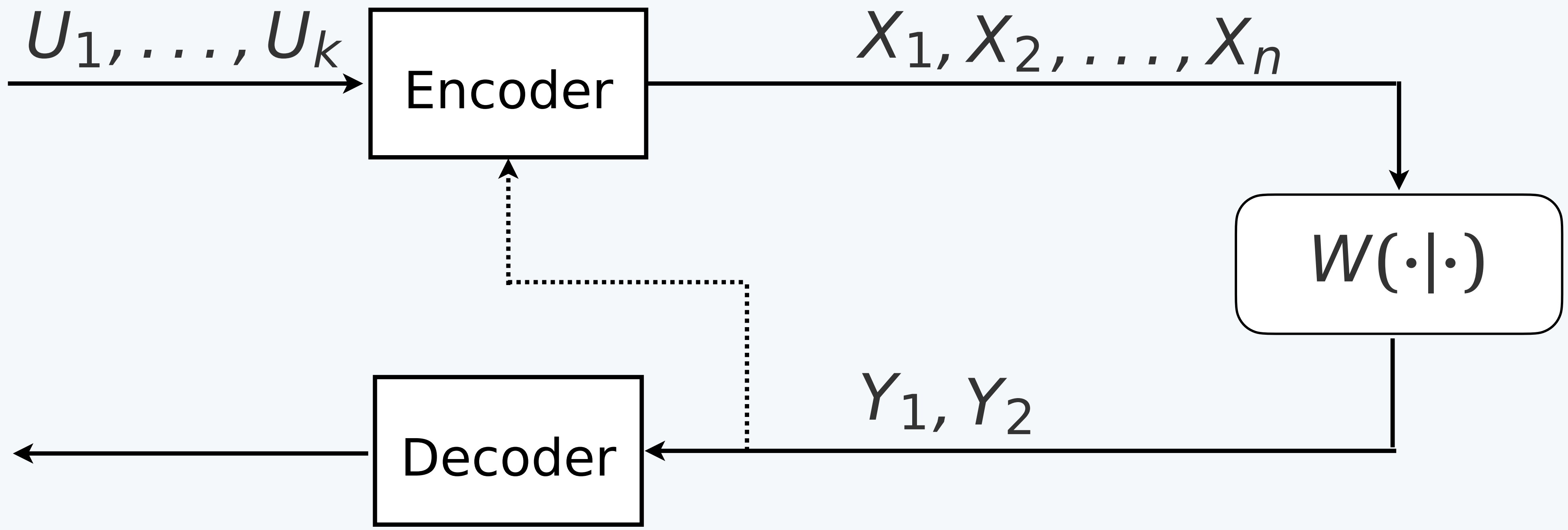
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



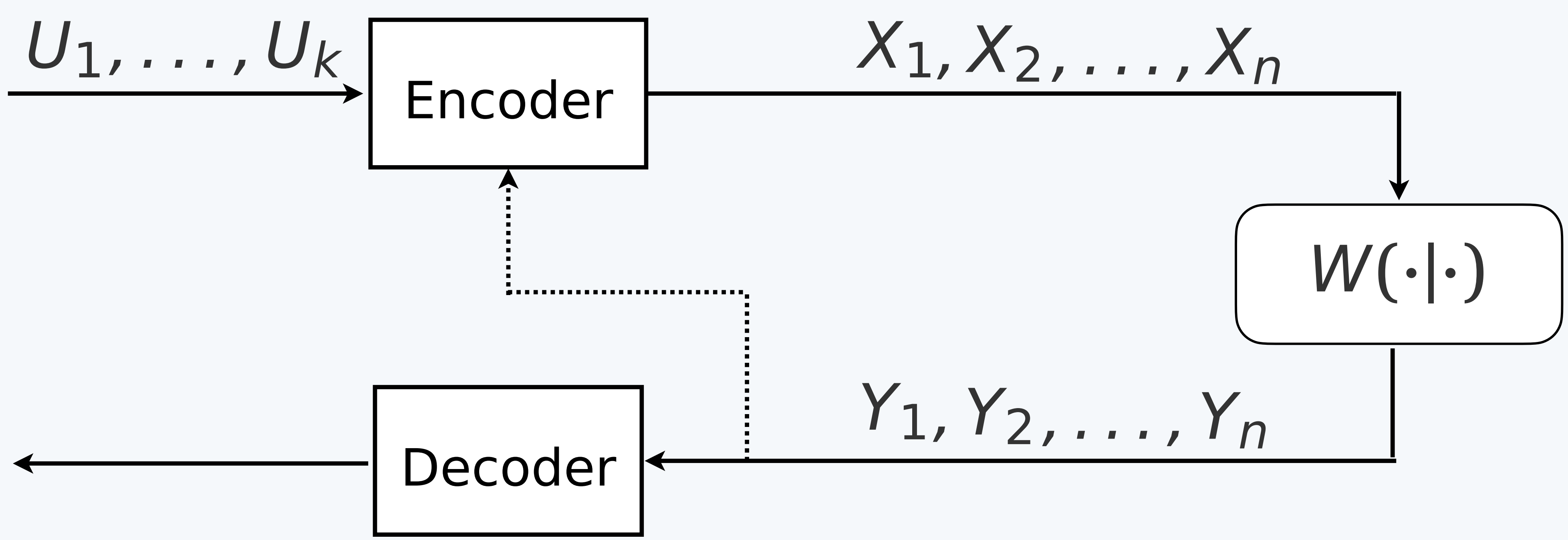
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



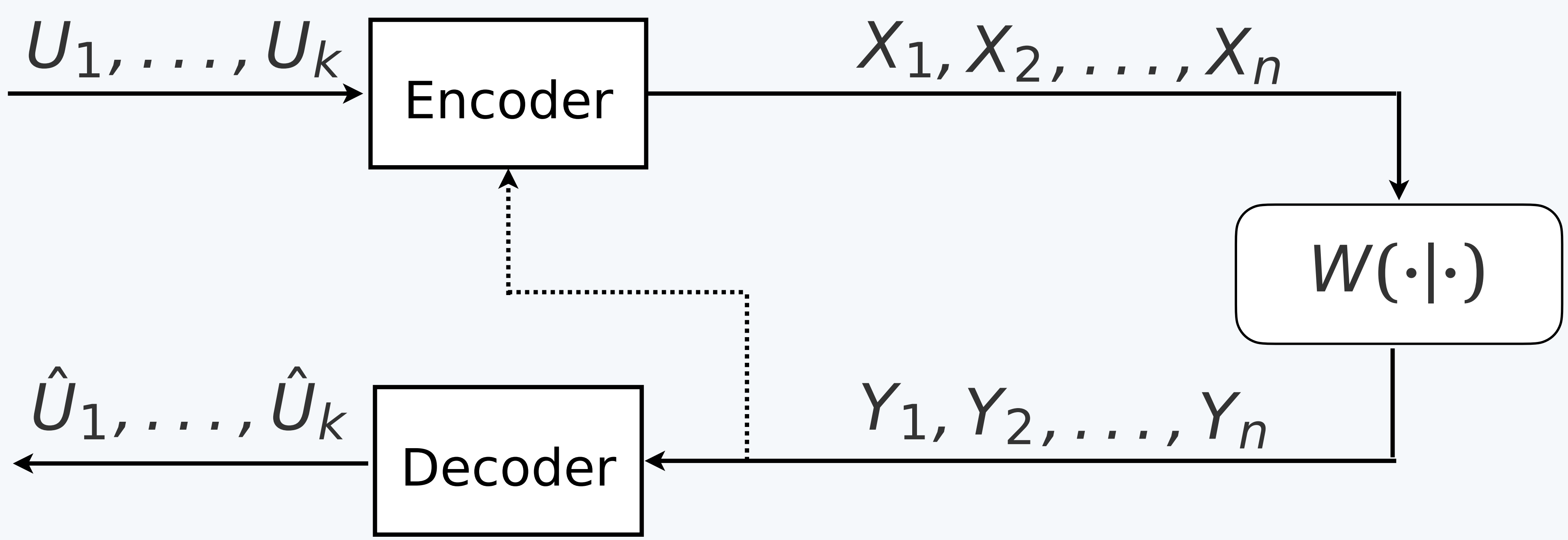
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)

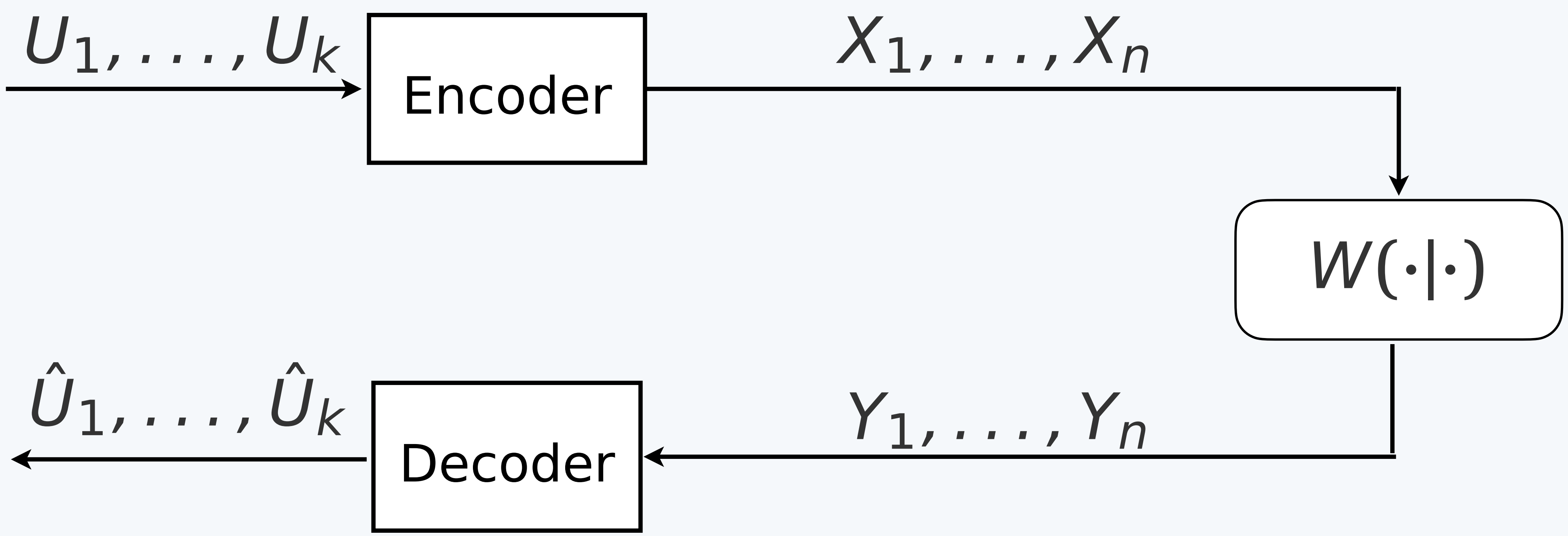


Discrete Memoryless Channels without Feedback

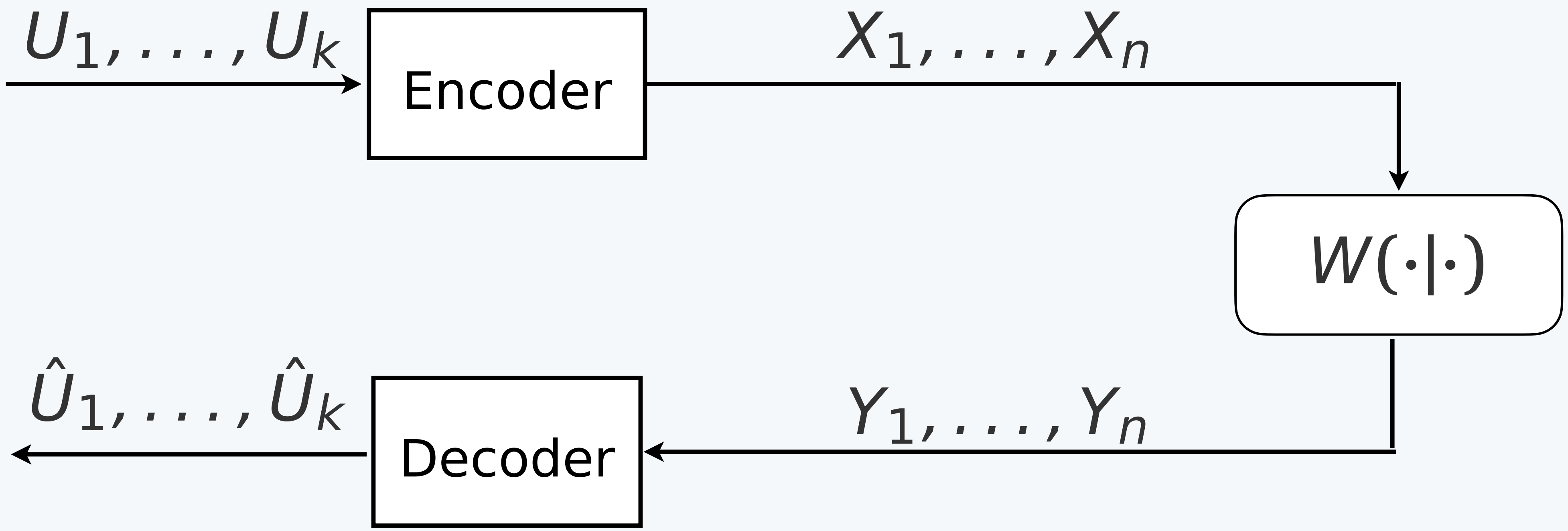
- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)



Figures of Merit

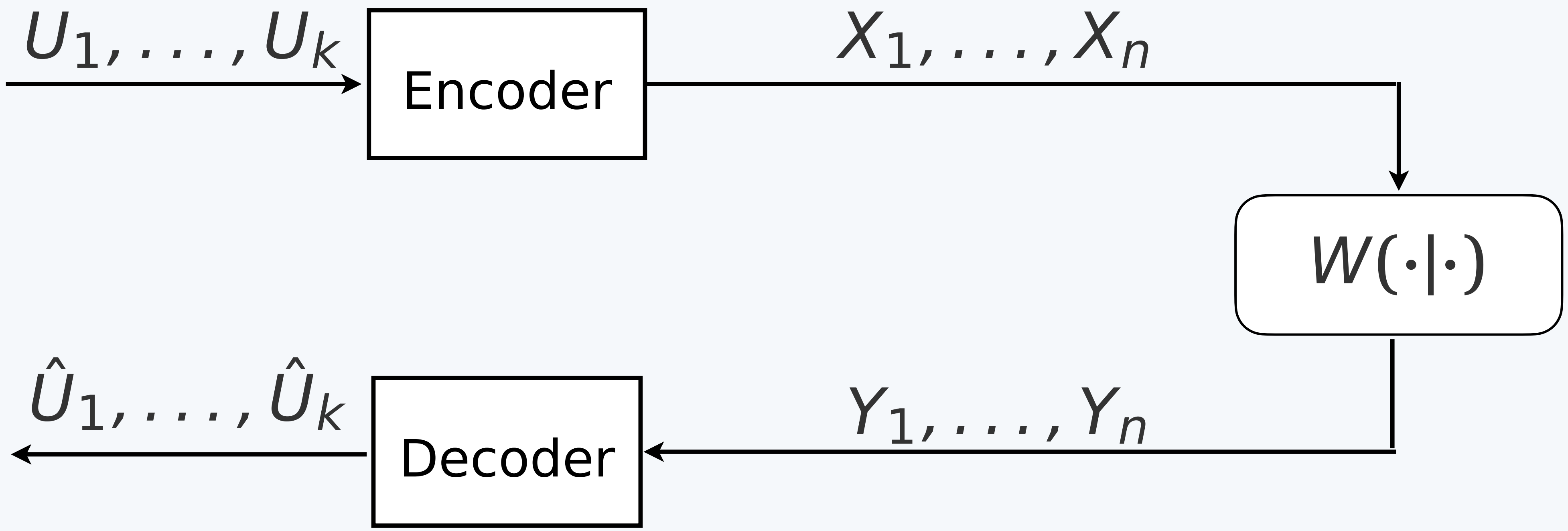


Figures of Merit



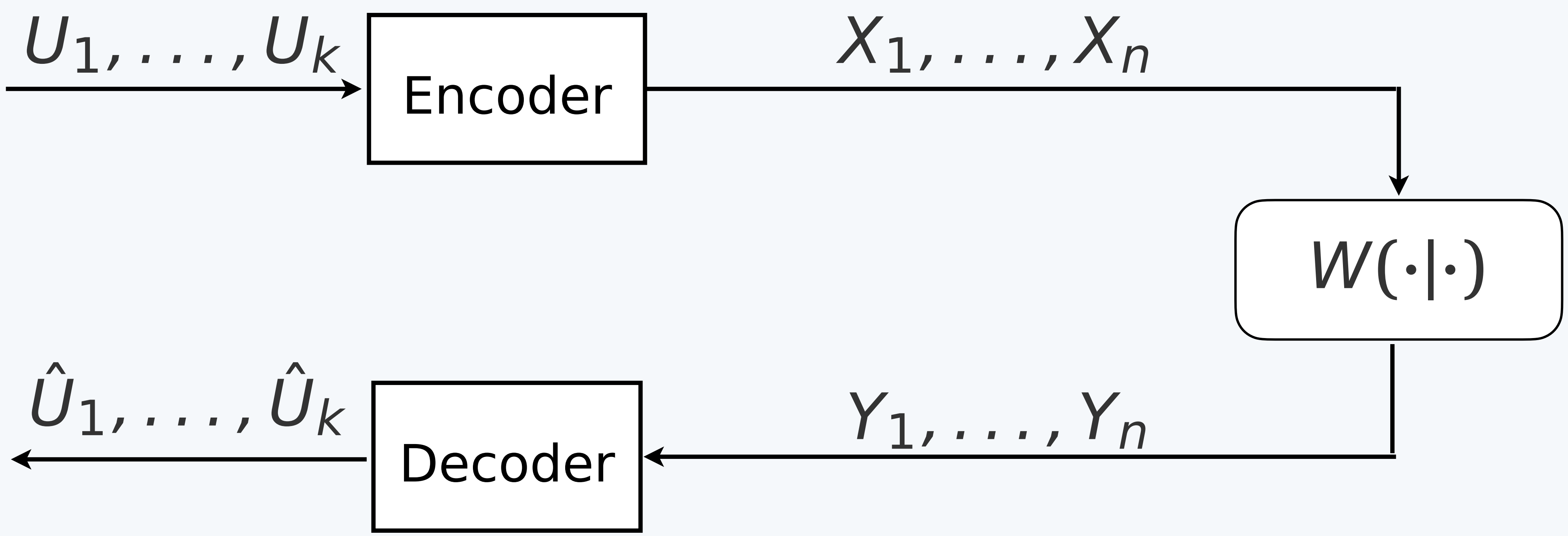
- Number of bits sent: k

Figures of Merit



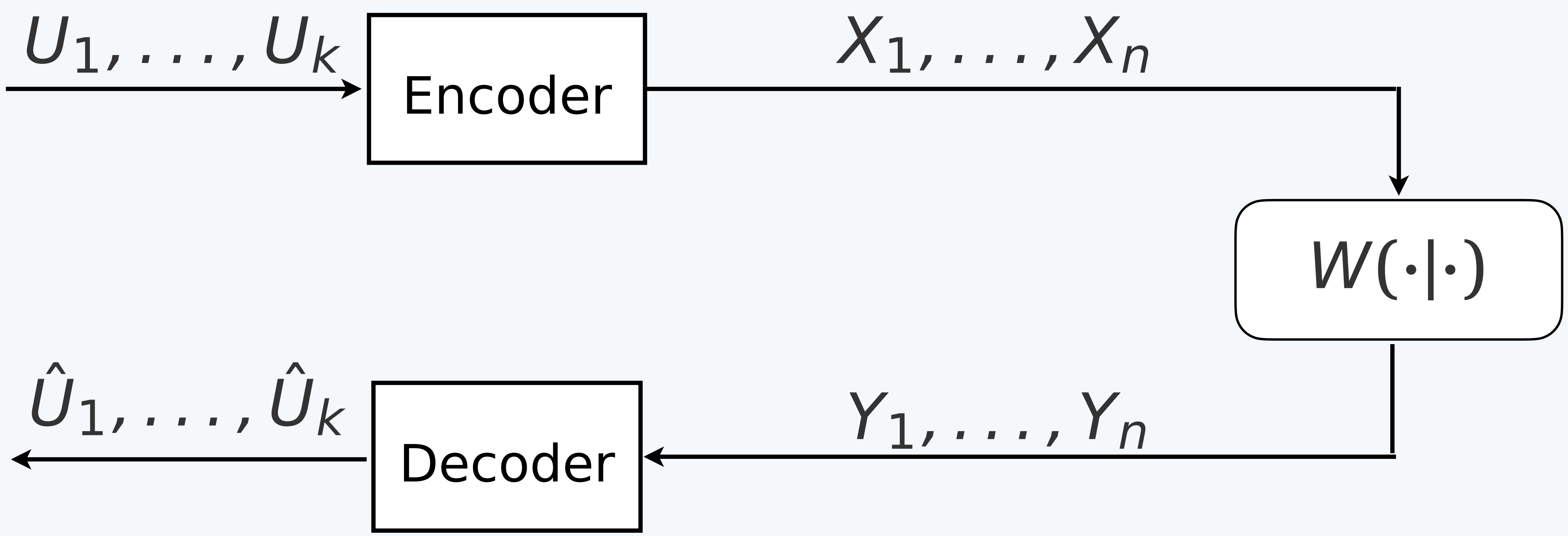
- Number of bits sent: k
- Transmission time: n

Figures of Merit



- Number of bits sent: k
- Transmission time: n
- Rate $R = k/n$

Figures of Merit



- Number of bits sent: k
- Transmission time: n
- Rate $R = k/n$
- Error probability: $P_e = P(U^k \neq \hat{U}^k)$

Asymptotic Metrics

Asymptotic Metrics

- Capacity

Asymptotic Metrics

- Capacity
- Error exponents

Asymptotic Metrics

- Capacity
- Error exponents
- Second-order coding rate (normal approximation)

Asymptotic Metrics

- Capacity
- Error exponents
- Second-order coding rate (normal approximation)
- Moderate deviations performance

I.I.D. Sums

I.I.D. Sums

- Let $\{X_i\}$ be an i.i.d. sequence with zero mean, unit variance

I.I.D. Sums

- Let $\{X_i\}$ be an i.i.d. sequence with zero mean, unit variance
- (Weak) Law of large numbers:

$$\lim_{n\to\infty} \Pr\left(\sum_{i=1}^{n} X_i > \epsilon n\right) = 0 \qquad \epsilon > 0$$

I.I.D. Sums

- Let $\{X_i\}$ be an i.i.d. sequence with zero mean, unit variance
- (Weak) Law of large numbers:

$$\lim_{n\to\infty} \Pr\left(\sum_{i=1}^{n} X_i > \epsilon n\right) = 0 \qquad \epsilon > 0$$

- Large deviations*:

$$\lim_{n\to\infty} -\frac{1}{n}\log \Pr\left(\sum_{i=1}^{n} X_i > \epsilon n\right) = \Lambda^*(\epsilon) > 0 \qquad \epsilon > 0$$

I.I.D. Sums

- Let $\{X_i\}$ be an i.i.d. sequence with zero mean, unit variance
- (Weak) Law of large numbers:

$$\lim_{n\to\infty} \Pr\left(\sum_{i=1}^{n} X_i > \epsilon n\right) = 0 \qquad \epsilon > 0$$

- Large deviations*:

$$\lim_{n\to\infty} -\frac{1}{n}\log \Pr\left(\sum_{i=1}^{n} X_i > \epsilon n\right) = \Lambda^*(\epsilon) > 0 \qquad \epsilon > 0$$

- Central Limit Theorem (CLT):

$$\lim_{n\to\infty} \Pr\left(\sum_{i=1}^{n} X_i > \epsilon\sqrt{n}\right) = Q(\epsilon)$$

I.I.D. Sums

- Moderate deviations*: if β is in (1/2, 1):

$$\lim_{n\to\infty} -\frac{1}{n^{2\beta-1}} \log \Pr\left(\sum_{i=1}^{n} X_i > \epsilon n^{\beta}\right) = \Lambda^*_{\mathcal{N}}(\epsilon) \quad \epsilon > 0$$

Connection to Channel Coding

Connection to Channel Coding

- **Lemma** (Shannon '57);: For a DMC without feedback, for any input dist. P and any $\theta > 0$, there exists a code with rate R, block length n, and error prob.

$$P_e \leq \Pr\left(\sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta\right) + 2^{-n\theta}$$

Connection to Channel Coding

- **Lemma** (Shannon '57);: For a DMC without feedback, for any input dist. P and any $\theta > 0$, there exists a code with rate R, block length n, and error prob.

$$P_e \leq \Pr\left(\sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta\right) + 2^{-n\theta}$$

↑

i.i.d. $P \circ W$

Connection to Channel Coding

- **Lemma** (Shannon '57);: For a DMC without feedback, for any input dist. P and any $\theta > 0$, there exists a code with rate R, block length n, and error prob.

$$P_e \leq \Pr\left(\sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta\right) + 2^{-n\theta}$$

↑ "information density"

Connection to Channel Coding

- **Lemma** (Shannon '57);: For a DMC without feedback, for any input dist. P and any $\theta > 0$, there exists a code with rate R, block length n, and error prob.

$$P_e \leq \Pr\left(\sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta\right) + 2^{-n\theta}$$

"information density"

- For the information density,

Connection to Channel Coding

- **Lemma** (Shannon '57);: For a DMC without feedback, for any input dist. P and any $\theta > 0$, there exists a code with rate R, block length n, and error prob.

$$P_e \leq \Pr\left(\sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta\right) + 2^{-n\theta}$$

↑ "information density"

- For the information density,
  - Law of large numbers → capacity

Connection to Channel Coding

- **Lemma** (Shannon ’57);: For a DMC without feedback, for any input dist. P and any $\theta > 0$, there exists a code with rate R, block length n, and error prob.

$$P_e \leq \Pr\left(\sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta\right) + 2^{-n\theta}$$

↑ “information density”

- For the information density,
  - Law of large numbers → capacity
  - Large deviations → error exponents

Connection to Channel Coding

- **Lemma** (Shannon '57);: For a DMC without feedback, for any input dist. P and any $\theta > 0$, there exists a code with rate R, block length n, and error prob.

$$P_e \leq \Pr\left(\sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta\right) + 2^{-n\theta}$$

↑ "information density"

- For the information density,
  - Law of large numbers → capacity
  - Large deviations → error exponents
  - Central limit theorem → second-order coding rate

Connection to Channel Coding

- **Lemma** (Shannon '57);: For a DMC without feedback, for any input dist. P and any $\theta > 0$, there exists a code with rate R, block length n, and error prob.

$$P_e \leq \Pr\left(\sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta\right) + 2^{-n\theta}$$

↑ "information density"

- For the information density,
  - Law of large numbers → capacity
  - Large deviations → error exponents
  - Central limit theorem → second-order coding rate
  - Moderate deviations → moderate deviations

Error Exponents

- **Def**:

$$P_e(n, R) = \min \{P_e : \exists \text{ an } (n, k, P_e) \text{ code with } k/n \geq R\}$$

Error Exponents

- **Def**:

$$P_e(n,R) = \min\{P_e : \exists \text{ an } (n,k,P_e) \text{ code with } k/n \geq R\}$$

- **Def**: The *reliability function* or *error exponent* at rate R is

$$E(R) = \lim_{n\to\infty} -\frac{1}{n}\log P_e(n,R)$$

Error Exponents

- **Def**:

$$P_e(n,R) = \min\{P_e : \exists \text{ an } (n,k,P_e) \text{ code with } k/n \geq R\}$$

- **Def**: The *reliability function* or *error exponent* at rate R is

$$E(R) = \lim_{n\to\infty} -\frac{1}{n}\log P_e(n,R)$$

- Characterized w/o feedback for a range of rates close to capacity and at very low rates [Shannon, Gallager, Berlekamp ('67)].

Second-Order Coding Rate

- **Def:**

$$R(n, \epsilon) = \max \left\{ \frac{k}{n} : \exists \text{ an } (n, k, P_e) \text{ code with } P_e \leq \epsilon \right\}$$

Second-Order Coding Rate

- **Def:**

$$R(n,\epsilon) = \max\left\{\frac{k}{n} : \exists \text{ an } (n,k,P_e) \text{ code with } P_e \le \epsilon\right\}$$

$$\left[\text{Think: } R(n,\epsilon) \approx C + \frac{\beta(\epsilon)}{\sqrt{n}} + \cdots\right]$$

Second-Order Coding Rate

- **Def:**

$$R(n,\epsilon) = \max\left\{\frac{k}{n} : \exists \text{ an } (n,k,P_e) \text{ code with } P_e \leq \epsilon\right\}$$

$$\left[\text{Think: } R(n,\epsilon) \approx C + \frac{\beta(\epsilon)}{\sqrt{n}} + \cdots\right]$$

- **Def:** Second-Order Coding Rate (SOCR):

$$\beta(\epsilon) = \lim_{n\to\infty} (R(n,\epsilon) - C)\sqrt{n}$$

Second-Order Coding Rate

- **Def:**

$$R(n,\epsilon) = \max\left\{\frac{k}{n} : \exists \text{ an } (n,k,P_e) \text{ code with } P_e \leq \epsilon\right\}$$

$$\left[\text{Think: } R(n,\epsilon) \approx C + \frac{\beta(\epsilon)}{\sqrt{n}} + \cdots\right]$$

- **Def:** Second-Order Coding Rate (SOCR):

$$\beta(\epsilon) = \lim_{n\to\infty} (R(n,\epsilon) - C)\sqrt{n}$$

- Characterized w/o feedback by Strassen ('62).

Moderate Deviations

- **Theorem** (Altuğ-Wagner '14):

  Consider a DMC without feedback. Let $R_n = C - \epsilon_n$ be s.t.

$$\lim_{n\to\infty} \epsilon_n = 0 \qquad \lim_{n\to\infty} \epsilon_n \sqrt{n} = \infty$$

  Then

$$\lim_{n\to\infty} -\frac{\log P_e(n, R_n)}{\epsilon_n^2 \cdot n} = \frac{1}{2V_{\min}}$$

Moderate Deviations

- **Theorem** (Altuğ-Wagner '14):

  Consider a DMC without feedback. Let $R_n = C - \epsilon_n$ be s.t.

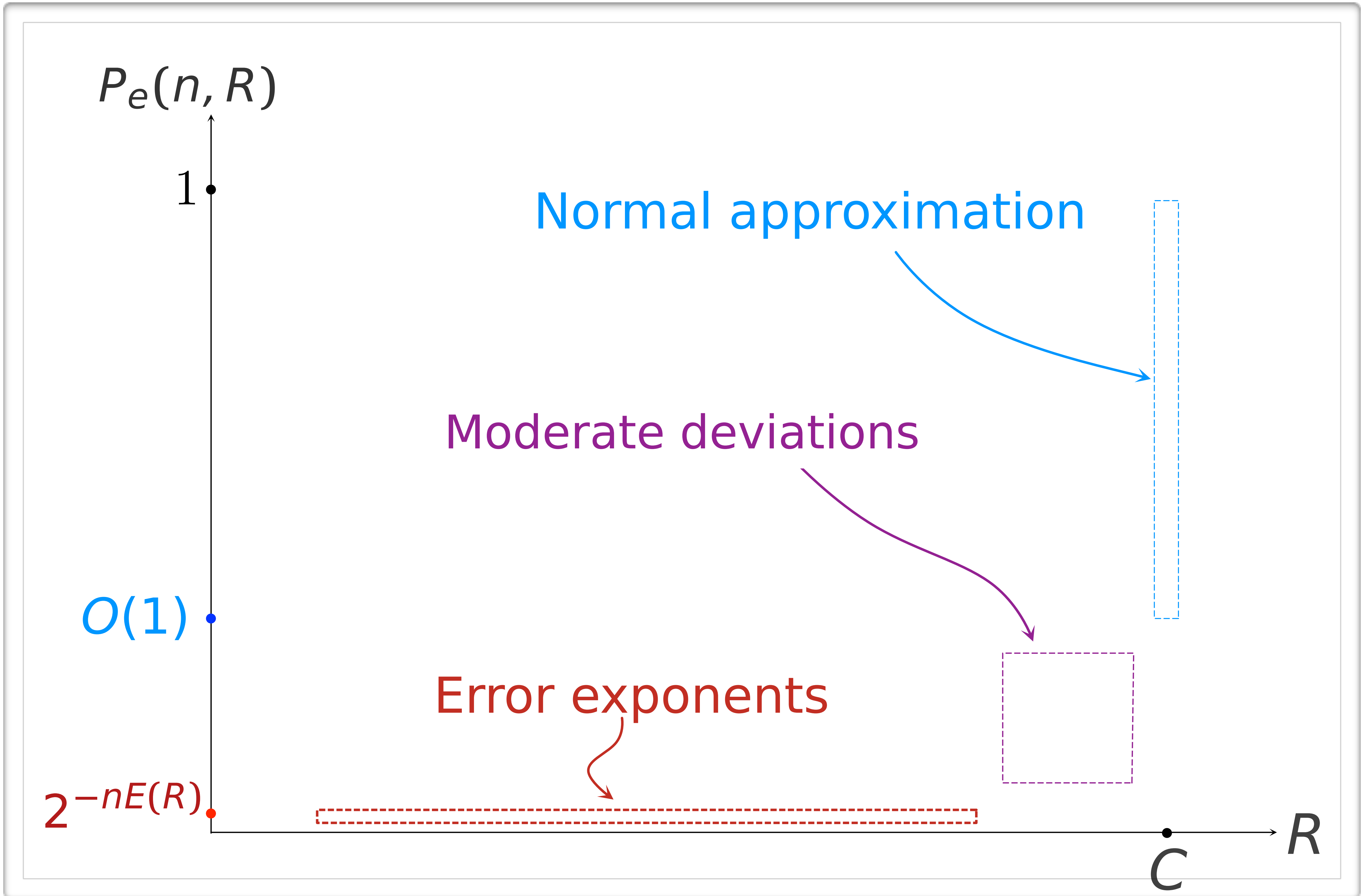
$$\lim_{n\to\infty} \epsilon_n = 0 \qquad \lim_{n\to\infty} \epsilon_n \sqrt{n} = \infty$$

  Then

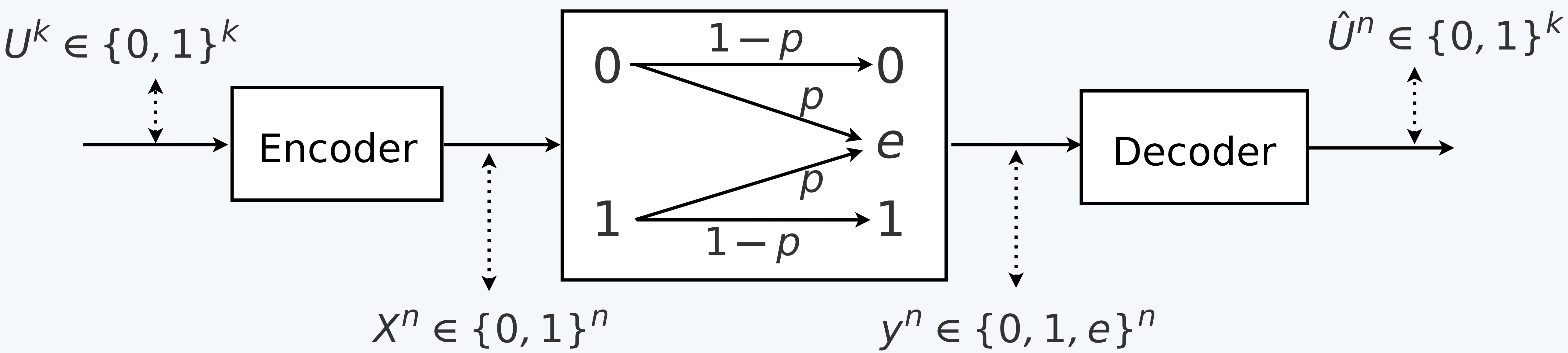
$$\lim_{n\to\infty} -\frac{\log P_e(n, R_n)}{\epsilon_n^2 \cdot n} = \frac{1}{2V_{\min}}$$

constant depending on the channel

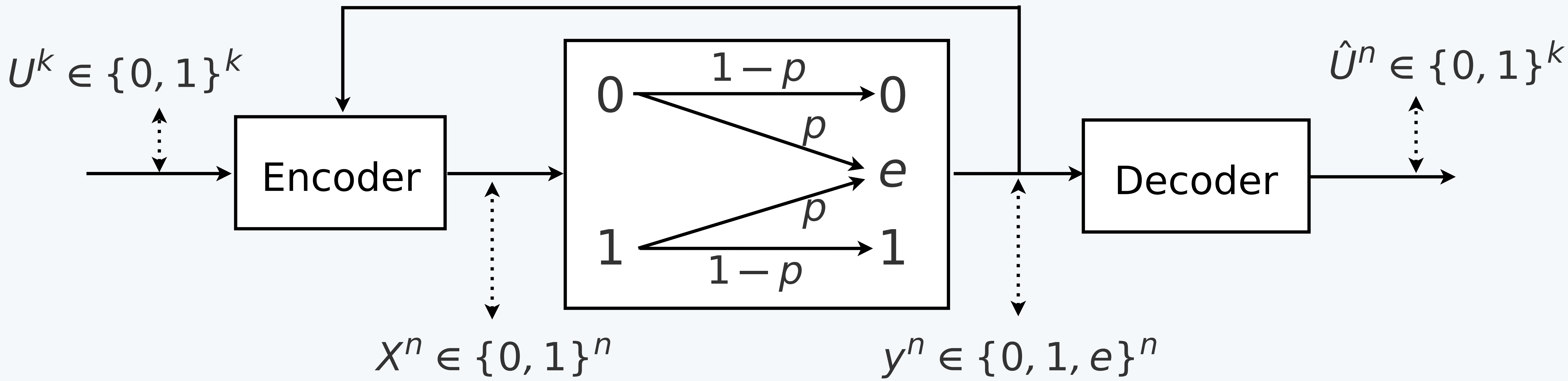
Moderate Deviations



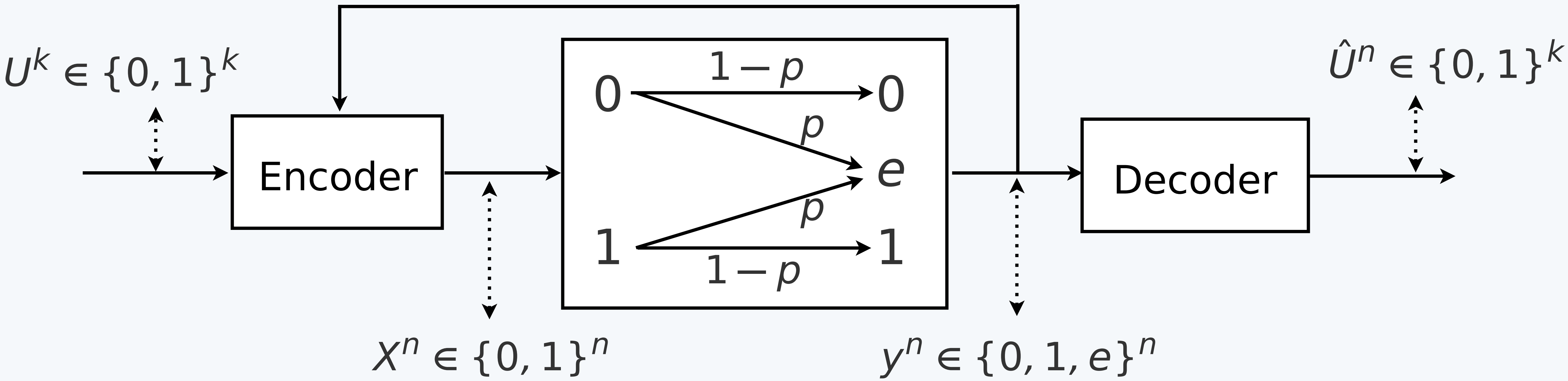
A Non-Example



A Non-Example

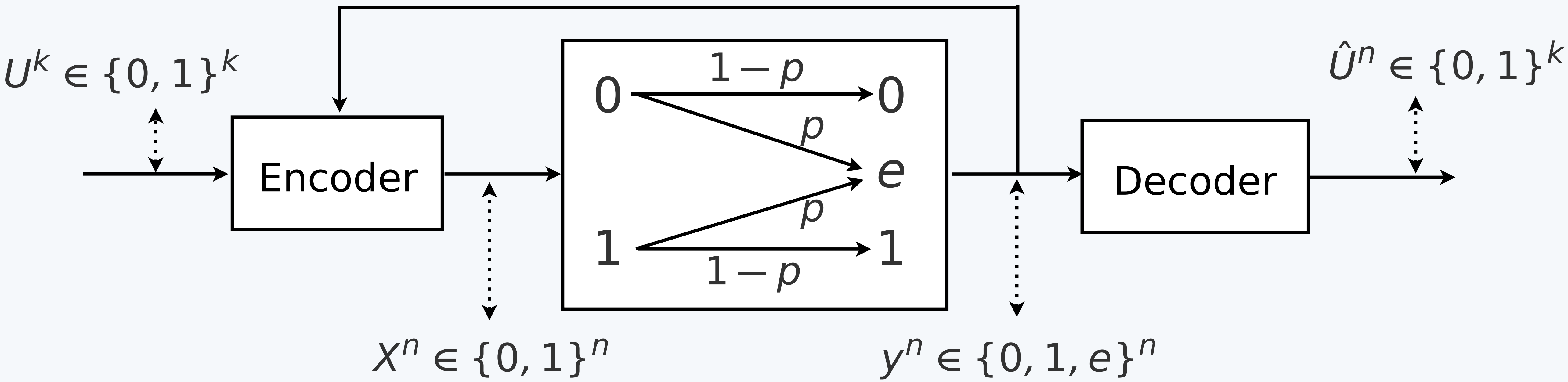


A Non-Example



- Scheme: repeatedly transmit each bit until it gets through

A Non-Example

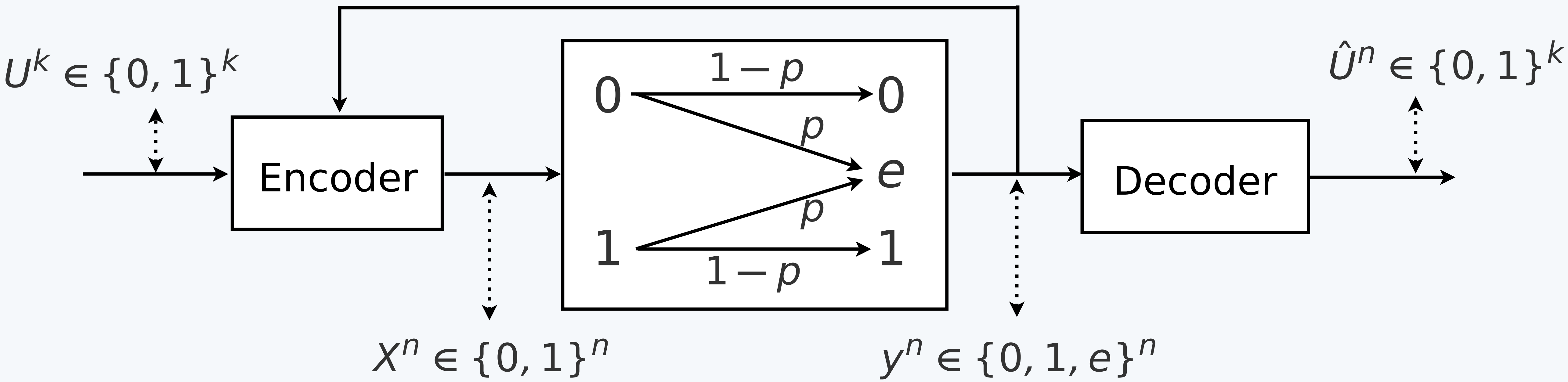


- Scheme: repeatedly transmit each bit until it gets through

$$P_e = \sum_{\ell=0}^{k-1} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot \left(1 - \frac{1}{2^{k-\ell}}\right)$$

$$\leq P\left(\sum_{i=1}^{n} Z_i < k\right)$$

A Non-Example



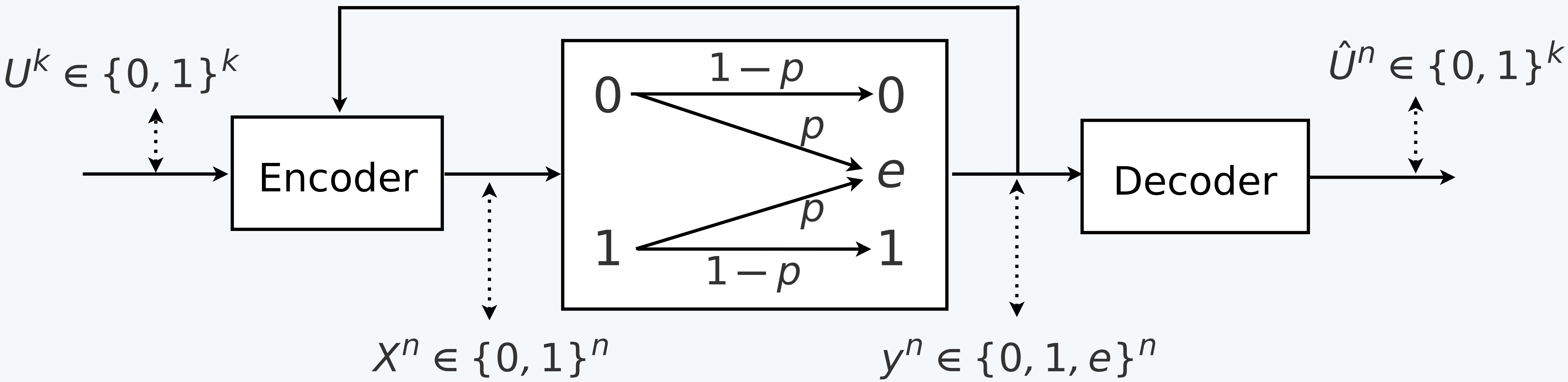
- Scheme: repeatedly transmit each bit until it gets through

[i.i.d. Bernoulli$(1-p)$]

$$P_e = \sum_{\ell=0}^{k-1} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot \left(1 - \frac{1}{2^{k-\ell}}\right)$$

$$\leq P\left(\sum_{i=1}^{n} Z_i < k\right)$$

A Non-Example

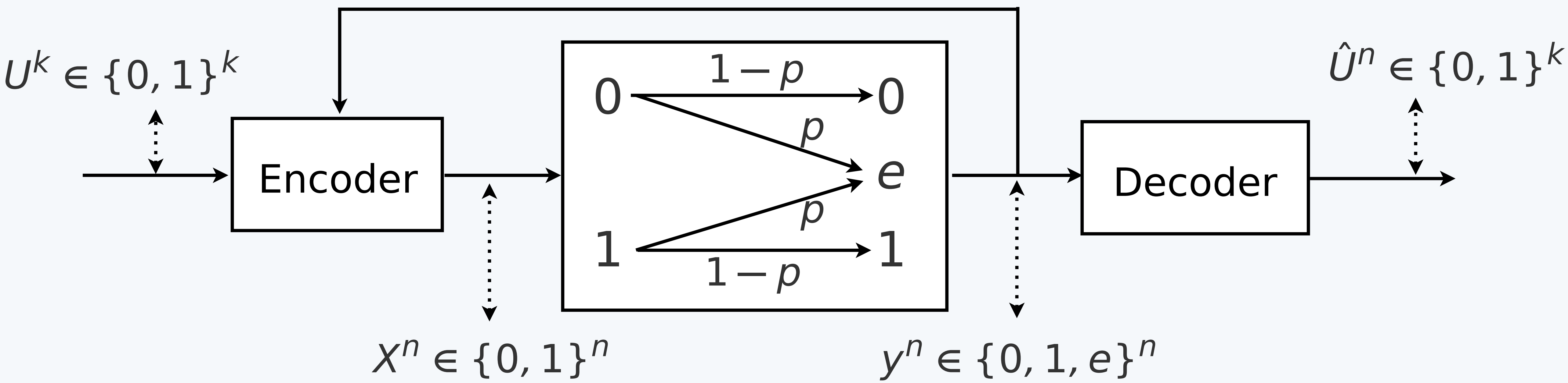


- Scheme: repeatedly transmit each bit until it gets through

$$P_e = \sum_{\ell=0}^{k-1} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot \left(1 - \frac{1}{2^{k-\ell}}\right)$$

$$\leq P\left(\sum_{i=1}^{n} Z_i < k\right)$$

A Non-Example



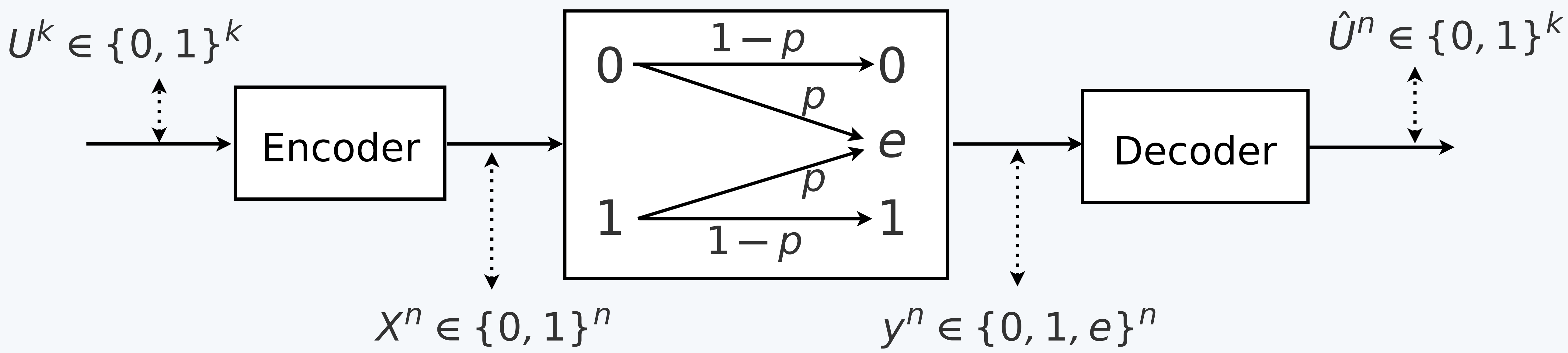
- Scheme: repeatedly transmit each bit until it gets through

$$P_e = \sum_{\ell=0}^{k-1} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot \left(1 - \frac{1}{2^{k-\ell}}\right)$$

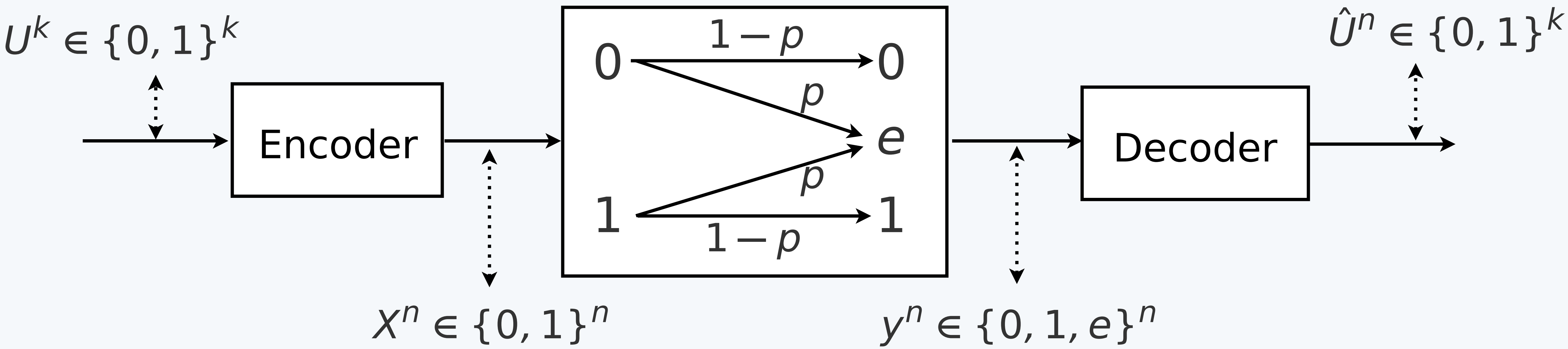
$$\leq P\left(\sum_{i=1}^{n} Z_i < k\right) \to 0$$

if $n, k \to \infty$ as $k = nR$ with $R < 1 - p$

A Non-Example

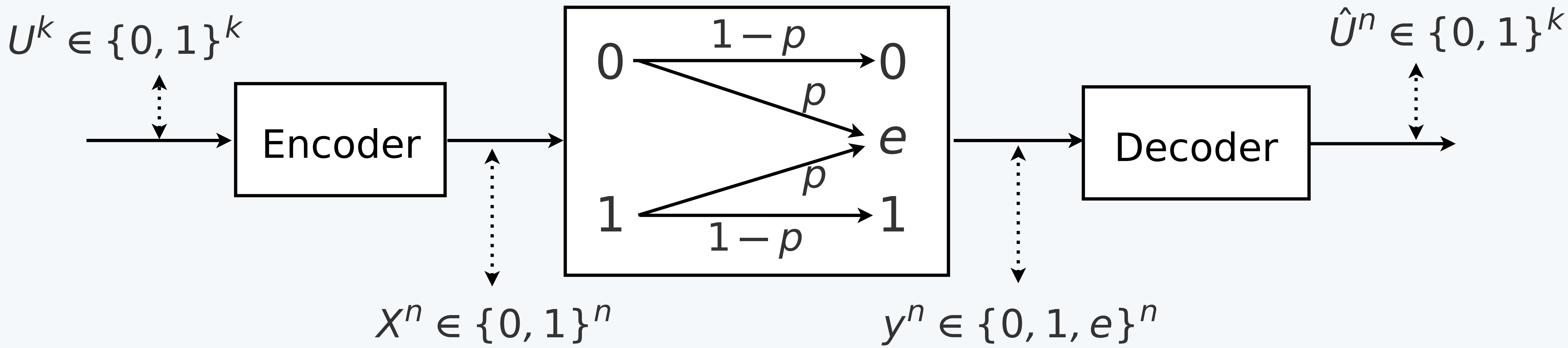


A Non-Example



$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

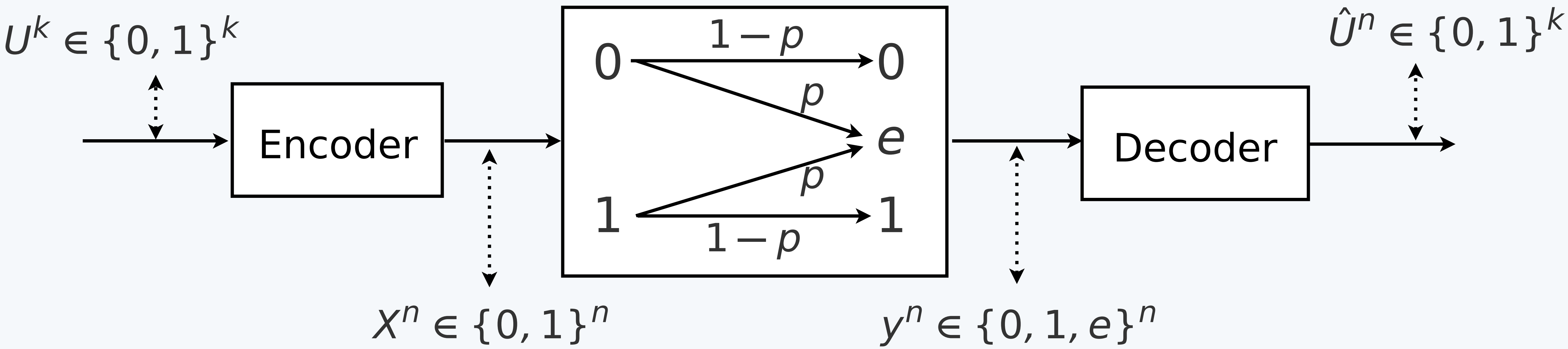
A Non-Example



$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

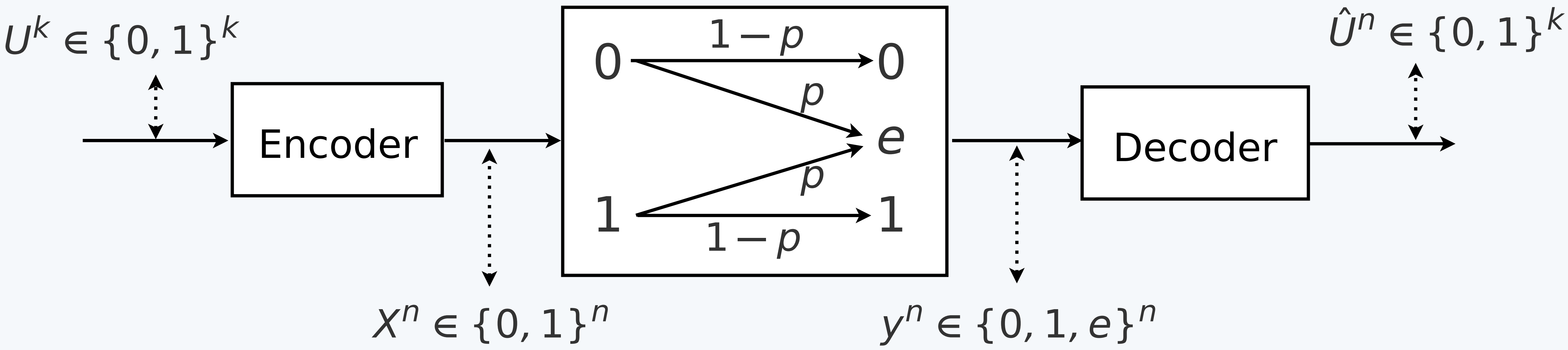
$\underline{G} \in \{0,1\}^{n \times k}$ [uniform]

A Non-Example



$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

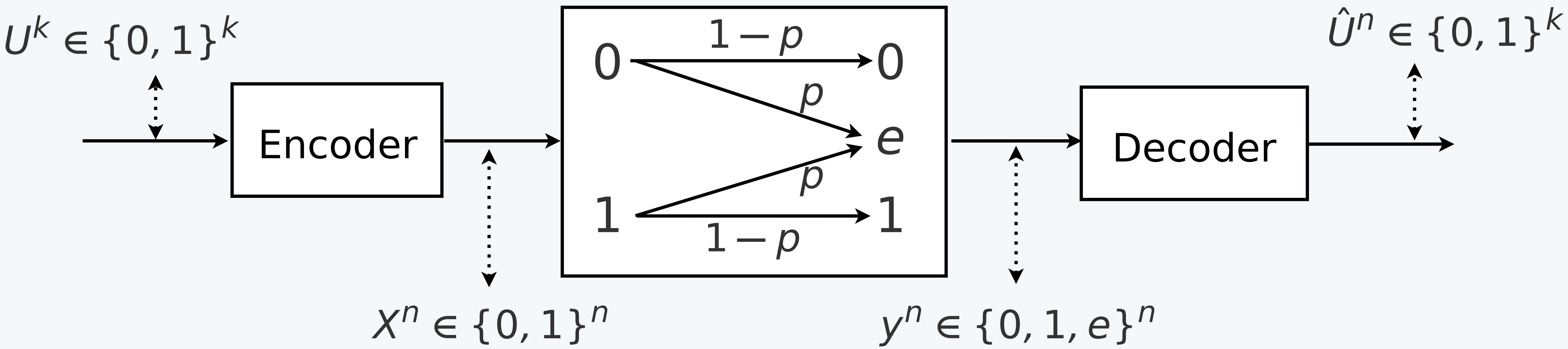
A Non-Example



$U^k \in \{0,1\}^k$

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

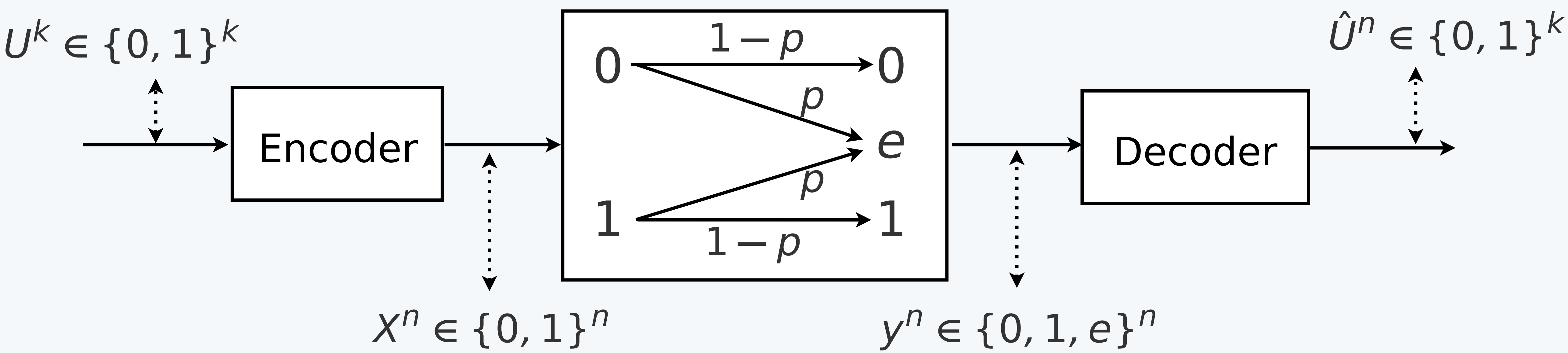
A Non-Example



$X^n \in \{0,1\}^n$

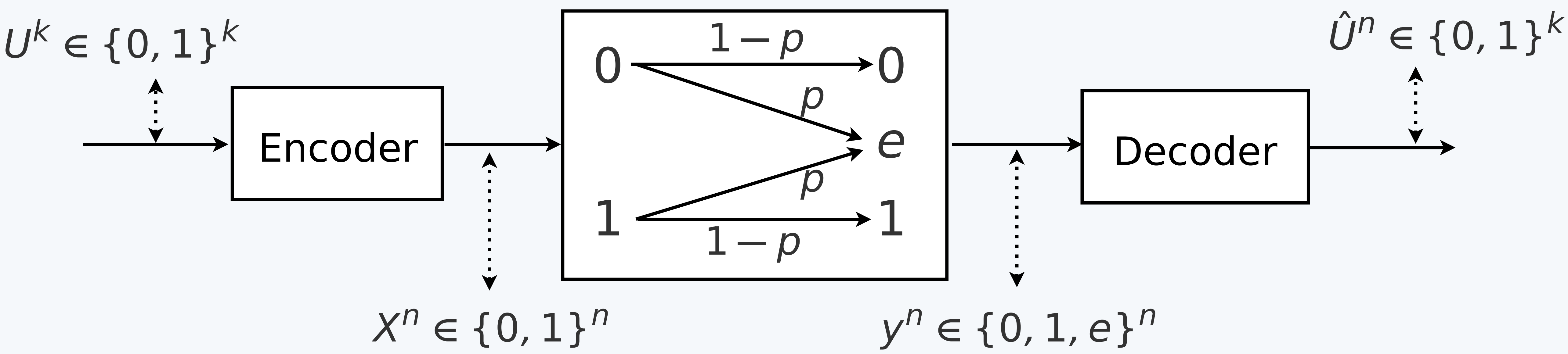
$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

A Non-Example



$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

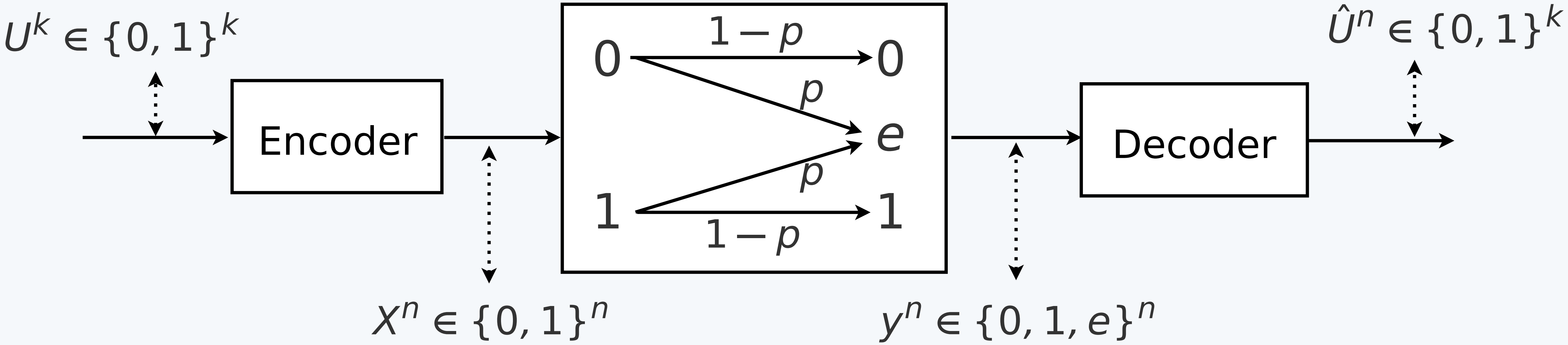
A Non-Example



$Y^n \in \{0,1,e\}^n$

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

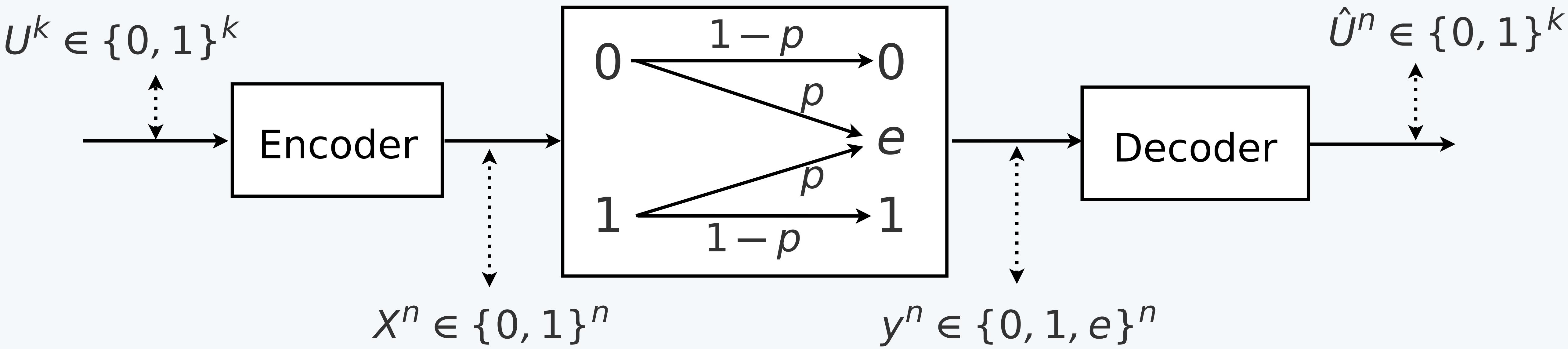
A Non-Example



$$P_e \leq \sum_{\ell=0}^{n} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot P(\ell \times k \text{ sub-matrix of } \underline{G} \text{ not full column-rank})$$

$$\leq \sum_{\ell=0}^{n} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot \max(2^{k-\ell}, 1)$$

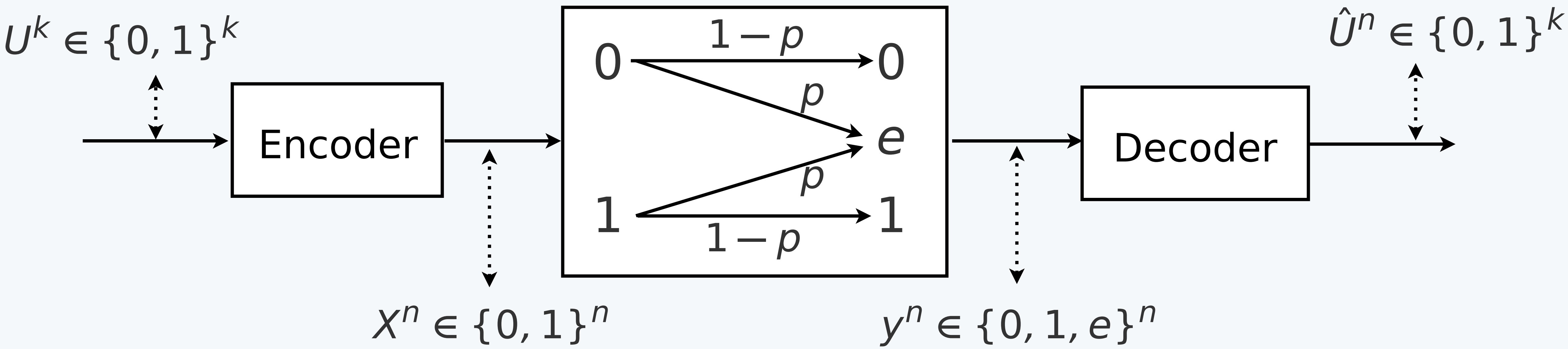
A Non-Example



$$P_e \leq \sum_{\ell=0}^{n} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot P(\ell \times k \text{ sub-matrix of } \underline{G} \text{ not full column-rank})$$

$$\leq \sum_{\ell=0}^{n} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot \max(2^{k-\ell}, 1) \to 0$$

A Non-Example

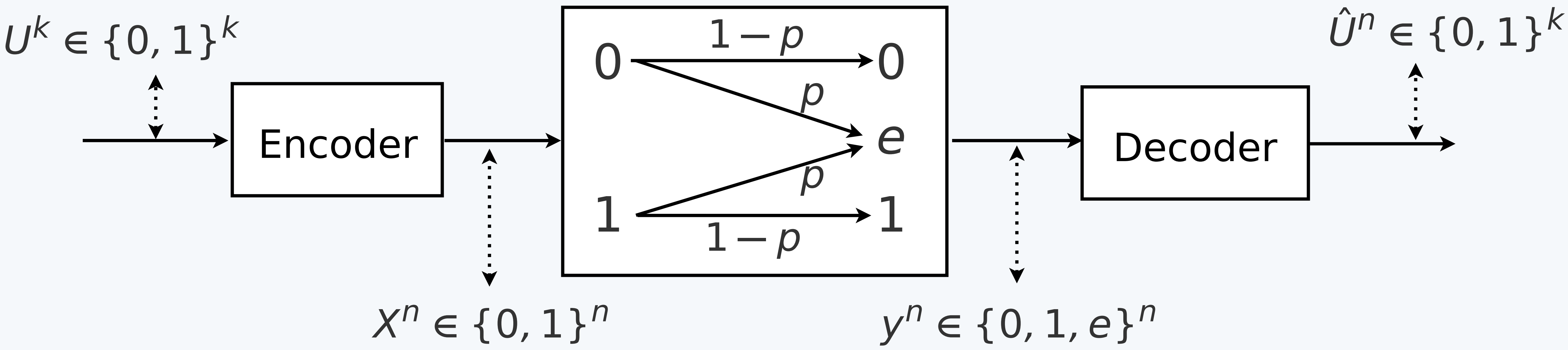


$$P_e \leq \sum_{\ell=0}^{n} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot P(\ell \times k \text{ sub-matrix of } \underline{G} \text{ not full column-rank})$$

$$\leq \sum_{\ell=0}^{n} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot \max(2^{k-\ell}, 1) \to 0$$

if $n, k \to \infty$ as $k = nR$ with $R < 1 - p$

A Non-Example



$$P_e \leq \sum_{\ell=0}^{n} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot P(\ell \times k \text{ sub-matrix of } \underline{G} \text{ not full column-rank})$$

$$\leq \sum_{\ell=0}^{n} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot \max(2^{k-\ell}, 1) \to 0$$

if $n, k \to \infty$ as $k = nR$ with $R < 1 - p$

Also no improvement in (high rate) error exponents, SOCR, or moderate deviations.

More Generally...

- **Def**: A channel (stochastic matrix) W is *symmetric* if its columns (outputs) can be partitioned so that, within each partition, the columns are permutations of each other, as are the rows.

More Generally...

- **Def**: A channel (stochastic matrix) W is *symmetric* if its columns (outputs) can be partitioned so that, within each partition, the columns are permutations of each other, as are the rows.

$$\left[\begin{array}{cc|c} 1-p & 0 & p \\ 0 & 1-p & p \end{array}\right]$$

Symmetric

More Generally...

- **Def**: A channel (stochastic matrix) W is *symmetric* if its columns (outputs) can be partitioned so that, within each partition, the columns are permutations of each other, as are the rows.

$$\left[\begin{array}{cc|c} 1-p & 0 & p \\ 0 & 1-p & p \end{array}\right]$$

Symmetric

$$\begin{bmatrix} 3/4 & 1/4 \\ 1/3 & 2/3 \end{bmatrix}$$

Not symmetric

More Generally...

More Generally...

- For symmetric channels, feedback does not improve:

More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]

More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]

More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]

More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the *third-order* coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]

More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the *third-order* coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
  - the moderate deviations performance [Altuğ-Poor-Verdú (’15)]

More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the *third-order* coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
  - the moderate deviations performance [Altuğ-Poor-Verdú (’15)]
- For asymmetric channels,

More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the *third-order* coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
  - the moderate deviations performance [Altuğ-Poor-Verdú (’15)]
- For asymmetric channels,
  - The high-rate error exponent is not improved by feedback [Nakiboğlu ’19, Augustin ’78]

More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the *third-order* coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
  - the moderate deviations performance [Altuğ-Poor-Verdú (’15)]
- For asymmetric channels,
  - The high-rate error exponent is not improved by feedback [Nakiboğlu ’19, Augustin ’78]
  - The second-order coding rate can be improved by feedback [*Part II*]

More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the *third-order* coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
  - the moderate deviations performance [Altuğ-Poor-Verdú (’15)]
- For asymmetric channels,
  - The high-rate error exponent is not improved by feedback [Nakiboğlu ’19, Augustin ’78]
  - The second-order coding rate can be improved by feedback [*Part II*]
  - Moderate deviations?

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

Channels with Memory

Channels with Memory

- First attempt at an example: consider the binary symmetric channel (BSC):

$$\mathcal{X} = \mathcal{Y} = \{0, 1\}$$

$$Y^n = X^n \oplus Z^n$$

where $\{Z_n\}$ is an arbitrary stationary and ergodic process.

Channels with Memory

- First attempt at an example: consider the binary symmetric channel (BSC):

$$\mathcal{X} = \mathcal{Y} = \{0, 1\}$$

$$Y^n = X^n \oplus Z^n$$

where $\{Z_n\}$ is an arbitrary stationary and ergodic process.

- Then $C = 1 - H(\{Z_n\})$

Channels with Memory

With feedback:

$$
\begin{aligned}
I(U^k; Y^n) &= H(Y^n) - H(Y^n|U^k) \\
&= H(Y^n) - \sum_{i=1}^{n} H(Y_i|U^k, Y^{i-1}) \\
&= H(Y^n) - \sum_{i=1}^{n} H(X_i \oplus Z_i|U^k, Y^{i-1}) \\
&= H(Y^n) - \sum_{i=1}^{n} H(Z_i|U^k, Y^{i-1}, Z^{i-1}) \\
&= H(Y^n) - \sum_{i=1}^{n} H(Z_i|Z^{i-1}) \\
&= H(Y^n) - H(Z^n) \\
&\leq n - H(Z^n)
\end{aligned}
$$

Channels with Memory

With feedback:
$$
\begin{aligned}
I(U^k;Y^n) &= H(Y^n) - H(Y^n|U^k) \\
&= H(Y^n) - \sum_{i=1}^{n} H(Y_i|U^k, Y^{i-1}) \\
&= H(Y^n) - \sum_{i=1}^{n} H(X_i \oplus Z_i|U^k, Y^{i-1}) \\
&= H(Y^n) - \sum_{i=1}^{n} H(Z_i|U^k, Y^{i-1}, Z^{i-1}) \\
&= H(Y^n) - \sum_{i=1}^{n} H(Z_i|Z^{i-1}) \\
&= H(Y^n) - H(Z^n) \\
&\le n - H(Z^n)
\end{aligned}
$$

$$C_{\text{FB}} = 1 - H(\{Z_n\})$$

Channels with Memory

With feedback:

$$
\begin{aligned}
I(U^k; Y^n) &= H(Y^n) - H(Y^n | U^k) \\
&= H(Y^n) - \sum_{i=1}^{n} H(Y_i | U^k, Y^{i-1}) \\
&= H(Y^n) - \sum_{i=1}^{n} H(X_i \oplus Z_i | U^k, Y^{i-1}) \\
&= H(Y^n) - \sum_{i=1}^{n} H(Z_i | U^k, Y^{i-1}, Z^{i-1}) \\
&= H(Y^n) - \sum_{i=1}^{n} H(Z_i | Z^{i-1}) \\
&= H(Y^n) - H(Z^n) \\
&\leq n - H(Z^n)
\end{aligned}
$$

$$C_{FB} = 1 - H(\{Z_n\})$$

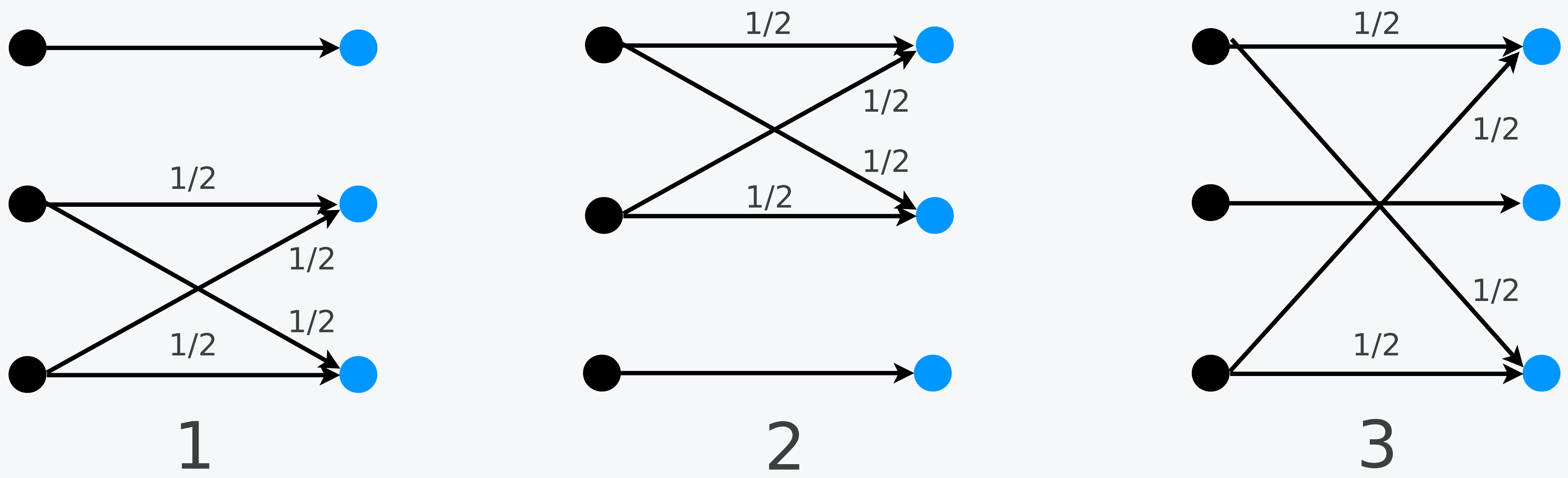
Feedback does not increase the capacity of discrete additive-noise channels

[Alajaji ('95)]

Channels with Memory

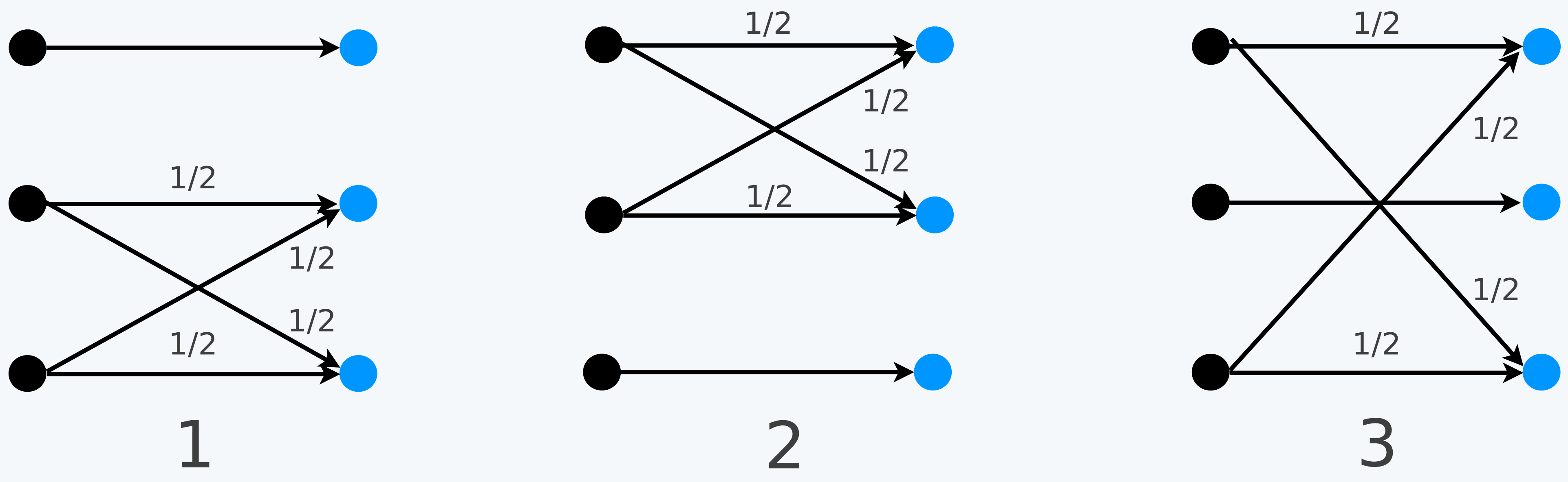
Channels with Memory

- Consider a channel with ternary channel with three "states"



Channels with Memory

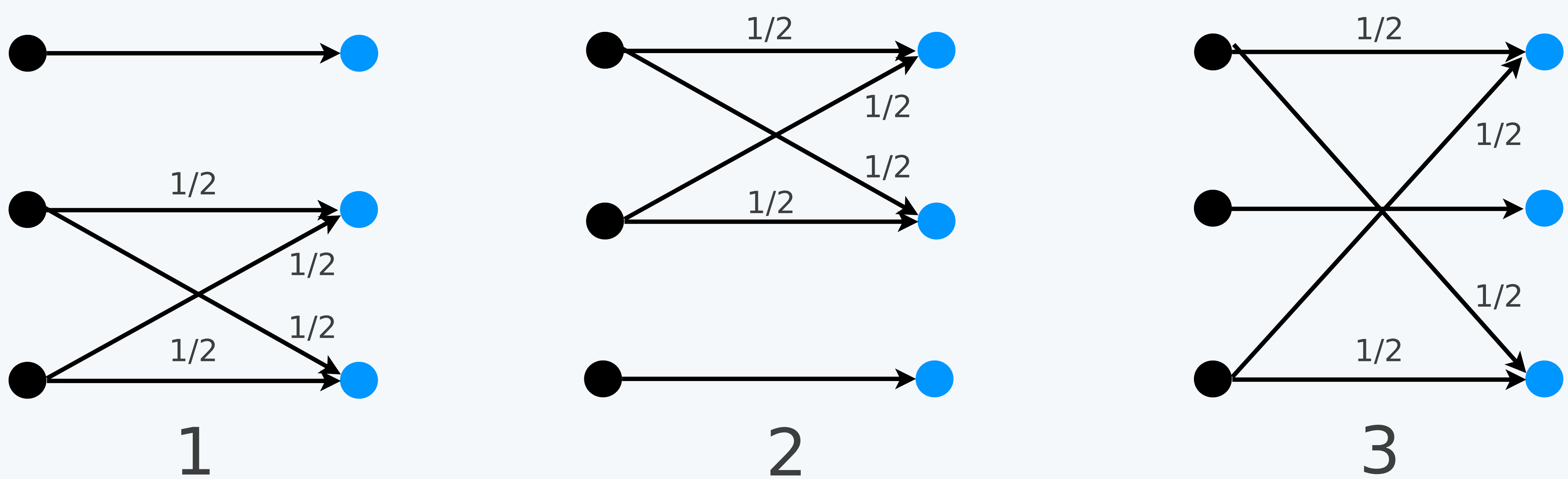
- Consider a channel with ternary channel with three "states"



- The channel starts in a random state and then deterministically cycles 1 → 2, 2 → 3, 3 → 1.

Channels with Memory

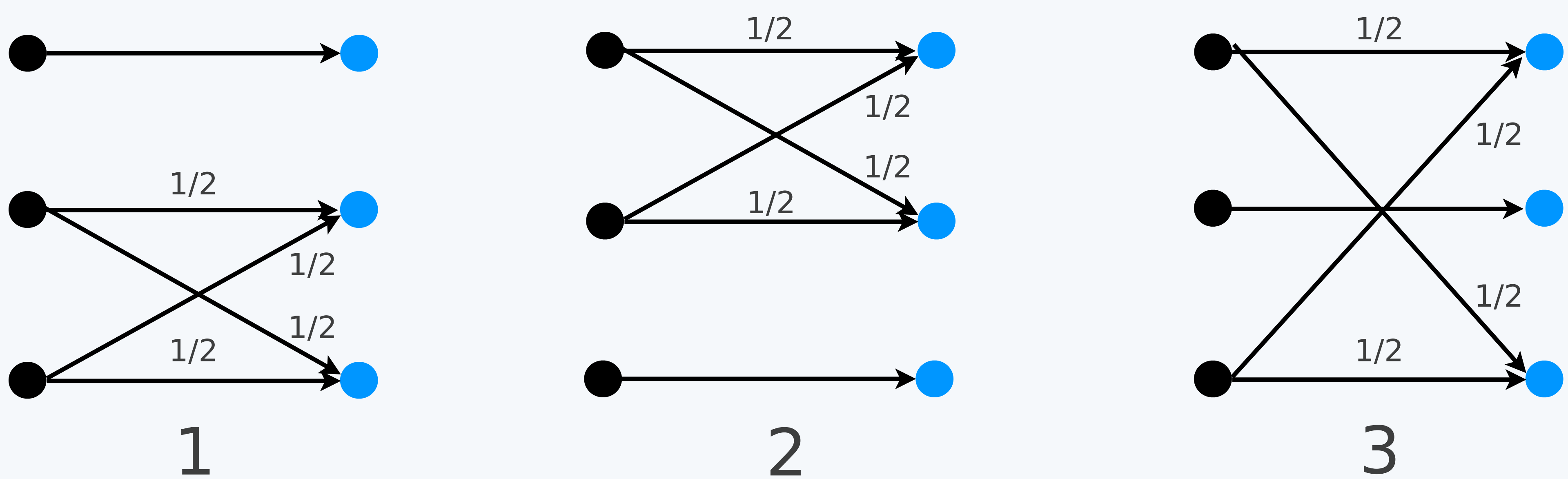
- Consider a channel with ternary channel with three "states"



- The channel starts in a random state and then deterministically cycles 1 → 2, 2 → 3, 3 → 1.
- Each constituent channel has $C = 1$ bit.

Channels with Memory

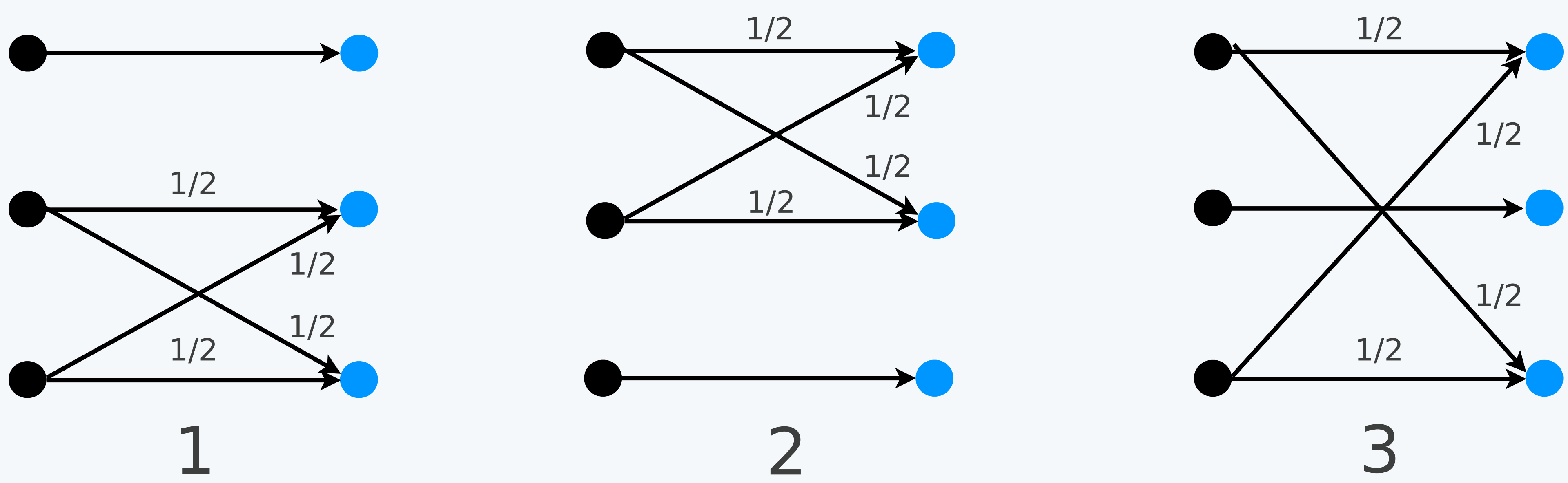
- Consider a channel with ternary channel with three "states"



- The channel starts in a random state and then deterministically cycles $1 \to 2$, $2 \to 3$, $3 \to 1$.
- Each constituent channel has $C = 1$ bit.
- With feedback, encoder can learn the phase: $C_{FB} = 1$ bit

Channels with Memory

- Consider a channel with ternary channel with three "states"



- The channel starts in a random state and then deterministically cycles 1 → 2, 2 → 3, 3 → 1.
- Each constituent channel has $C = 1$ bit.
- With feedback, encoder can learn the phase: $C_{FB} = 1$ bit
- Without feedback, encoder uses each input equally:

$$C = H(B(1/3)) < 1 \text{ bit}$$

Channels with Memory

Channels with Memory

- … are closely related to unknown channels.

Channels with Memory

- ... are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

Channels with Memory

- ... are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

$$\begin{aligned} I(U^k;Y^n) &= h(Y^n) - h(Y^n|U^k) \\ &= h(Y^n) - h(Z^n) \\ &\leq \frac{1}{2}\log((2\pi e)^n|K_{Y^n}|) - \frac{1}{2}\log((2\pi e)^n|K_{Z^n}|) \end{aligned}$$

Channels with Memory

- ... are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

$$\begin{aligned} I(U^k; Y^n) &= h(Y^n) - h(Y^n | U^k) \\ &= h(Y^n) - h(Z^n) \\ &\leq \frac{1}{2}\log((2\pi e)^n |K_{Y^n}|) - \frac{1}{2}\log((2\pi e)^n |K_{Z^n}|) \end{aligned}$$

achieved
with Gaussian
inputs

Channels with Memory

- ... are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

$$
\begin{aligned}
I(U^k;Y^n) &= h(Y^n) - h(Y^n|U^k) \\
&= h(Y^n) - h(Z^n) \\
&\le \frac{1}{2}\log((2\pi e)^n|K_{Y^n}|) - \frac{1}{2}\log((2\pi e)^n|K_{Z^n}|)
\end{aligned}
$$

independent of the input

achieved with Gaussian inputs

Channels with Memory

- … are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

$$
\begin{aligned}
I(U^k; Y^n) &= h(Y^n) - h(Y^n | U^k) \\
&= h(Y^n) - h(Z^n) \\
&\leq \frac{1}{2} \log((2\pi e)^n |K_{Y^n}|) - \frac{1}{2} \log((2\pi e)^n |K_{Z^n}|)
\end{aligned}
$$

independent of the input

achieved with Gaussian inputs

can be better whitened with feedback

Channels with Memory

- ... are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

$$
\begin{aligned}
I(U^k; Y^n) &= h(Y^n) - h(Y^n | U^k) \\
&= h(Y^n) - h(Z^n) \\
&\leq \frac{1}{2} \log((2\pi e)^n |K_{Y^n}|) - \frac{1}{2} \log((2\pi e)^n |K_{Z^n}|)
\end{aligned}
$$

independent of the input

achieved with Gaussian inputs

can be better whitened with feedback

- ARMA(k) Gaussian feedback capacity found by Kim ('10)

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

Opportunistic Use of the Channel

Opportunistic Use of the Channel

- Up to now, # of channel uses has been fixed.

Opportunistic Use of the Channel

- Up to now, # of channel uses has been fixed.
- For some transmissions, we might wish we had more. For others, we could do with fewer.

Opportunistic Use of the Channel

- Up to now, # of channel uses has been fixed.
- For some transmissions, we might wish we had more. For others, we could do with fewer.
- Suppose the transmission ends at a random (stopping) time N.

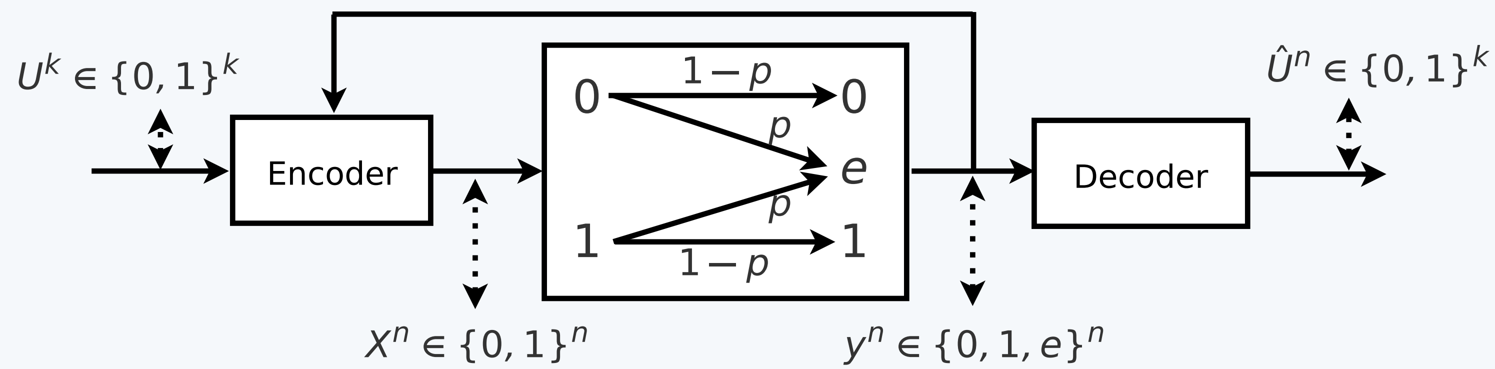
Opportunistic Use of the Channel

- Up to now, # of channel uses has been fixed.
- For some transmissions, we might wish we had more. For others, we could do with fewer.
- Suppose the transmission ends at a random (stopping) time N.
- Define the effective rate $k/E[N]$.

Opportunistic Use of the Channel

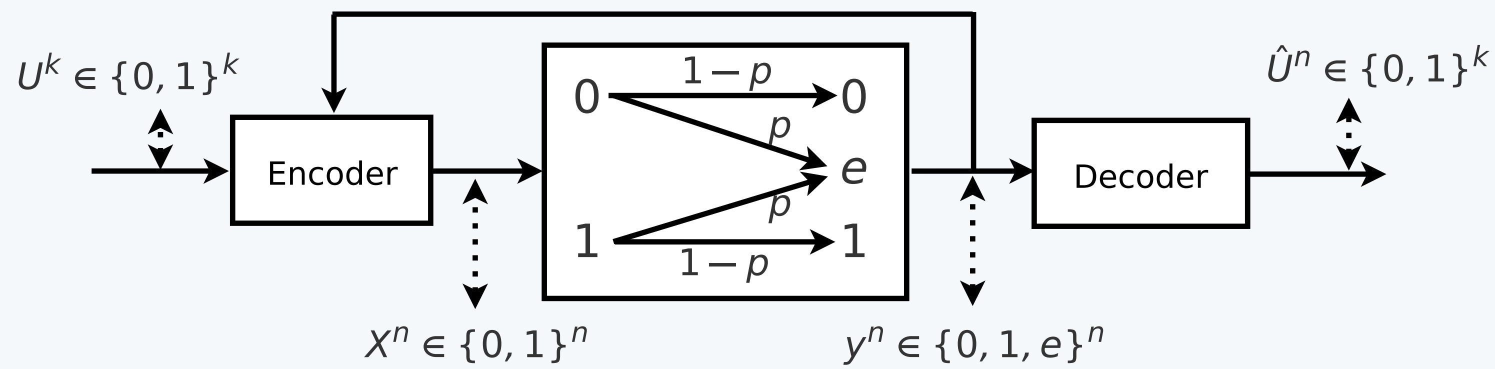
Opportunistic Use of the Channel

- Consider the BEC:



Opportunistic Use of the Channel

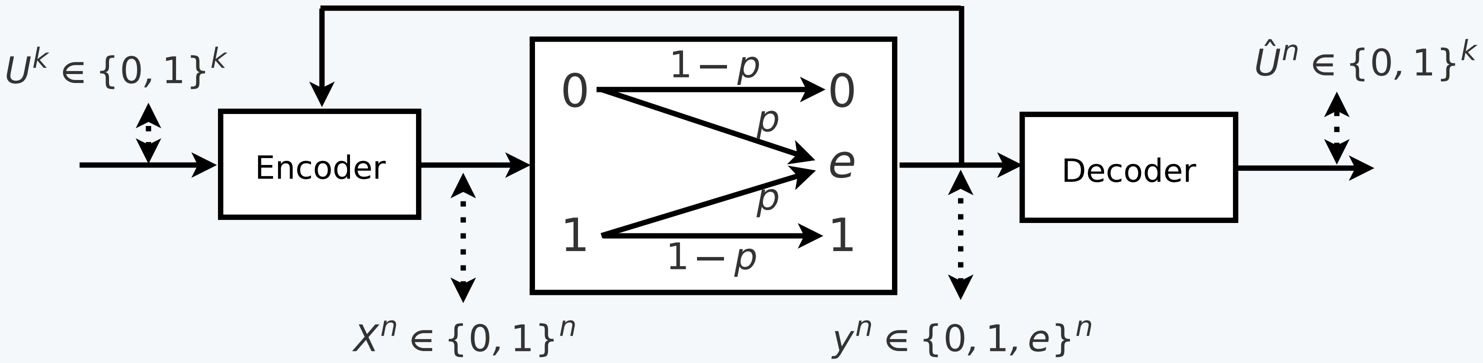
- Consider the BEC:



- Suppose we transmit each bit until it passes through.

Opportunistic Use of the Channel

- Consider the BEC:

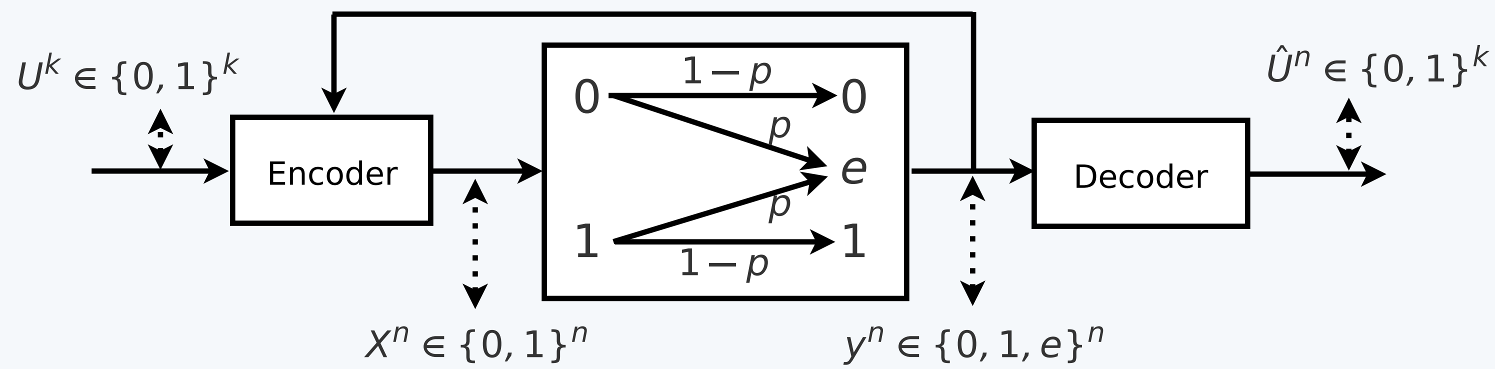


- Suppose we transmit each bit until it passes through.
- Let N be the # channel uses required for all bits to pass through. Then $E[N] = k/(1-p)$. So

$$\frac{k}{E[N]} = 1 - p = C = C_{\text{FB}}$$

Opportunistic Use of the Channel

- Consider the BEC:



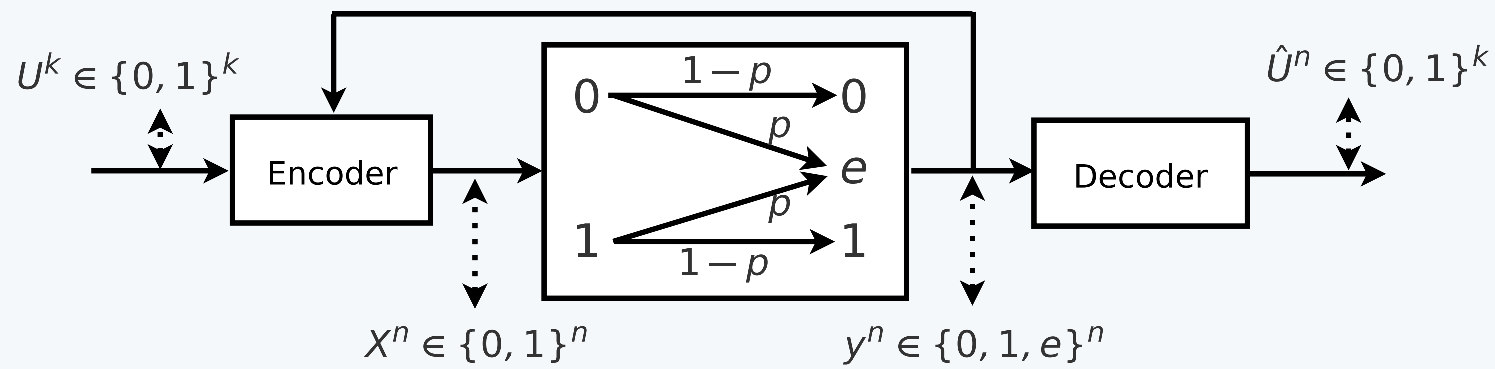
- Suppose we transmit each bit until it passes through.
- Let N be the # channel uses required for all bits to pass through. Then $E[N] = k/(1-p)$. So

$$\frac{k}{E[N]} = 1 - p = C = C_{\text{FB}}$$

$$P_e = 0$$

Opportunistic Use of the Channel

- Consider the BEC:



- Suppose we transmit each bit until it passes through.
- Let N be the # channel uses required for all bits to pass through. Then $E[N] = k/(1-p)$. So

$$\frac{k}{E[N]} = 1 - p = C = C_{\mathrm{FB}}$$

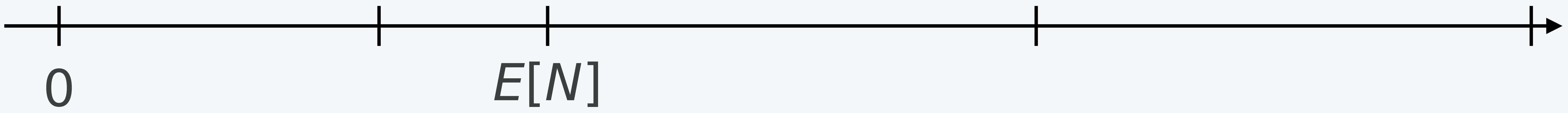
$$P_e = 0$$

- A little opportunism goes a long way:

$$\lim_{n \to \infty} \Pr(N \geq (1+\epsilon)E[N]) = 0 \text{ for any } \epsilon > 0.$$

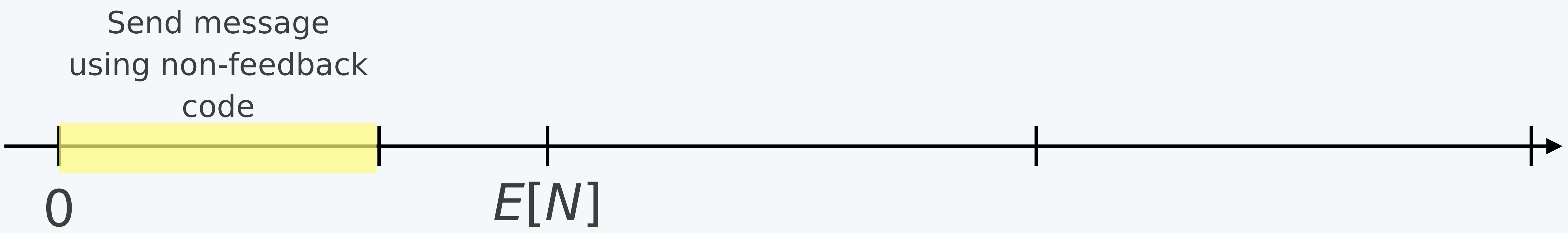
A General Scheme

- Following Burnashev ('76), reflecting later refinements:



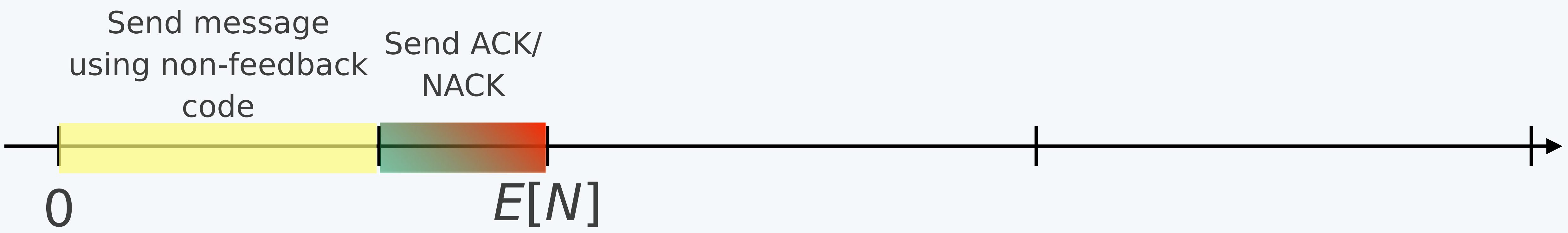
A General Scheme

- Following Burnashev ('76), reflecting later refinements:



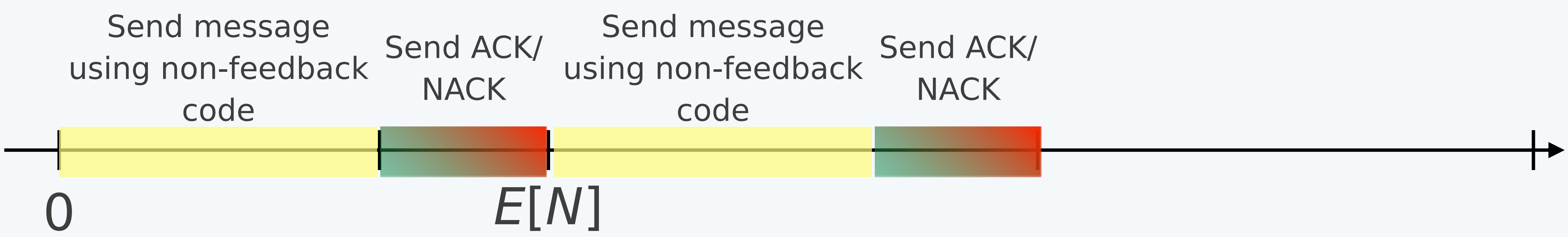
A General Scheme

- Following Burnashev ('76), reflecting later refinements:



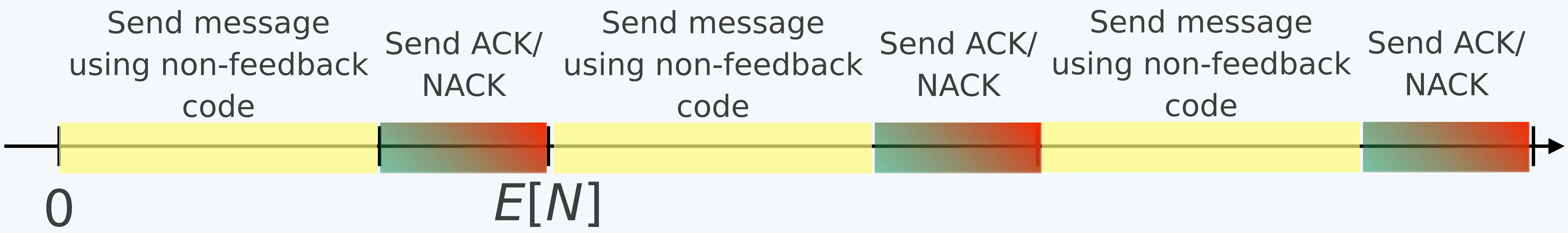
A General Scheme

- Following Burnashev ('76), reflecting later refinements:



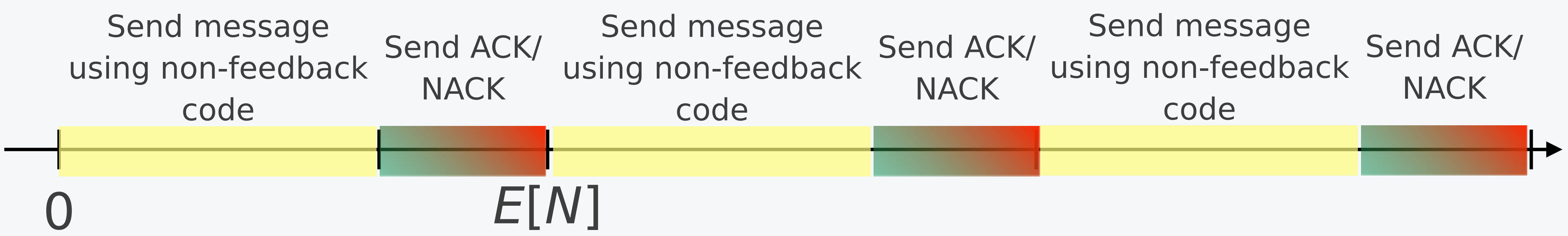
A General Scheme

- Following Burnashev ('76), reflecting later refinements:



A General Scheme

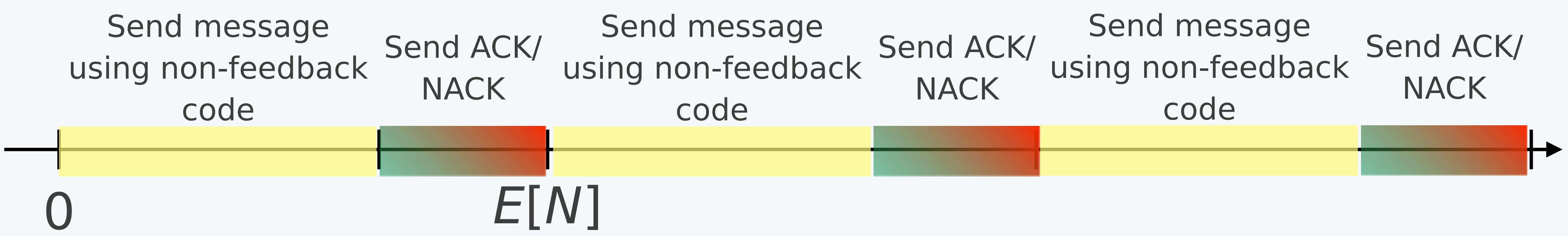
- Following Burnashev ('76), reflecting later refinements:



- Error exponent determined by Burnashev ('76)

A General Scheme

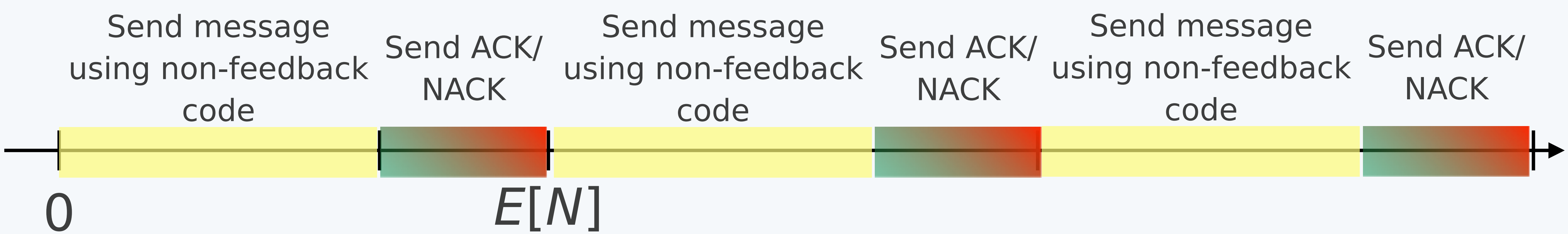
- Following Burnashev ('76), reflecting later refinements:



- Error exponent determined by Burnashev ('76)
  - Typically beats non-feedback error exponent at all rates

A General Scheme

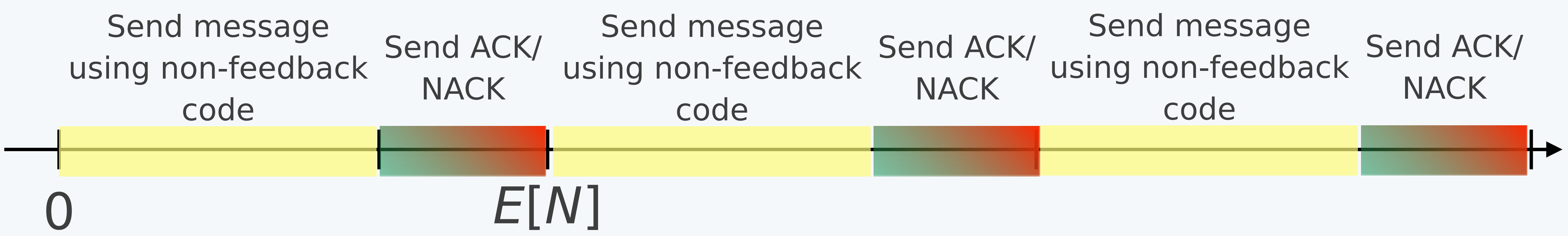
- Following Burnashev ('76), reflecting later refinements:



- Error exponent determined by Burnashev ('76)
  - Typically beats non-feedback error exponent at all rates
- Feedback provides an order improvement in

A General Scheme

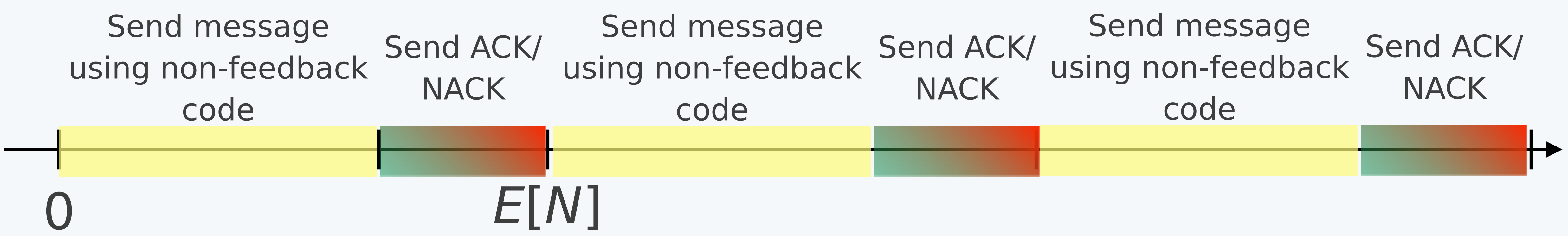
- Following Burnashev ('76), reflecting later refinements:



- Error exponent determined by Burnashev ('76)
  - Typically beats non-feedback error exponent at all rates
- Feedback provides an order improvement in
  - the moderate deviations regime [Truong and Tan ('19)]

A General Scheme

- Following Burnashev ('76), reflecting later refinements:



- Error exponent determined by Burnashev ('76)
  - Typically beats non-feedback error exponent at all rates
- Feedback provides an order improvement in
  - the moderate deviations regime [Truong and Tan ('19)]
  - the second-order coding rate regime [Polyanskiy *et al.* ('11)]

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

Opportunistic Use of Power

- Consider the AWGN

$$Y^n = X^n + Z^n \qquad\qquad Z^n \text{ i.i.d. } \mathcal{N}(0,1)$$

- Power constraint:

$$E\left[\frac{1}{n}\sum_{i=1}^{n} X_i^2(u^k, Y^{i-1})\right] \leq P \quad \text{for all messages } u^k$$

Schalkwijk-Kailath ('66) Scheme

Schalkwijk-Kailath ('66) Scheme

- Partition $\left[-\sqrt{P}, \sqrt{P}\right]$ into 2^k equal-sized intervals.

Schalkwijk-Kailath ('66) Scheme

- Partition $\left[-\sqrt{P}, \sqrt{P}\right]$ into 2^k equal-sized intervals.
- Assign each message string to one of the intervals

Schalkwijk-Kailath ('66) Scheme

- Partition $\left[-\sqrt{P}, \sqrt{P}\right]$ into 2^k equal-sized intervals.
- Assign each message string to one of the intervals
- Let $\theta(u^k)$ be the midpoint of the interval for string u^k

Schalkwijk-Kailath ('66) Scheme

- Partition $\left[-\sqrt{P}, \sqrt{P}\right]$ into 2^k equal-sized intervals.
- Assign each message string to one of the intervals
- Let $\theta(u^k)$ be the midpoint of the interval for string u^k
- Encoding:

Schalkwijk-Kailath ('66) Scheme

- Partition $\left[-\sqrt{P}, \sqrt{P}\right]$ into 2^k equal-sized intervals.
- Assign each message string to one of the intervals
- Let $\theta(u^k)$ be the midpoint of the interval for string u^k
- Encoding:
  - Time 1: $\theta(U^k)$

Schalkwijk-Kailath ('66) Scheme

- Partition $\left[-\sqrt{P}, \sqrt{P}\right]$ into 2^k equal-sized intervals.
- Assign each message string to one of the intervals
- Let $\theta(u^k)$ be the midpoint of the interval for string u^k
- Encoding:
  - Time 1: $\theta(U^k)$
  - Time j: Send $\gamma_j\left(\theta(U^k) - E[\theta(U^k)|Y^{j-1}]\right)$, where γ_j is chosen so that

$$E\left[\gamma_j^2\left(\theta(U^k) - E[\theta(U^k)|Y^{j-1}]\right)^2 \middle| U^k\right] \le P \quad \text{a.s.}$$

Schalkwijk-Kailath ('66) Scheme

- Partition $\left[-\sqrt{P}, \sqrt{P}\right]$ into 2^k equal-sized intervals.
- Assign each message string to one of the intervals
- Let $\theta(u^k)$ be the midpoint of the interval for string u^k
- Encoding:
  - Time 1: $\theta(U^k)$
  - Time j: Send $\gamma_j \left(\theta(U^k) - E[\theta(U^k)|Y^{j-1}]\right)$, where γ_j is chosen so that

$$E\left[\gamma_j^2 \left(\theta(U^k) - E[\theta(U^k)|Y^{j-1}]\right)^2 \middle| U^k\right] \leq P \quad \text{a.s.}$$

- Decoding: output string whose interval contains $E[\theta(U^k)|Y^n]$.

Schalkwijk-Kailath ('66) Scheme

- Partition $\left[-\sqrt{P}, \sqrt{P}\right]$ into 2^k equal-sized intervals.
- Assign each message string to one of the intervals
- Let $\theta(u^k)$ be the midpoint of the interval for string u^k
- Encoding:
  - Time 1: $\theta(U^k)$
  - Time j: Send $\gamma_j\left(\theta(U^k) - E[\theta(U^k)|Y^{j-1}]\right)$, where γ_j is chosen so that

$$E\left[\gamma_j^2\left(\theta(U^k) - E[\theta(U^k)|Y^{j-1}]\right)^2 \middle| U^k\right] \leq P \quad \text{a.s.}$$

- Decoding: output string whose interval contains $E[\theta(U^k)|Y^n]$.
- Performance:

$$P_e \leq \sqrt{\frac{2}{\pi}} e^{-\frac{2^{2n(C-R)}\sqrt{P}}{2}}$$

Schalkwijk-Kailath ('66) Scheme

- Partition $\left[-\sqrt{P}, \sqrt{P}\right]$ into 2^k equal-sized intervals.
- Assign each message string to one of the intervals
- Let $\theta(u^k)$ be the midpoint of the interval for string u^k
- Encoding:
  - Time 1: $\theta(U^k)$
  - Time j: Send $\gamma_j\left(\theta(U^k) - E[\theta(U^k)|Y^{j-1}]\right)$, where γ_j is chosen so that

$$E\left[\gamma_j^2\left(\theta(U^k) - E[\theta(U^k)|Y^{j-1}]\right)^2\middle|U^k\right] \leq P \quad \text{a.s.}$$

- Decoding: output string whose interval contains $E[\theta(U^k)|Y^n]$.
- Performance:

$$P_e \leq \sqrt{\frac{2}{\pi}}e^{-\frac{2^{2n(C-R)}\sqrt{P}}{2}} \qquad [!!]$$

Notes on the SK Scheme

Notes on the SK Scheme

- The Schalkwijk-Kailath scheme uses (a lot) more power when decoding errors are imminent:

$$E\left[\gamma_j^2\left(\theta(U^k)-E[\theta(U^k)|Y^{j-1}]\right)^2\middle|U^k\right]\leq P\quad \text{a.s.}$$

Notes on the SK Scheme

- The Schalkwijk-Kailath scheme uses (a lot) more power when decoding errors are imminent:

$$E\left[\gamma_j^2\left(\theta(U^k) - E[\theta(U^k)|Y^{j-1}]\right)^2 \middle| U^k\right] \leq P \quad \text{a.s.}$$

- Performance is much degraded if the power constraint is imposed a.s. [Pinkser ('68), Shepp *et al.* ('69), Altuğ-Poor-Verdú ('15)]

Notes on the SK Scheme

- The Schalkwijk-Kailath scheme uses (a lot) more power when decoding errors are imminent:

$$E\left[\gamma_j^2\left(\theta(U^k) - E[\theta(U^k)|Y^{j-1}]\right)^2 \middle| U^k\right] \leq P \quad \text{a.s.}$$

- Performance is much degraded if the power constraint is imposed a.s. [Pinkser ('68), Shepp *et al.* ('69), Altuğ-Poor-Verdú ('15)]
- Error exponent of fixed-length coding for DMCs with a cost constraint?

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

Min. Distance Example

Min. Distance Example

- Consider the binary symmetric channel, w/o feedback,

$$Y^n = X^n \oplus Z^n \quad Z^n \text{ i.i.d. } B(p)$$

  and at low rate, $k = \epsilon n, \epsilon \approx 0.$ Then P_e is exp. small.

Min. Distance Example

- Consider the binary symmetric channel, w/o feedback,

$$Y^n = X^n \oplus Z^n \quad Z^n \text{ i.i.d. } B(p)$$

  and at low rate, $k = \epsilon n, \epsilon \approx 0.$ Then P_e is exp. small.
- Suppose the codewords are

$$x_1^n, x_2^n, \ldots, x_{2^k}^n$$

Min. Distance Example

- Consider the binary symmetric channel, w/o feedback,

$$Y^n = X^n \oplus Z^n \quad Z^n \text{ i.i.d. } B(p)$$

and at low rate, $k = \epsilon n, \epsilon \approx 0.$ Then P_e is exp. small.

- Suppose the codewords are

$$x_1^n, x_2^n, \ldots, x_{2^k}^n$$

- ML decoding rule

$$\text{argmin}_i \; d_H(x_i^n, Y^n)$$

Min. Distance Example

- Consider the binary symmetric channel, w/o feedback,

$$Y^n = X^n \oplus Z^n \quad Z^n \text{ i.i.d. } B(p)$$

  and at low rate, $k = \epsilon n, \epsilon \approx 0.$ Then P_e is exp. small.
- Suppose the codewords are

$$x_1^n, x_2^n, \ldots, x_{2^k}^n$$

- ML decoding rule

$$\text{argmin}_i \, d_H(x_i^n, Y^n)$$

Hamming distance

Min. Distance Example

$$P_e = 2^{-k} \sum_{m=1}^{2^k} \Pr\left(\text{error} \middle| x_m^n\right)$$

Min. Distance Example

$$P_e = 2^{-k} \sum_{m=1}^{2^k} \Pr\left(\text{error} \middle| x_m^n\right)$$

$$\leq 2^{-k} \sum_{m=1}^{2^k} \sum_{\ell=1, \ell \neq m}^{2^k} \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

Min. Distance Example

$$P_e = 2^{-k} \sum_{m=1}^{2^k} \Pr\left(\text{error} \middle| x_m^n\right)$$

$$\leq 2^{-k} \sum_{m=1}^{2^k} \sum_{\ell=1,\ell\neq m}^{2^k} \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

$$\leq 2^k \cdot \max_{\ell\neq m} \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

Min. Distance Example

$$P_e = 2^{-k} \sum_{m=1}^{2^k} \Pr\left(\text{error} \middle| x_m^n\right)$$

$$\leq 2^{-k} \sum_{m=1}^{2^k} \sum_{\ell=1,\ell\neq m}^{2^k} \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

$$\leq 2^k \cdot \max_{\ell\neq m} \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

$$P_e \geq 2^{-k} \cdot \max_{\ell\neq m} \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

Min. Distance Example

$$P_e = 2^{-k} \sum_{m=1}^{2^k} \Pr\left(\text{error} \middle| x_m^n\right)$$

$$\leq 2^{-k} \sum_{m=1}^{2^k} \sum_{\ell=1,\ell\neq m}^{2^k} \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

$$\leq 2^k \cdot \max_{\ell\neq m} \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

$$P_e \geq 2^{-k} \cdot \max_{\ell\neq m} \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

$$\frac{1}{n} \log P_e \approx \max_{\ell\neq m} \frac{1}{n} \log \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

Min. Distance Example

$$\frac{1}{n}\log P_e \approx \max_{\ell \neq m} \frac{1}{n}\log \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

Min. Distance Example

$$\frac{1}{n}\log P_e \approx \max_{\ell \neq m} \frac{1}{n}\log \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

$$\approx \exp\left(-d_H(x_\ell^n, x_m^n) \cdot D\left(\frac{1}{2}\middle\|p\right)\right)$$

Min. Distance Example

$$\frac{1}{n}\log P_e \approx \max_{\ell \neq m} \frac{1}{n}\log \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n\right)$$

$$\approx \exp\left(-d_H(x_\ell^n, x_m^n) \cdot D\left(\frac{1}{2}\middle\|p\right)\right)$$

So

$$-\frac{1}{n}\log P_e \approx \min_{\ell \neq m} \frac{d_H(x_\ell^n, x_m^n)}{n} \cdot D\left(\frac{1}{2}\middle\|p\right)$$

Min. Distance Example

$$\frac{1}{n}\log P_e \approx \max_{\ell\neq m}\frac{1}{n}\log \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n)\middle| x_m^n\right)$$

$$\approx \exp\left(-d_H(x_\ell^n, x_m^n)\cdot D\left(\frac{1}{2}\middle\|p\right)\right)$$

So

$$-\frac{1}{n}\log P_e \approx \min_{\ell\neq m}\frac{d_H(x_\ell^n, x_m^n)}{n}\cdot D\left(\frac{1}{2}\middle\|p\right)$$

min. distance
of the code

Min. Distance Example

$$\frac{1}{n}\log P_e \approx \max_{\ell \neq m} \frac{1}{n} \log \Pr\left(d_H(x_\ell^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n \right)$$

$$\approx \exp\left(-d_H(x_\ell^n, x_m^n) \cdot D\left(\frac{1}{2} \middle\| p \right) \right)$$

So

$$-\frac{1}{n}\log P_e \approx \min_{\ell \neq m} \frac{d_H(x_\ell^n, x_m^n)}{n} \cdot D\left(\frac{1}{2} \middle\| p \right)$$

min. distance
of the code

Q: How large can the minimum distance be?

Min. Distance Example

Min. Distance Example

- How large can the minimum distance be?

Min. Distance Example

- How large can the minimum distance be?
  - If $k = 1$, min. distance is n.

Min. Distance Example

- How large can the minimum distance be?
  - If $k = 1$, min. distance is n.
    - 000000000000 vs. 111111111111

Min. Distance Example

- How large can the minimum distance be?
  - If $k = 1$, min. distance is n.
    - 000000000000 vs. 111111111111
  - If $k/n = \epsilon$, where ϵ is small, then min. distance $\approx n/2$

Min. Distance Example

- How large can the minimum distance be?
  - If $k = 1$, min. distance is n.
    - 000000000000 vs. 111111111111
  - If $k/n = \epsilon$, where ϵ is small, then min. distance $\approx n/2$
- Suppose near the end of transmission, a genie ruled out all but one of the incorrect codewords.

Min. Distance Example

- How large can the minimum distance be?
  - If $k = 1$, min. distance is n.
    - 000000000000 vs. 111111111111
  - If $k/n = \epsilon$, where ϵ is small, then min. distance $\approx n/2$
- Suppose near the end of transmission, a genie ruled out all but one of the incorrect codewords.
  - Remaining transmission can be 0000... vs. 1111.....

Min. Distance Example

- How large can the minimum distance be?
  - If $k = 1$, min. distance is n.
    - 000000000000 vs. 111111111111
  - If $k/n = \epsilon$, where ϵ is small, then min. distance $\approx n/2$
- Suppose near the end of transmission, a genie ruled out all but one of the incorrect codewords.
  - Remaining transmission can be 0000... vs. 1111.....
  - Would yield an *effective* min. distance increase.

Min. Distance Example

- How large can the minimum distance be?
  - If $k = 1$, min. distance is n.
    - 000000000000 vs. 111111111111
  - If $k/n = \epsilon$, where ϵ is small, then min. distance $\approx n/2$
- Suppose near the end of transmission, a genie ruled out all but one of the incorrect codewords.
  - Remaining transmission can be 0000... vs. 1111.....
  - Would yield an *effective* min. distance increase.
  - We can achieve a similar effect with feedback ....

Min. Distance Example

Min. Distance Example

- Following Zigangirov ('70),

Min. Distance Example

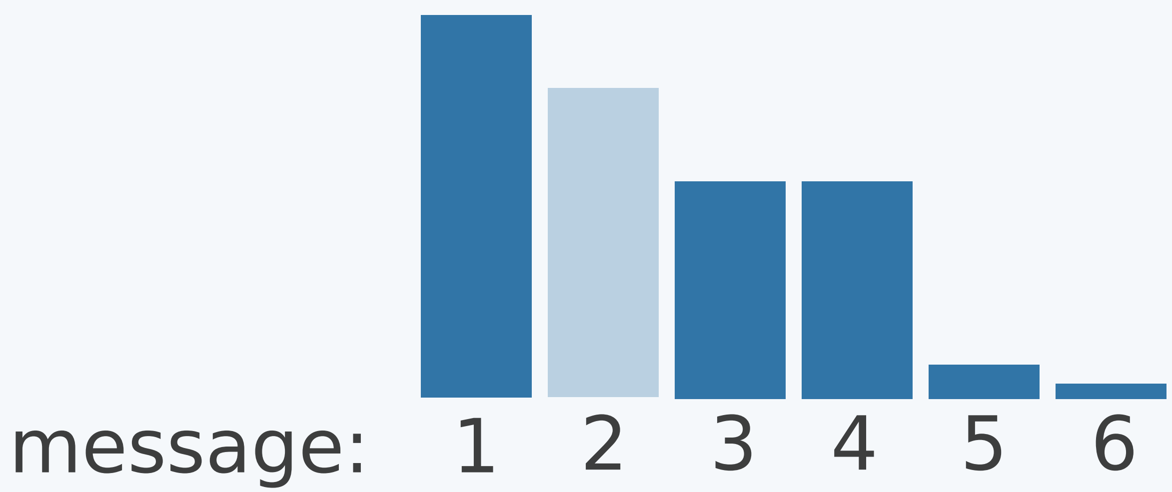
- Following Zigangirov ('70),
  - At time i, compute posterior prob. of messages given Y^{i-1}.

Min. Distance Example

- Following Zigangirov ('70),
  - At time i, compute posterior prob. of messages given Y^{i-1}.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:

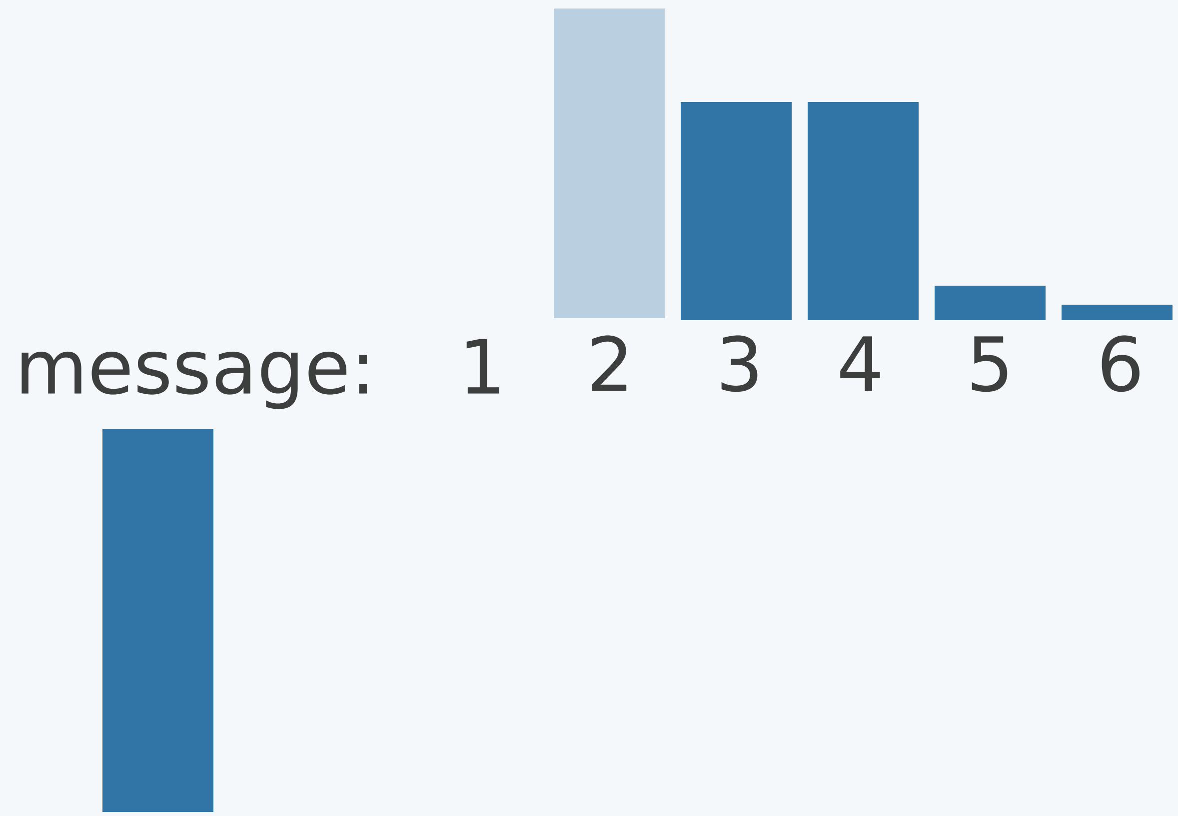
Min. Distance Example

- Following Zigangirov ('70),
  - At time *i*, compute posterior prob. of messages given Y^{i-1}.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:



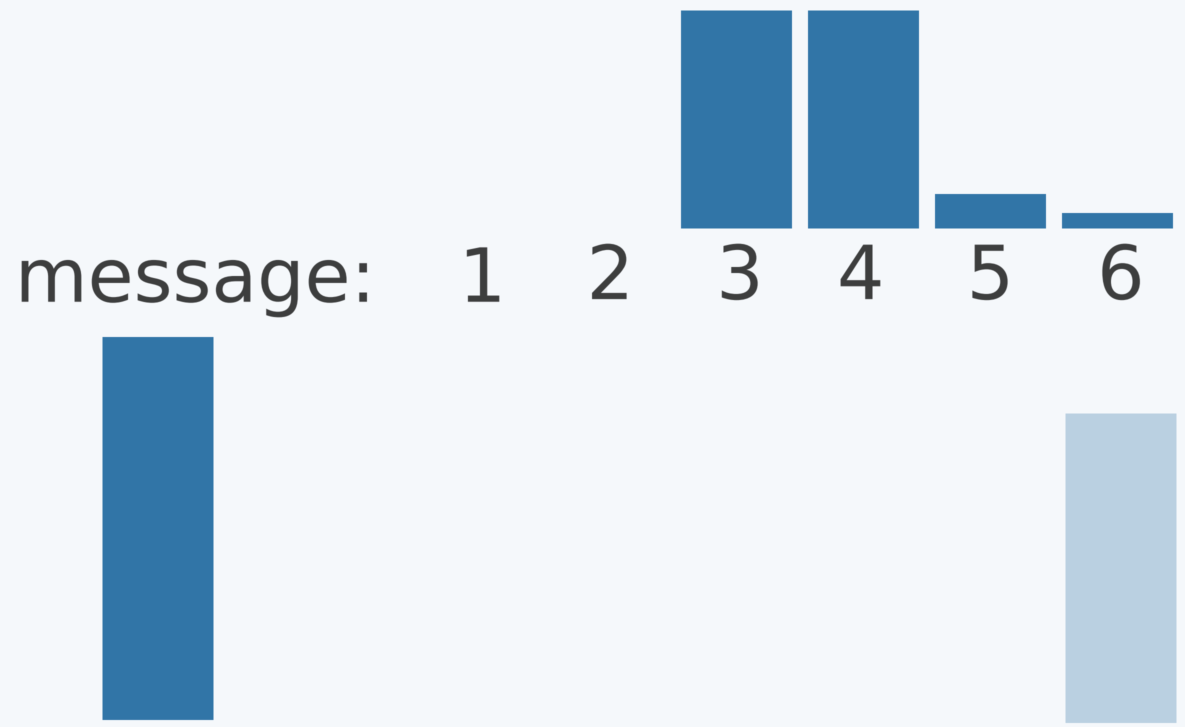
Min. Distance Example

- Following Zigangirov ('70),
  - At time *i*, compute posterior prob. of messages given Y^{i-1}.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:



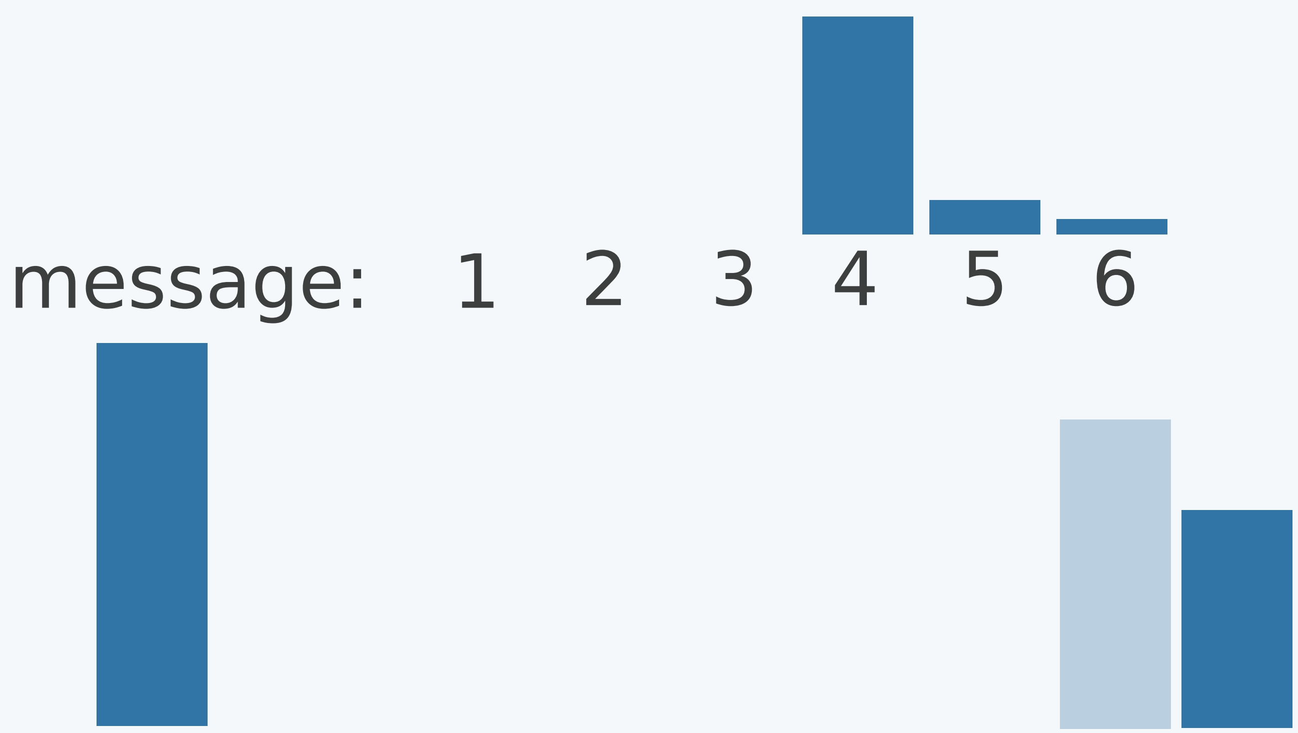
Min. Distance Example

- Following Zigangirov ('70),
  - At time *i*, compute posterior prob. of messages given Y^{i-1}.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:



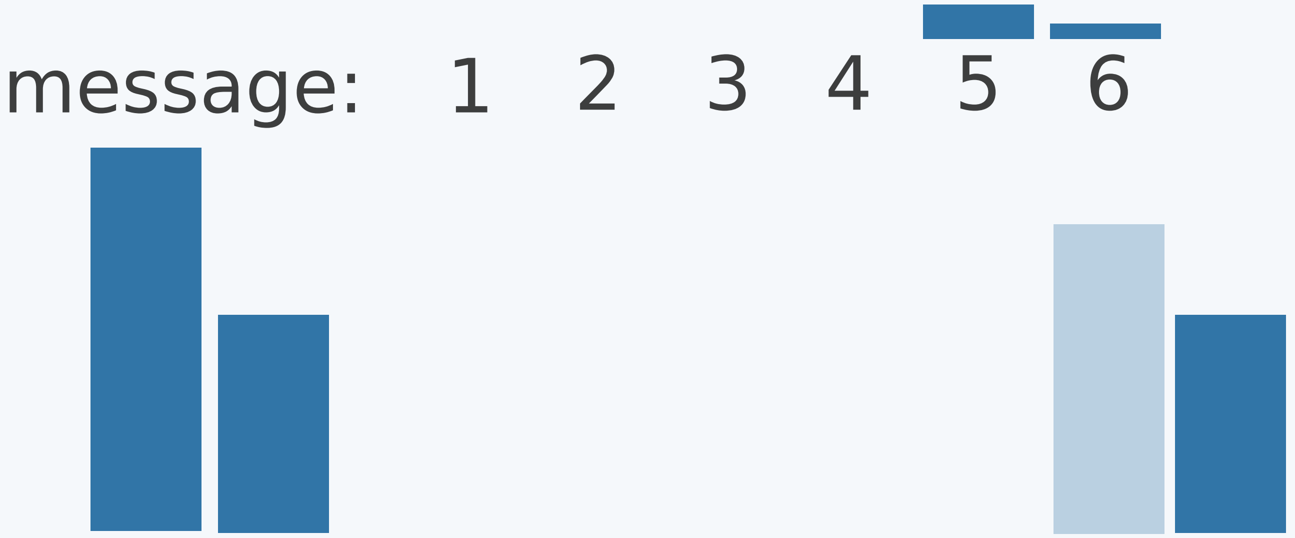
Min. Distance Example

- Following Zigangirov ('70),
  - At time *i*, compute posterior prob. of messages given $Y^{i\text{-}1}$.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:



Min. Distance Example

- Following Zigangirov ('70),
  - At time i, compute posterior prob. of messages given Y^{i-1}.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:



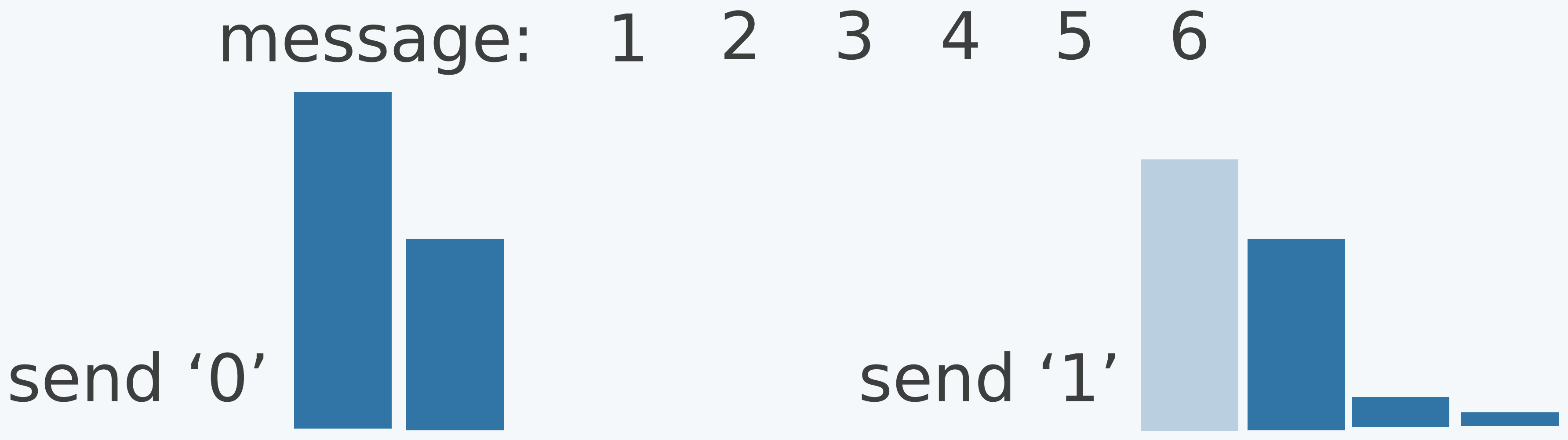
Min. Distance Example

- Following Zigangirov ('70),
  - At time i, compute posterior prob. of messages given Y^{i-1}.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:

message: 1 2 3 4 5 6

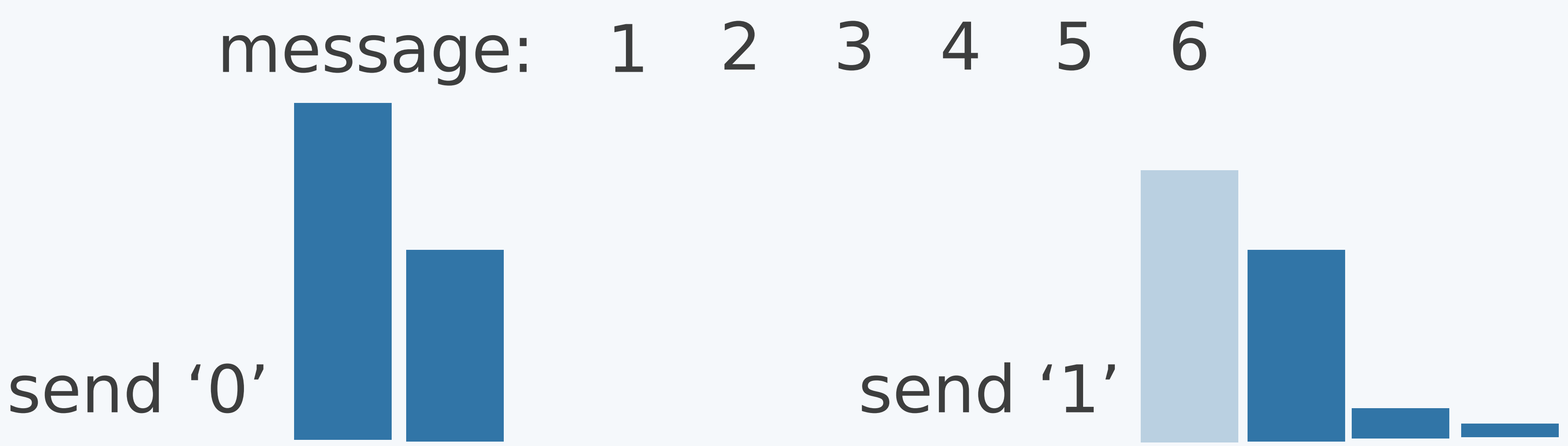
Min. Distance Example

- Following Zigangirov ('70),
  - At time *i*, compute posterior prob. of messages given Y^{i-1}.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:



Min. Distance Example

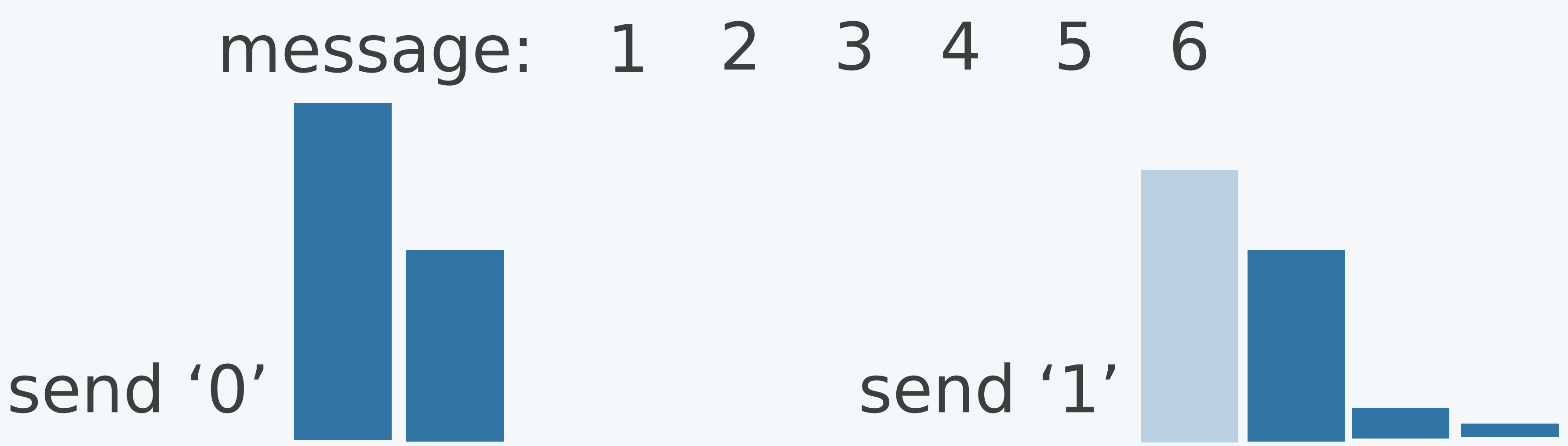
- Following Zigangirov ('70),
  - At time i, compute posterior prob. of messages given Y^{i-1}.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:



  - Improves low-rate error exponent over non-feedback case.

Min. Distance Example

- Following Zigangirov ('70),
  - At time i, compute posterior prob. of messages given Y^{i-1}.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:



- Symmetric channel: no high-rate error exponent, moderate deviations, or second-order coding rate improvement.

Mechanisms

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
  - [See *Part II*]