Zero-Error Shift-Correcting and Shift-Detecting Codes

Miloš Stojaković

Joint work with Mladen Kovačević and Vincent Y. F. Tan.

Zero-Error Communication

Zero-Error Communication

- A *code* (over alphabet $\{0,1\}$ of length n) is a non-empty subset of $\{0,1\}^n$.

Zero-Error Communication

- A *code* (over alphabet $\{0, 1\}$ of length n) is a non-empty subset of $\{0, 1\}^n$.

  Words from the code are sent over a channel (that may change them)...

Zero-Error Communication

- A *code* (over alphabet $\{0, 1\}$ of length n) is a non-empty subset of $\{0, 1\}^n$.

  Words from the code are sent over a channel (that may change them)...
- *Zero-error code* for a channel is a code with error probability equal to zero (under optimal decoding).

Zero-Error Communication

- A *code* (over alphabet $\{0,1\}$ of length n) is a non-empty subset of $\{0,1\}^n$.

  Words from the code are sent over a channel (that may change them)...
- *Zero-error code* for a channel is a code with error probability equal to zero (under optimal decoding).

  $\iff$ No two codewords can produce the same output.

Zero-Error Communication

- A *code* (over alphabet $\{0,1\}$ of length n) is a non-empty subset of $\{0,1\}^n$.

  Words from the code are sent over a channel (that may change them)...
- *Zero-error code* for a channel is a code with error probability equal to zero (under optimal decoding).

  $\iff$ No two codewords can produce the same output.
- *Zero-error capacity* of a channel, denoted C_0, is the largest rate achievable with zero-error codes:

Zero-Error Communication

- A *code* (over alphabet $\{0,1\}$ of length n) is a non-empty subset of $\{0,1\}^n$.

  Words from the code are sent over a channel (that may change them)...
- *Zero-error code* for a channel is a code with error probability equal to zero (under optimal decoding).

  $\iff$ No two codewords can produce the same output.
- *Zero-error capacity* of a channel, denoted C_0, is the largest rate achievable with zero-error codes:
  - If $M(n)$ is the size of the largest zero-error code of length n for a given channel, then

$$C_0 = \limsup_{n\to\infty} \frac{1}{n} \log M(n).$$

Zero-Error Communication

- A *code* (over alphabet $\{0,1\}$ of length n) is a non-empty subset of $\{0,1\}^n$.

  Words from the code are sent over a channel (that may change them)...
- *Zero-error code* for a channel is a code with error probability equal to zero (under optimal decoding).

  $\iff$ No two codewords can produce the same output.
- *Zero-error capacity* of a channel, denoted C_0, is the largest rate achievable with zero-error codes:
  - If $M(n)$ is the size of the largest zero-error code of length n for a given channel, then

$$C_0 = \limsup_{n\to\infty} \frac{1}{n} \log M(n).$$

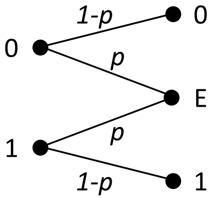
C. E. Shannon, "The Zero Error Capacity of a Noisy Channel", *IRE Trans. Inf. Theory* 2 (3), 1956.

Zero-Error Communication

- Example: The binary erasure channel (BEC)

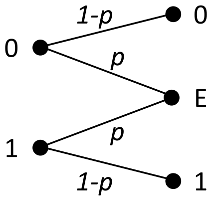
Zero-Error Communication

- Example: The binary erasure channel (BEC)



Zero-Error Communication

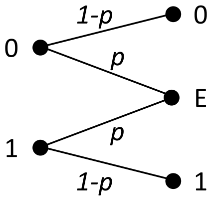
- Example: The binary erasure channel (BEC)



- Observation: No matter which codeword is sent, the sequence $\mathrm{EE}\cdots\mathrm{E}$ can be received with positive probability.

Zero-Error Communication

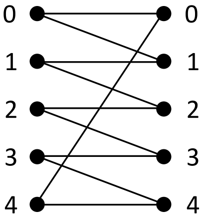
- Example: The binary erasure channel (BEC)



- Observation: No matter which codeword is sent, the sequence EE $\cdots$ E can be received with positive probability.
  - Every two codewords are confusable $\Longrightarrow$ the zero-error capacity of the BEC is equal to zero.

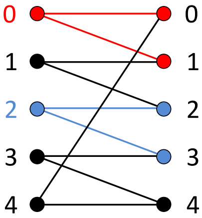
Zero-Error Communication

- Another example:



Zero-Error Communication

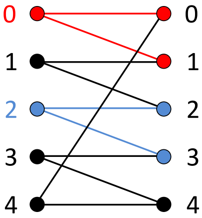
- Another example:



- Now, since the symbols 0 and 2 are not confusable, we can use only them and communicate error-free.

Zero-Error Communication

- Another example:



- Now, since the symbols 0 and 2 are not confusable, we can use only them and communicate error-free.
  - We can transmit one bit per channel use in this way.

Zero-Error Communication

- We can do better by looking at sequences of length two:

Zero-Error Communication

- We can do better by looking at sequences of length two:
  - 00, 12, 24, 31, and 43 are non-confusable.

Zero-Error Communication

- We can do better by looking at sequences of length two:
  - 00, 12, 24, 31, and 43 are non-confusable.
  - The rate of this code is $\frac{1}{2} \log 5 = \log \sqrt{5}$.

Zero-Error Communication

- We can do better by looking at sequences of length two:
  - 00, 12, 24, 31, and 43 are non-confusable.
  - The rate of this code is $\frac{1}{2}\log 5 = \log\sqrt{5}$.

- It turns out that this is the maximal possible rate, i.e., the zero-error capacity of this channel.

L. Lovasz, "On the Shannon Capacity of a Graph", *IEEE Trans. Inf. Theory* 25 (1), 1979.

The Shift Channel

The Shift Channel

- Units of transmission: *packets*

The Shift Channel

- Units of transmission: *packets*
- The basic model:

The Shift Channel

- Units of transmission: *packets*
- The basic model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;

The Shift Channel

- Units of transmission: *packets*
- The basic model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent/received in each time slot;

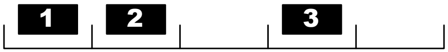
The Shift Channel

- Units of transmission: *packets*
- The basic model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent/received in each time slot;
  3) Every packet is delayed in the channel for a number of slots chosen from the set $\{0, 1, \dots, K\}$ (say, randomly);

The Shift Channel

- Units of transmission: *packets*
- The basic model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent/received in each time slot;
  3) Every packet is delayed in the channel for a number of slots chosen from the set $\{0, 1, \ldots, K\}$ (say, randomly);
  4) The packets are indistinguishable, and hence the information is conveyed via timing only.

- Units of transmission: *packets*
- The basic model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent/received in each time slot;
  3) Every packet is delayed in the channel for a number of slots chosen from the set $\{0, 1, \ldots, K\}$ (say, randomly);
  4) The packets are indistinguishable, and hence the information is conveyed via timing only.



The Shift Channel

- Units of transmission: *packets*
- The basic model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent/received in each time slot;
  3) Every packet is delayed in the channel for a number of slots chosen from the set $\{0, 1, \ldots, K\}$ (say, randomly);
  4) The packets are indistinguishable, and hence the information is conveyed via timing only.

The Shift Channel

- Units of transmission: *packets*
- The basic model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent/received in each time slot;
  3) Every packet is delayed in the channel for a number of slots chosen from the set $\{0, 1, \ldots, K\}$ (say, randomly);
  4) The packets are indistinguishable, and hence the information is conveyed via timing only.

The Shift Channel

- Units of transmission: *packets*
- The basic model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent/received in each time slot;
  3) Every packet is delayed in the channel for a number of slots chosen from the set $\{0, 1, \ldots, K\}$ (say, randomly);
  4) The packets are indistinguishable, and hence the information is conveyed via timing only.

The Shift Channel: Comments

The Shift Channel: Comments

- This model is equivalent to a discrete-time queue with bounded *residence* times.

  V. Anantharam and S. Verdu, "Bits Through Queues," *IEEE Trans. Inf. Theory* 42 (1), 1996.

The Shift Channel: Comments

- This model is equivalent to a discrete-time queue with bounded *residence* times.

  V. Anantharam and S. Verdu, "Bits Through Queues," *IEEE Trans. Inf. Theory* 42 (1), 1996.

- If the duration of transmission is n slots, the transmitted sequence of packets can be identified with a binary sequence from $\{0,1\}^n$.

The Shift Channel: Comments

- This model is equivalent to a discrete-time queue with bounded *residence* times.

  V. Anantharam and S. Verdu, "Bits Through Queues," *IEEE Trans. Inf. Theory* 42 (1), 1996.

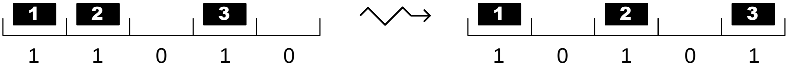
- If the duration of transmission is n slots, the transmitted sequence of packets can be identified with a binary sequence from $\{0,1\}^n$.

The Shift Channel: Comments

- This model is equivalent to a discrete-time queue with bounded *residence* times.

  V. Anantharam and S. Verdu, "Bits Through Queues," *IEEE Trans. Inf. Theory* 42 (1), 1996.

- If the duration of transmission is n slots, the transmitted sequence of packets can be identified with a binary sequence from $\{0, 1\}^n$.



The Shift Channel: Comments

- In channels with delays, packets from one codeword can interfere with the following codeword.

The Shift Channel: Comments

- In channels with delays, packets from one codeword can interfere with the following codeword.
- Example: Let $\mathcal{C} = \{000, 001, 100\}$ and $K = 1$

The Shift Channel: Comments

- In channels with delays, packets from one codeword can interfere with the following codeword.
- Example: Let $\mathcal{C} = \{000, 001, 100\}$ and $K = 1$
  - No two codewords can produce the same sequence at the output.

The Shift Channel: Comments

- In channels with delays, packets from one codeword can interfere with the following codeword.
- Example: Let $\mathcal{C} = \{000, 001, 100\}$ and $K = 1$
  - No two codewords can produce the same sequence at the output.
  - However, $001000 \rightsquigarrow 000100$ and so the sequences of codewords $001, 000$ and $000, 100$ are confusable.

The Shift Channel: Comments

- In channels with delays, packets from one codeword can interfere with the following codeword.
- Example: Let $\mathcal{C} = \{000, 001, 100\}$ and $K = 1$
  - No two codewords can produce the same sequence at the output.
  - However, $001000 \rightsquigarrow 000100$ and so the sequences of codewords $001, 000$ and $000, 100$ are confusable.
- We need to redefine the notion of zero-error code:

The Shift Channel: Comments

- In channels with delays, packets from one codeword can interfere with the following codeword.
- Example: Let $\mathcal{C} = \{000, 001, 100\}$ and $K = 1$
  - No two codewords can produce the same sequence at the output.
  - However, $001000 \rightsquigarrow 000100$ and so the sequences of codewords $001, 000$ and $000, 100$ are confusable.
- We need to redefine the notion of zero-error code:
  - A code is said to be zero-error if no two *sequences* of codewords can produce the same output.

The Shift Channel: Comments

- The previous problem can be circumvented by *padding* each codeword with K zeros, i.e., empty slots.

The Shift Channel: Comments

- The previous problem can be circumvented by *padding* each codeword with K zeros, i.e., empty slots.
  - Empty slots at the end of each codeword serve to "catch" the packets that are sent in the preceding slots and are delayed in the channel.

The Shift Channel: Comments

- The previous problem can be circumvented by *padding* each codeword with K zeros, i.e., empty slots.
  - Empty slots at the end of each codeword serve to "catch" the packets that are sent in the preceding slots and are delayed in the channel.
- Restricting to these codes is not a loss in generality (K is a constant):

$$\lim_{n\to\infty} \frac{1}{n} \log |\mathcal{C}(n)| = \lim_{n\to\infty} \frac{1}{n+K} \log |\mathcal{C}(n)|.$$

Optimal zero-error codes

Optimal zero-error codes

- Observation: The Shift Channel does not affect the Hamming weight of the transmitted codeword.

Optimal zero-error codes

- Observation: The Shift Channel does not affect the Hamming weight of the transmitted codeword.
    - Each constant-weight case can be observed separately.

Optimal zero-error codes

- Observation: The Shift Channel does not affect the Hamming weight of the transmitted codeword.
    - Each constant-weight case can be observed separately.
- Sequences of length n and weight W can be represented as W-tuples of integers $(p_1, \ldots, p_W)$, where p_i is the position of the i'th 1 in the sequence.

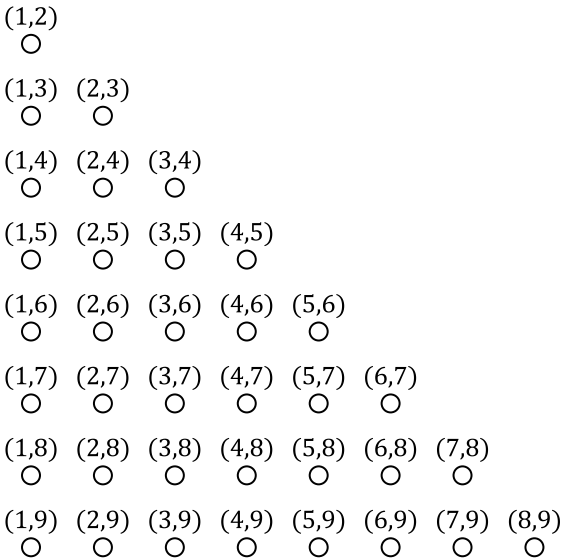
Optimal zero-error codes

- Observation: The Shift Channel does not affect the Hamming weight of the transmitted codeword.
    - Each constant-weight case can be observed separately.
- Sequences of length n and weight W can be represented as W-tuples of integers $(p_1, \ldots, p_W)$, where p_i is the position of the i'th 1 in the sequence.
    - $10010 \longleftrightarrow (1, 4)$

Optimal zero-error codes

- Observation: The Shift Channel does not affect the Hamming weight of the transmitted codeword.
    - Each constant-weight case can be observed separately.
- Sequences of length n and weight W can be represented as W-tuples of integers $(p_1, \ldots, p_W)$, where p_i is the position of the i'th 1 in the sequence.
    - $10010 \longleftrightarrow (1, 4)$
- Example: $n = 9$, $W = 2$, $K = 1$ – Let's try to construct a good code!

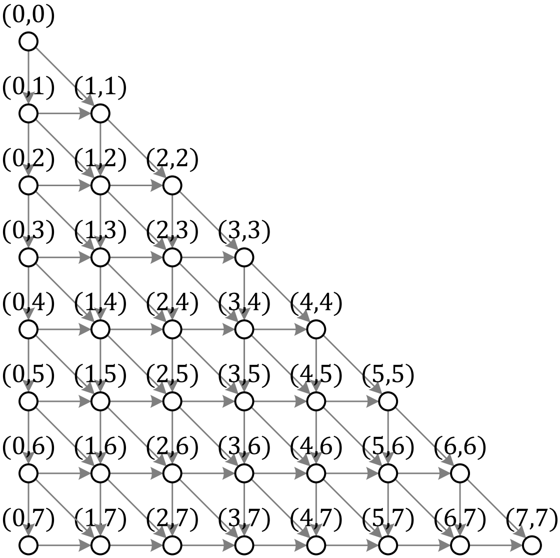
Optimal zero-error codes: Geometric approach



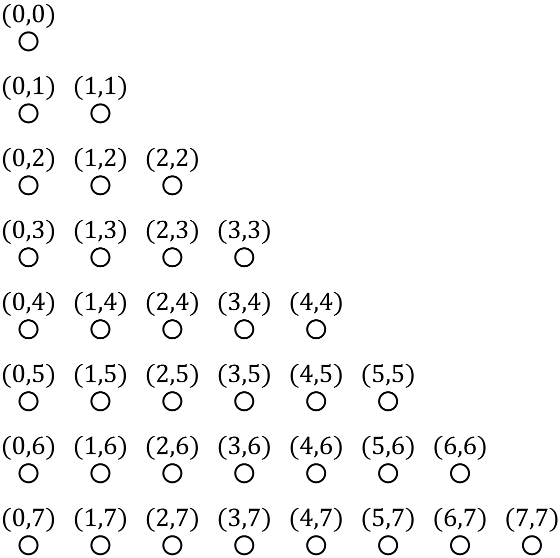
Optimal zero-error codes: Geometric approach

(0,0)							
(0,1)	(1,1)						
(0,2)	(1,2)	(2,2)					
(0,3)	(1,3)	(2,3)	(3,3)				
(0,4)	(1,4)	(2,4)	(3,4)	(4,4)			
(0,5)	(1,5)	(2,5)	(3,5)	(4,5)	(5,5)		
(0,6)	(1,6)	(2,6)	(3,6)	(4,6)	(5,6)	(6,6)	
(0,7)	(1,7)	(2,7)	(3,7)	(4,7)	(5,7)	(6,7)	(7,7)

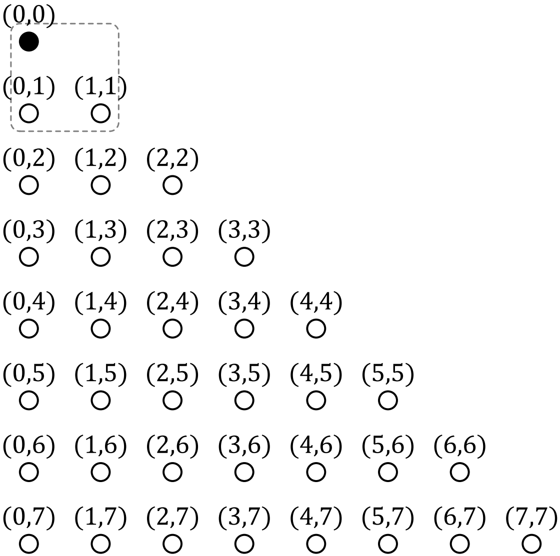
Optimal zero-error codes: Geometric approach



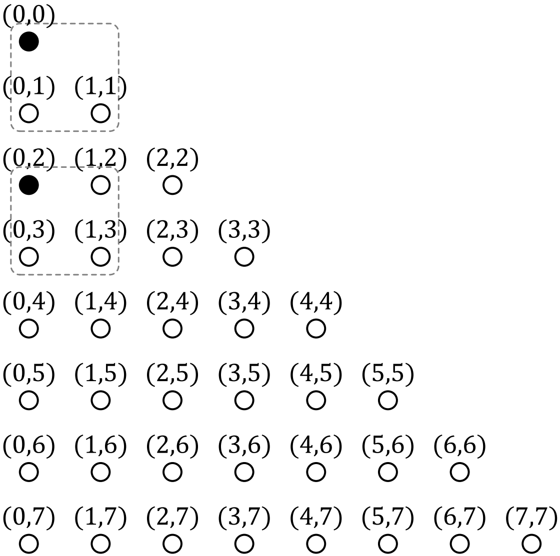
Optimal zero-error codes: Geometric approach



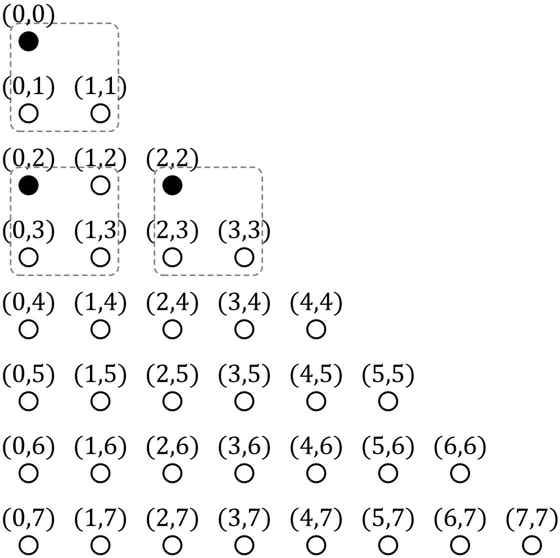
Optimal zero-error codes: Geometric approach



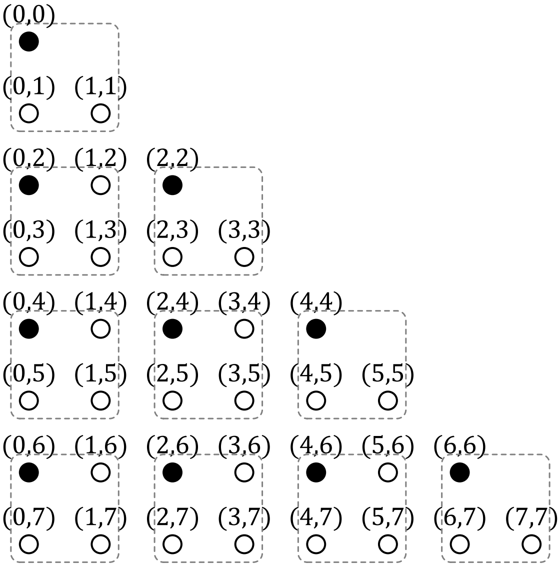
Optimal zero-error codes: Geometric approach



Optimal zero-error codes: Geometric approach



Optimal zero-error codes: Geometric approach



Optimal zero-error codes: Constant-weight case

Thm. Optimal code of length n and weight W is given by

$$\mathcal{C}(n, W) = \left\{ \mathrm{x} \in \Delta_{n-W}^{W} \; : \; \mathrm{x} = 0 \pmod{K+1} \right\}.$$

Optimal zero-error codes: Constant-weight case

Thm. Optimal code of length n and weight W is given by

$$\mathcal{C}(n, W) = \left\{ \mathrm{x} \in \Delta_{n-W}^{W} \ : \ \mathrm{x} = 0 \pmod{K+1} \right\}.$$

- The size of the optimal constant-weight code is therefore

$$M(n, W) = \binom{\lfloor \frac{n-W}{K+1} \rfloor + W}{W},$$

$$M(n) = \sum_{W=0}^{n} M(n, W).$$

Zero-error capacity

Zero-error capacity

- Since we proved the constructed codes are optimal, the zero-error capacity is equal to

$$C_0 = \lim_{n \to \infty} \frac{1}{n} \log M(n)$$

Zero-error capacity

- Since we proved the constructed codes are optimal, the zero-error capacity is equal to

$$C_0 = \lim_{n \to \infty} \frac{1}{n} \log M(n)$$

Theorem

The zero-error capacity of the Shift Channel with parameter K is equal to $\log r$, where r is the unique positive real root of the polynomial $x^{K+1} - x^K - 1$.

Proof:

- Turns out $M(n)$ can also be described recursively

$$M(n) = M(n-1) + M(n-K-1),$$

with $M(n) = n + 1$ for $n \leq K$.

- This implies that

$$M(n) = \sum_{k=0}^{K} a_k r_k^n,$$

where r_k are the roots of the polynomial $x^{K+1} - x^K - 1$, and a_k are (complex) constants.

- Therefore, $M(n) \sim ar^n$, where r is the largest of these roots (which is the unique positive real root). $\square$

Zero-error capacity: Constant-weight case

Zero-error capacity: Constant-weight case

- We can find the constant-weight zero-error capacity

Zero-error capacity: Constant-weight case

- We can find the constant-weight zero-error capacity (i.e. the largest rate attainable with the requirement that the ω fraction of the slots is used):

Zero-error capacity: Constant-weight case

- We can find the constant-weight zero-error capacity (i.e. the largest rate attainable with the requirement that the ω fraction of the slots is used):

$$C_0(\omega) = \lim_{n\to\infty} \frac{1}{n} \log M(n, \omega n) = \frac{\omega K + 1}{K+1} \mathcal{H}\left(\frac{\omega(K+1)}{\omega K + 1}\right).$$

Zero-error capacity: Constant-weight case

- We can find the constant-weight zero-error capacity (i.e. the largest rate attainable with the requirement that the ω fraction of the slots is used):

$$C_0(\omega) = \lim_{n\to\infty} \frac{1}{n} \log M(n, \omega n) = \frac{\omega K + 1}{K+1} \mathcal{H}\left(\frac{\omega(K+1)}{\omega K + 1} \right).$$

- As there are linearly many different weights, the zero-error capacity can be achieved with constant-weight codes, so

$$C_0 = \max_{\omega\in[0,1]} C_0(\omega) = \frac{\omega^* K + 1}{K+1} \mathcal{H}\left(\frac{\omega^*(K+1)}{\omega^* K + 1} \right),$$

for some ω^*.

Zero-error capacity: Constant-weight case

Zero-error capacity: Constant-weight case

- We can estimate this function further:

$$\frac{1}{n}\log M(n,\omega^* n) = C_0 - \frac{1}{2n}\log n + \mathcal{O}\left(\frac{1}{n}\right).$$

Zero-error capacity: Constant-weight case

- We can estimate this function further:

$$\frac{1}{n} \log M(n, \omega^* n) = C_0 - \frac{1}{2n} \log n + \mathcal{O}\left(\frac{1}{n}\right).$$

- Note: Even though the capacity can be achieved with constant-weight codes, their performance is worse at finite blocklengths.
  - ...quantified by the second-order term $-\frac{1}{2n} \log n$

Generalizations

Generalizations

- P types of packets; the delay of each packet is at most K, as before, and the packets cannot be reordered (queue with a FIFO service procedure).

Generalizations

- P types of packets; the delay of each packet is at most K, as before, and the packets cannot be reordered (queue with a FIFO service procedure).
    - Now the packets themselves also carry information.

Generalizations

- P types of packets; the delay of each packet is at most K, as before, and the packets cannot be reordered (queue with a FIFO service procedure).
    - Now the packets themselves also carry information.

Main observation: We can first design the "timing" code ($P = 1$), and then assign to every such codeword of weight W all possible sequences of packets (P^W of them)

Generalizations

- P types of packets; the delay of each packet is at most K, as before, and the packets cannot be reordered (queue with a FIFO service procedure).
  - Now the packets themselves also carry information.

Main observation: We can first design the "timing" code ($P = 1$), and then assign to every such codeword of weight W all possible sequences of packets (P^W of them)

  - $P = 2$: $\quad 10010 \longrightarrow A00A0, A00B0, B00A0, B00B0$

Generalizations

- P types of packets; the delay of each packet is at most K, as before, and the packets cannot be reordered (queue with a FIFO service procedure).
  - Now the packets themselves also carry information.

  Main observation: We can first design the "timing" code ($P = 1$), and then assign to every such codeword of weight W all possible sequences of packets (P^W of them)
  - $P = 2$: $\quad 10010 \longrightarrow A00A0, A00B0, B00A0, B00B0$
  - This construction is optimal.

Generalizations

- P types of packets; the delay of each packet is at most K, as before, and the packets cannot be reordered (queue with a FIFO service procedure).
  - Now the packets themselves also carry information.

Main observation: We can first design the "timing" code ($P = 1$), and then assign to every such codeword of weight W all possible sequences of packets (P^W of them)

- $P = 2$: $\quad 10010 \longrightarrow A00A0, A00B0, B00A0, B00B0$
- This construction is optimal.
- Zero-error capacity is equal to $\log r$, where r is the unique positive real root of the polynomial $x^{K+1} - Px^K - 1$.

Generalizations

Generalizations

- Allowed shifts from $\{-K_1, \ldots, 0, \ldots, K_2\}$

Generalizations

- Allowed shifts from $\{-K_1, \ldots, 0, \ldots, K_2\}$
  - Equivalent to shifts belonging to $\{0, \ldots, K_1 + K_2\}$

Generalizations

- Allowed shifts from $\{-K_1, \ldots, 0, \ldots, K_2\}$
  - Equivalent to shifts belonging to $\{0, \ldots, K_1 + K_2\}$
- Continuous-time channel with emissions separated by at least τ seconds, and with the maximum delay of T seconds
  - The capacity equals $\frac{1}{\tau} \log r$, where r is the unique positive root of the polynomial $x^{T/\tau} - x^{T/\tau - 1} - 1$

Zero-error-detecting codes

Zero-error-detecting codes

- *Zero-error-detecting code* is a code which can *detect* all errors (in our case shifts) allowed in the model. (We do not need to figure out what was sent.)

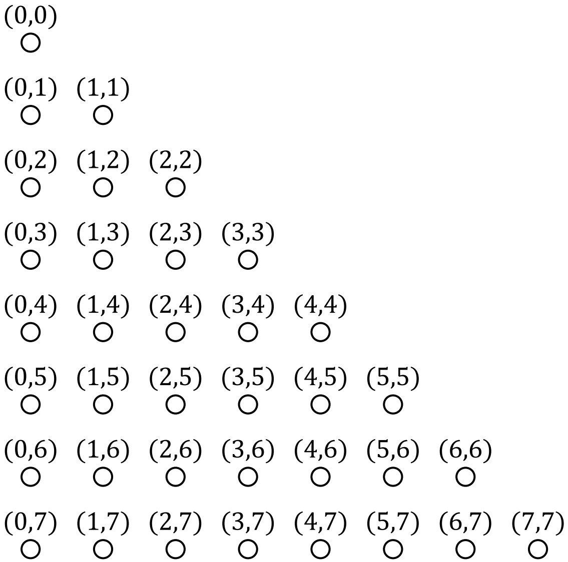
Zero-error-detecting codes

- *Zero-error-detecting code* is a code which can *detect* all errors (in our case shifts) allowed in the model. (We do not need to figure out what was sent.)
    - No codeword can produce another *codeword* at the output.

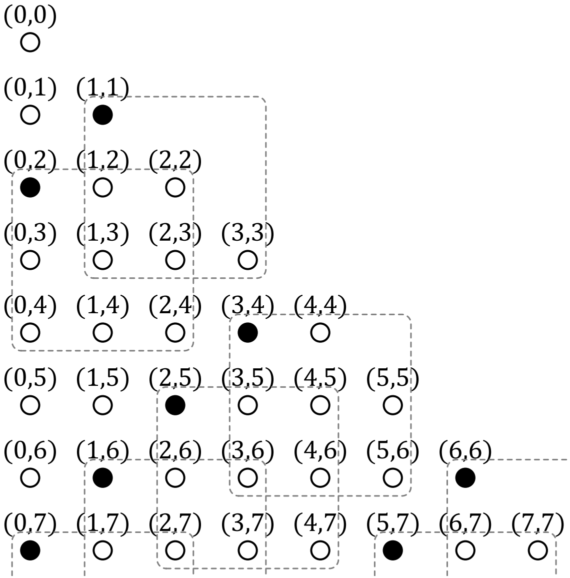
Zero-error-detecting codes

- *Zero-error-detecting code* is a code which can *detect* all errors (in our case shifts) allowed in the model. (We do not need to figure out what was sent.)
  - No codeword can produce another *codeword* at the output.
- *Zero-error-detection capacity* of a channel is the largest rate achievable (asymptotically) with zero-error-detecting codes.

Zero-error-detecting codes: Example ($K = 2$)



Zero-error-detecting codes: Example ($K = 2$)



Zero-error-detecting codes: Construction

Zero-error-detecting codes: Construction

- The shifts are now assumed to be $\in \{-K_1, \ldots, 0, \ldots, K_2\}$

Zero-error-detecting codes: Construction

- The shifts are now assumed to be $\in \{-K_1, \ldots, 0, \ldots, K_2\}$
  - Suppose also w.l.o.g. that $K_1 \leq K_2$

Zero-error-detecting codes: Construction

- The shifts are now assumed to be $\in \{-K_1, \dots, 0, \dots, K_2\}$
  - Suppose also w.l.o.g. that $K_1 \leq K_2$
- Code construction:

Zero-error-detecting codes: Construction

- The shifts are now assumed to be $\in \{-K_1, \ldots, 0, \ldots, K_2\}$
  - Suppose also w.l.o.g. that $K_1 \leq K_2$
- Code construction:

$$\mathcal{D}^{(a)}(n, W) = \Big\{ \mathrm{x} \in \Delta^W_{n-W} \ : \ \mathrm{x} = 0 \pmod{K_1 + 1}, \\ \sum_{i=1}^{W} x_i = a \pmod{WK_2 + 1} \Big\}.$$

Zero-error-detecting codes: Construction

- The shifts are now assumed to be $\in \{-K_1, \ldots, 0, \ldots, K_2\}$
  - Suppose also w.l.o.g. that $K_1 \leq K_2$
- Code construction:

$$\mathcal{D}^{(a)}(n, W) = \Big\{ \mathrm{x} \in \Delta_{n-W}^{W} \ : \ \mathrm{x} = 0 \pmod{K_1 + 1}, \quad \sum_{i=1}^{W} x_i = a \pmod{W K_2 + 1} \Big\}.$$

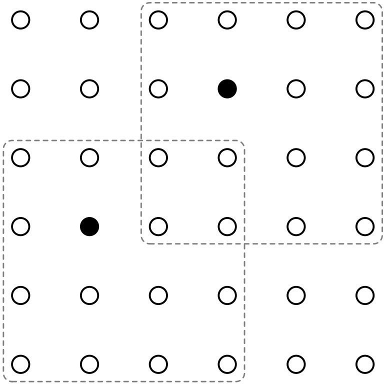
  - This code is a subcode of $\mathcal{C}(n, W)$ obtained as its intersection with the hyperplanes $\sum_{i=1}^{W} x_i = a \pmod{W K_2 + 1}$

Zero-error-detection capacity

Claim. Every code *detecting* shifts from $\{-K_1, \ldots, 0, \ldots, K_2\}$ is a code *correcting* shifts from $\{-K_1, \ldots, 0\}$

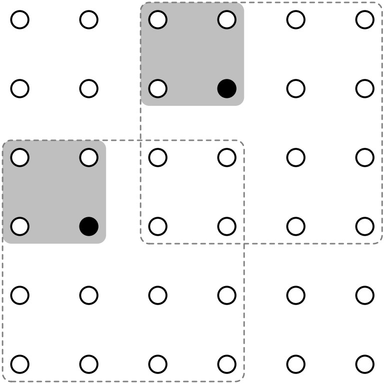
Zero-error-detection capacity

Claim. Every code *detecting* shifts from $\{-K_1, \dots, 0, \dots, K_2\}$ is a code *correcting* shifts from $\{-K_1, \dots, 0\}$



Zero-error-detection capacity

Claim. Every code *detecting* shifts from $\{-K_1, \ldots, 0, \ldots, K_2\}$ is a code *correcting* shifts from $\{-K_1, \ldots, 0\}$



Zero-error-detection capacity

Theorem

The zero-error-detection capacity of the Shift Channel with parameters K_1, K_2, *is equal to* $\log r$, *where* r *is the unique positive real root of the polynomial* $x^{\min\{K_1,K_2\}+1} - x^{\min\{K_1,K_2\}} - 1$.

Zero-error-detection capacity

Theorem

The zero-error-detection capacity of the Shift Channel with parameters K_1, K_2, is equal to $\log r$, where r is the unique positive real root of the polynomial $x^{\min\{K_1,K_2\}+1} - x^{\min\{K_1,K_2\}} - 1$.

- ...which is the same as the zero-error-correction capacity of the Shift Channel with parameters 0, $\min\{K_1, K_2\}$.

Discrete-time queues: Model description

Discrete-time queues: Model description

- DTQP—Discrete-Time Queue with bounded Processing times

- DTQP—Discrete-Time Queue with bounded Processing times
- The model:

Discrete-time queues: Model description

- DTQP—Discrete-Time Queue with bounded Processing times

- The model:

  1) Time is slotted, meaning that the packets are sent and received in integer time instants;

Discrete-time queues: Model description

- DTQP—Discrete-Time Queue with bounded Processing times
- The model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent in each time slot;

Discrete-time queues: Model description

- DTQP—Discrete-Time Queue with bounded Processing times
- The model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent in each time slot;
  3) Every packet is being *processed* by the server for a number of slots chosen randomly from the set $\{0, 1, \ldots, K\}$;

Discrete-time queues: Model description

- DTQP—Discrete-Time Queue with bounded Processing times

- The model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent in each time slot;
  3) Every packet is being *processed* by the server for a number of slots chosen randomly from the set $\{0, 1, \dots, K\}$;
  4) The packets are indistinguishable, and hence the information is conveyed via timing only.

Discrete-time queues: Model description

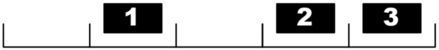
- DTQP—Discrete-Time Queue with bounded Processing times
- The model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent in each time slot;
  3) Every packet is being *processed* by the server for a number of slots chosen randomly from the set $\{0, 1, \ldots, K\}$;
  4) The packets are indistinguishable, and hence the information is conveyed via timing only.

Discrete-time queues: Model description

- DTQP—Discrete-Time Queue with bounded Processing times
- The model:
    1) Time is slotted, meaning that the packets are sent and received in integer time instants;
    2) At most 1 packet is sent in each time slot;
    3) Every packet is being *processed* by the server for a number of slots chosen randomly from the set $\{0, 1, \ldots, K\}$;
    4) The packets are indistinguishable, and hence the information is conveyed via timing only.

Discrete-time queues: Model description

- DTQP—Discrete-Time Queue with bounded Processing times
- The model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent in each time slot;
  3) Every packet is being *processed* by the server for a number of slots chosen randomly from the set $\{0, 1, \ldots, K\}$;
  4) The packets are indistinguishable, and hence the information is conveyed via timing only.



Discrete-time queues: Model description

- DTQP—Discrete-Time Queue with bounded Processing times

- The model:

  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent in each time slot;
  3) Every packet is being *processed* by the server for a number of slots chosen randomly from the set $\{0, 1, \ldots, K\}$;
  4) The packets are indistinguishable, and hence the information is conveyed via timing only.

Discrete-time queues: Comments

Discrete-time queues: Comments

- The total delay of a packet can now be much larger, because it has to wait for the other packets that arrived before it to be processed.

Discrete-time queues: Comments

- The total delay of a packet can now be much larger, because it has to wait for the other packets that arrived before it to be processed.
- The received sequence can be as long as $(K+1)n$ (longer than the input by a multiplicative constant!)...

Discrete-time queues: Comments

- The total delay of a packet can now be much larger, because it has to wait for the other packets that arrived before it to be processed.

- The received sequence can be as long as $(K+1)n$ (longer than the input by a multiplicative constant!)...

- We have to incorporate this fact in the definition of the code rate:

$$\frac{1}{L_{\mathrm{av}}(n)} \log M(n),$$

  where $L_{\mathrm{av}}(n)$ is the average output length (over all codewords and channel statistics).

Discrete-time queues: Comments

- The total delay of a packet can now be much larger, because it has to wait for the other packets that arrived before it to be processed.

- The received sequence can be as long as $(K+1)n$ (longer than the input by a multiplicative constant!)...

- We have to incorporate this fact in the definition of the code rate:
$$\frac{1}{L_{\mathrm{av}}(n)} \log M(n),$$
where $L_{\mathrm{av}}(n)$ is the average output length (over all codewords and channel statistics).

- Etc.

And finally...

And finally...

– the end –