Timing Channels and Shift-Correcting Codes

Mladen Kovačević

Department of Electrical & Computer Engineering
National University of Singapore

(with Petar Popovski, Miloš Stojaković, and Vincent Y. F. Tan)

Network Coding and Designs
Dubrovnik, Croatia

April 4, 2016

Contents

Zero-Error Communication

Zero-Error Communication

- *Zero-error code* is a code with the probability of error equal to zero (under optimal decoding)

Zero-Error Communication

- *Zero-error code* is a code with the probability of error equal to zero (under optimal decoding)
  - $\iff$ No two codewords can produce the same output

Zero-Error Communication

- *Zero-error code* is a code with the probability of error equal to zero (under optimal decoding)
  - $\Longleftrightarrow$ 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

- *Zero-error code* is a code with the probability of error equal to zero (under optimal decoding)
  - $\Longleftrightarrow$ 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

Zero-Error Communication

- *Zero-error code* is a code with the probability of error equal to zero (under optimal decoding)
  - $\Longleftrightarrow$ 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

- *Zero-error code* is a code with the probability of error equal to zero (under optimal decoding)
  - $\iff$ No two codewords can produce the same output

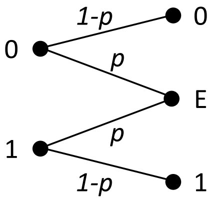
- *Zero-error capacity* of a channel, denoted C_0, is the largest rate achievable with zero-error codes
  - If $M(n)$ is the size of the largest zero-error code of length n for a given channel, then

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

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

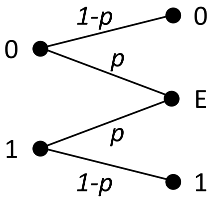
Zero-Error Communication

- Example: The BEC



Zero-Error Communication

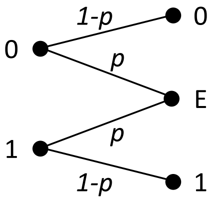
- Example: The BEC



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

Zero-Error Communication

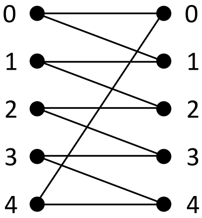
- Example: The BEC



- 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

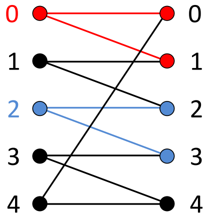
Zero-Error Communication

- Another Example:



Zero-Error Communication

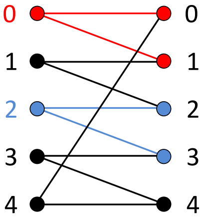
- Another Example:



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

Zero-Error Communication

- Another Example:



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

- 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

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. (IT paper award)

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 in each time slot;

The Shift Channel

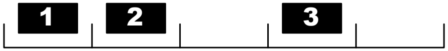
- Units of transmission: *packets*
- The basic model:
  1) Time is slotted, meaning that the packets are sent and received in integer time instants;
  2) At most 1 packet is sent in each time slot;
  3) Every packet is delayed in the channel for a number of slots chosen randomly from the set $\{0, 1,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

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 in each time slot;
  3) Every packet is delayed in the channel 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.

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 in each time slot;
  3) Every packet is delayed in the channel 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.



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 in each time slot;
  3) Every packet is delayed in the channel 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.

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 in each time slot;
  3) Every packet is delayed in the channel 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.

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 in each time slot;
  3) Every packet is delayed in the channel 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.

The Shift Channel: Comments

The Shift Channel: Comments

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

The Shift Channel: Comments

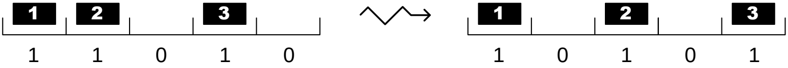
- This model is equivalent to a discrete-time queue with bounded *residence* times
  - V. Anantharam and S. Verdu, "Bits Through Queues," *IEEE Trans. Inf. Theory* 42 (1), 1996. (IT paper award)

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



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
  - It is enough to consider the constant-weight case

Optimal zero-error codes

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

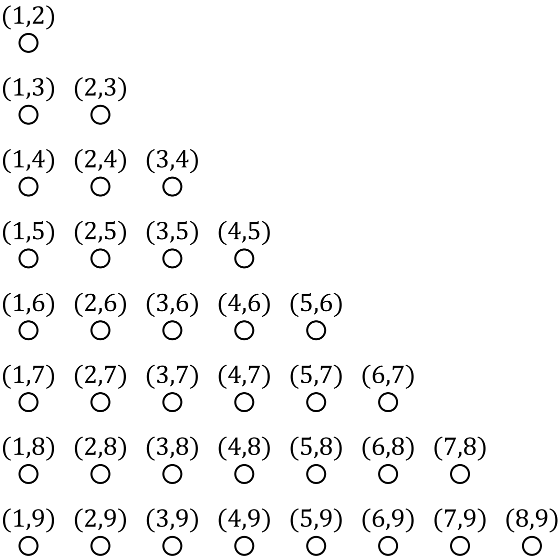
Optimal zero-error codes

- Observation: The Shift Channel does not affect the Hamming weight of the transmitted codeword
  - It is enough to consider the constant-weight case
- 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
  - $10010 \longleftrightarrow (1, 4)$

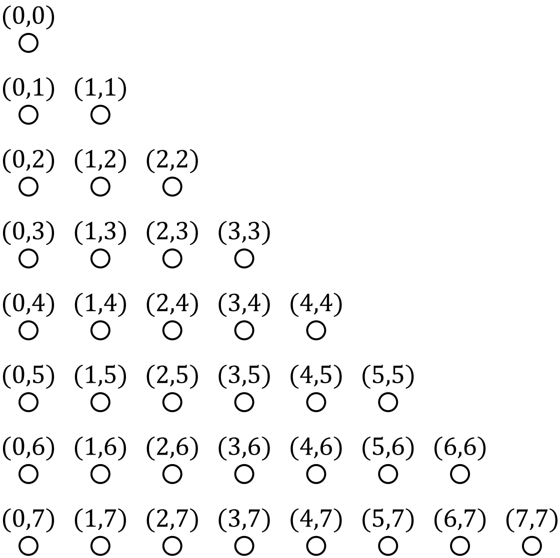
Optimal zero-error codes

- Observation: The Shift Channel does not affect the Hamming weight of the transmitted codeword
  - It is enough to consider the constant-weight case
- Sequences of length n and weight W can be represented as W-tuples of integers $(p_1, \dots, p_W)$, where p_i is the position of the i'th 1 in the sequence
  - $10010 \longleftrightarrow (1, 4)$
- Example: $n = 9$, $W = 2$, $K = 1$

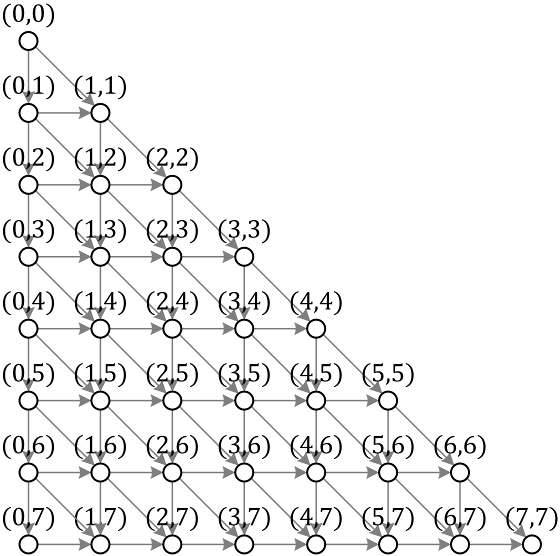
Optimal zero-error codes: Geometric approach



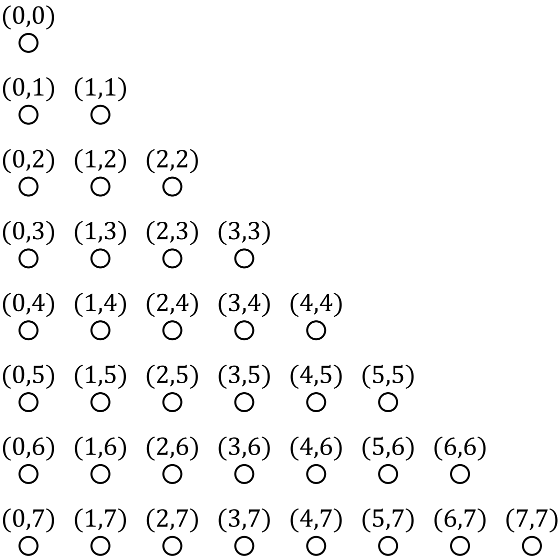
Optimal zero-error codes: Geometric approach



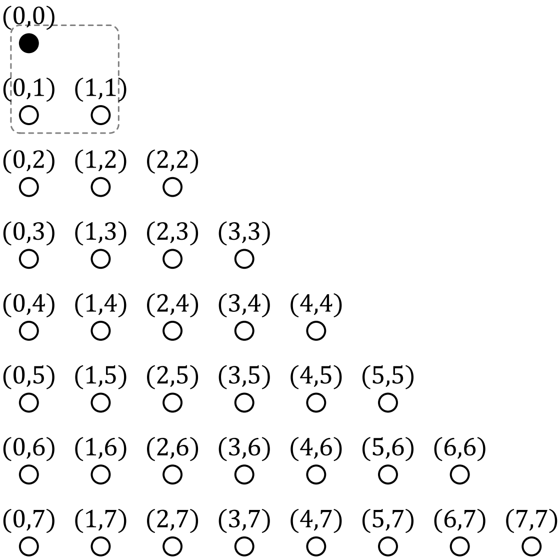
Optimal zero-error codes: Geometric approach



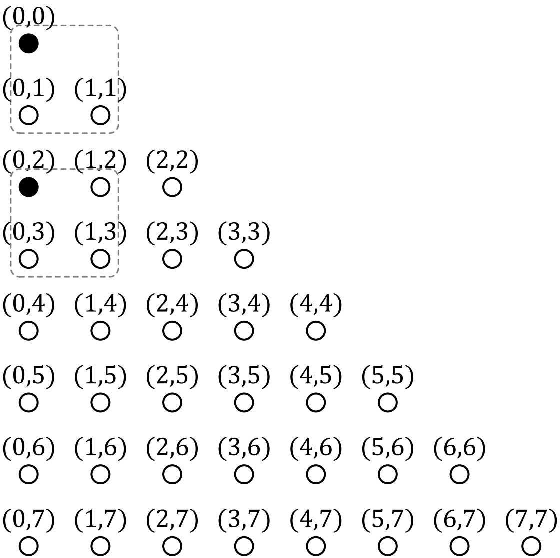
Optimal zero-error codes: Geometric approach



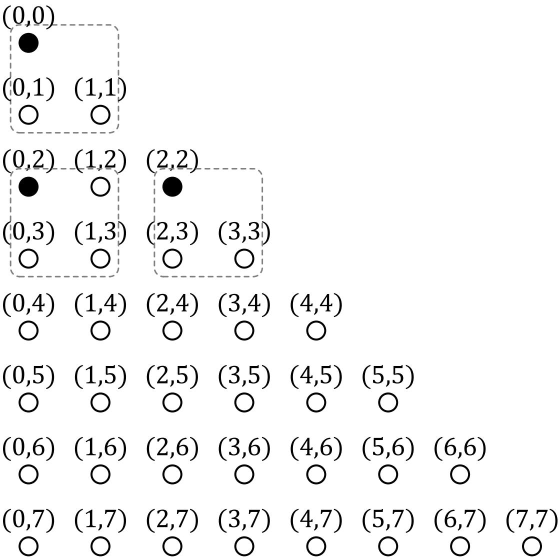
Optimal zero-error codes: Geometric approach



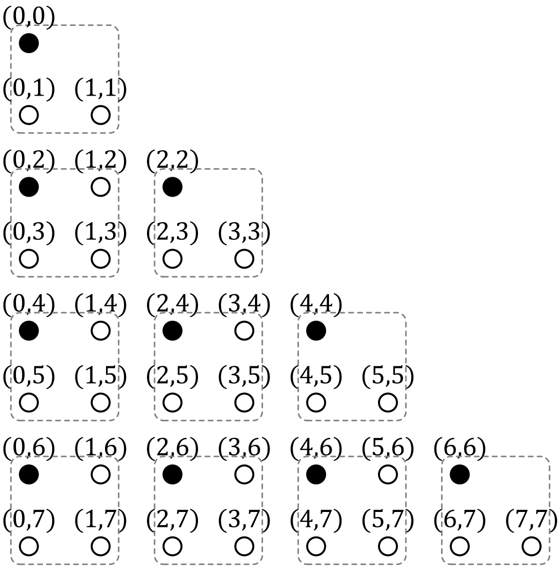
Optimal zero-error codes: Geometric approach



Optimal zero-error codes: Geometric approach



Optimal zero-error codes: Geometric approach



Optimal zero-error codes: Constant-weight case

Optimal zero-error codes: Constant-weight case

- Optimal code of length n and weight W is given by

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

Optimal zero-error codes: Constant-weight case

- Optimal code of length n and weight W is given by

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

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

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

Optimal zero-error codes: Constant-weight case

- Optimal code of length n and weight W is given by

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

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

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

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

Zero-error capacity

Zero-error capacity

- Since 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 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 $x^{K+1} - x^K - 1$.

Zero-error capacity

- Proof:
  - $M(n)$ 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)

Zero-error capacity: Constant-weight case

Zero-error capacity: Constant-weight case

- The constant-weight zero-error capacity—

- The constant-weight zero-error capacity—the largest rate attainable asymptotically with the requirement that the fraction of the slots used for transmitting packets equals ω:

Zero-error capacity: Constant-weight case

- The constant-weight zero-error capacity—the largest rate attainable asymptotically with the requirement that the fraction of the slots used for transmitting packets equals ω:

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

Zero-error capacity: Constant-weight case

- The constant-weight zero-error capacity—the largest rate attainable asymptotically with the requirement that the fraction of the slots used for transmitting packets equals ω:

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

- The zero-error capacity can be achieved with constant-weight codes, so

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

Zero-error capacity: Constant-weight case

Zero-error capacity: Constant-weight case

- More precise asymptotics:

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

Zero-error capacity: Constant-weight case

- More precise asymptotics:

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

- Note: Even though the capacity can be achieved with constant-weight codes, their performance is worse at finite blocklengths

Zero-error capacity: Constant-weight case

- More precise asymptotics:

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

- Note: Even though the capacity can be achieved with constant-weight codes, their performance is worse at finite blocklengths
    - This is 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)
  - $P = 2$: $10010 \longrightarrow A00A0, A00B0, B00A0, B00B0$
  - 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

- P types of packets; additional noise modeled as a memoryless channel "acting" on the packets and having zero-error capacity C_0'

Generalizations

- P types of packets; additional noise modeled as a memoryless channel "acting" on the packets and having zero-error capacity C_0'
  - Replace P with $2^{C_0'}$

Generalizations

- P types of packets; additional noise modeled as a memoryless channel "acting" on the packets and having zero-error capacity C_0'
  - Replace P with $2^{C_0'}$
- Allowed shifts from $\{-K_1, \ldots, 0, \ldots, K_2\}$

Generalizations

- P types of packets; additional noise modeled as a memoryless channel "acting" on the packets and having zero-error capacity C'_0
  - Replace P with $2^{C'_0}$
- Allowed shifts from $\{-K_1, \ldots, 0, \ldots, K_2\}$
  - Equivalent to shifts belonging to $\{0, \ldots, K_1 + K_2\}$

Generalizations

- P types of packets; additional noise modeled as a memoryless channel "acting" on the packets and having zero-error capacity C_0'
  - Replace P with $2^{C_0'}$
- 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

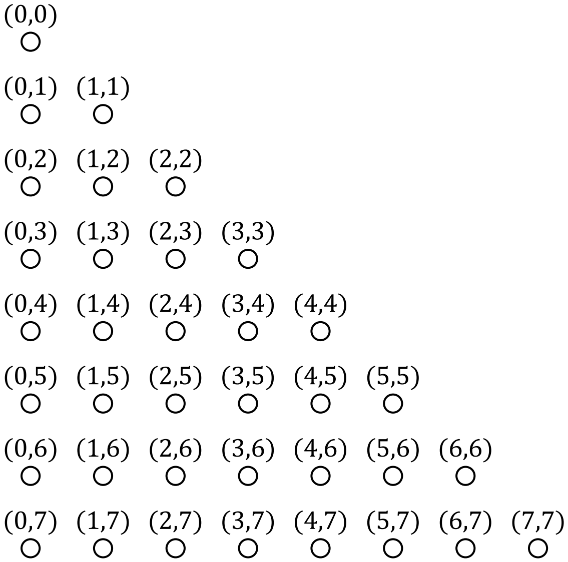
Zero-error-detecting codes

- *Zero-error-detecting code* is a code which can *detect* all errors (in our case shifts) allowed in the model
  - No codeword can produce another *codeword* at the output

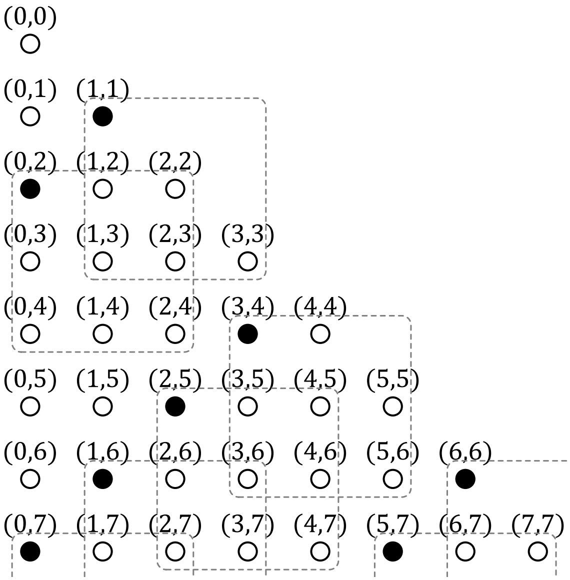
Zero-error-detecting codes

- *Zero-error-detecting code* is a code which can *detect* all errors (in our case shifts) allowed in the model
  - 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 $\in \{-K_1, \dots, 0, \dots, K_2\}$
  - Suppose also w.l.o.g. that $K_1 \leq K_2$

Zero-error-detecting codes: Construction

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

$$\mathcal{D}^{(a)}(n, W) = \Bigg\{ \mathbf{x} \in \Delta_{n-W}^{W} \;:\; \mathbf{x} = \mathbf{0} \pmod{K_1 + 1}, \\ \sum_{i=1}^{W} x_i = a \pmod{WK_2 + 1} \Bigg\}.$$

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:

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

  - This code is a subcode of $\mathcal{C}(n, W)$ obtained as its intersection with the hyperplanes $\sum_{i=1}^{W} x_i = a \pmod{WK_2 + 1}$

Zero-error-detecting codes: Construction

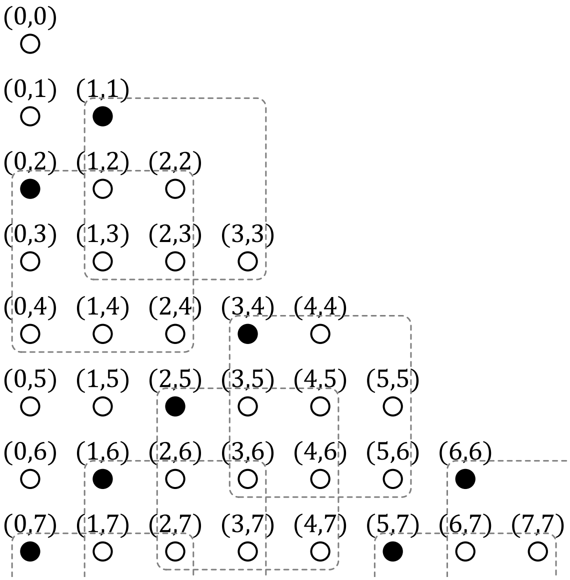
- The shifts are now assumed to be $\in \{-K_1, \ldots, 0, \ldots, K_2\}$
  - Suppose also w.l.o.g. that $K_1 \leq K_2$

- Code construction:

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

  - This code is a subcode of $\mathcal{C}(n, W)$ obtained as its intersection with the hyperplanes $\sum_{i=1}^{W} x_i = a \pmod{WK_2 + 1}$
  - Example: $n = 9, \; W = 2, \; K_1 = 0, \; K_2 = 2$

Zero-error-detecting codes: Construction $(K_1 = 0, K_2 = 2)$



Zero-error-detecting codes: Construction

Zero-error-detecting codes: Construction

- **Claim:** The code $\mathcal{D}^{(a)}(n, W)$ is zero-error-detecting

Zero-error-detecting codes: Construction

- **Claim:** The code $\mathcal{D}^{(a)}(n, W)$ is zero-error-detecting
- Proof:
  - Let $\mathbf{y}$ be the received sequence
  - If $\sum_{i=1}^{W} y_i \neq a \pmod{WK_2 + 1}$, the receiver detects an error
  - Suppose then that $\sum_{i=1}^{W} y_i = a \pmod{WK_2 + 1}$
  - This means that the sum of the coordinates has been changed in the channel by a multiple of $WK_2 + 1$
  - Since $-K_1 \leq y_i - x_i \leq K_2$ in our model, we have $-WK_1 \leq \sum_{i=1}^{W}(y_i - x_i) \leq WK_2$, so the sum could not have been changed for a nonzero multiple of $WK_2 + 1$
  - Therefore, the sum wasn't changed at all and, if there were any shifts in channel, some of them must have been shifts to the right and some of them to the left
  - Suppose that the i'th particle was shifted to the left, $y_i < x_i$
  - Then, since x_i is a multiple of $K_1 + 1$, and $-K_1 \leq y_i - x_i < 0$, y_i cannot be a multiple of $K_1 + 1$, and so $\mathbf{y}$ is not a codeword

Zero-error-detecting codes: Construction

Zero-error-detecting codes: Construction

- What is the rate of the codes $\mathcal{D}^{(a)}(n, W)$?

Zero-error-detecting codes: Construction

- What is the rate of the codes $\mathcal{D}^{(a)}(n, W)$?
- Note that

$$\mathcal{C}(n, W) = \bigcup_{a=0}^{WK_2} \mathcal{D}^{(a)}(n, W)$$

Zero-error-detecting codes: Construction

- What is the rate of the codes $\mathcal{D}^{(a)}(n, W)$?
- Note that

$$\mathcal{C}(n, W) = \bigcup_{a=0}^{WK_2} \mathcal{D}^{(a)}(n, W)$$

- This implies that, for at least one a

$$\left|\mathcal{D}^{(a)}(n, W)\right| \geq \frac{|\mathcal{C}(n, W)|}{WK_2 + 1}$$

Zero-error-detecting codes: Construction

- What is the rate of the codes $\mathcal{D}^{(a)}(n, W)$?
- Note that

$$\mathcal{C}(n, W) = \bigcup_{a=0}^{WK_2} \mathcal{D}^{(a)}(n, W)$$

- This implies that, for at least one a

$$\left|\mathcal{D}^{(a)}(n, W)\right| \geq \frac{|\mathcal{C}(n, W)|}{WK_2 + 1}$$

  - Asymptotically, as $n \to \infty$ and $W \sim \omega n$, the codes $\mathcal{D}^{(a)}(n, W)$ have the same rate as the codes $\mathcal{C}(n, W)$ designed for the smaller of the two parameters K_1, K_2

Zero-error-detection capacity

Zero-error-detection capacity

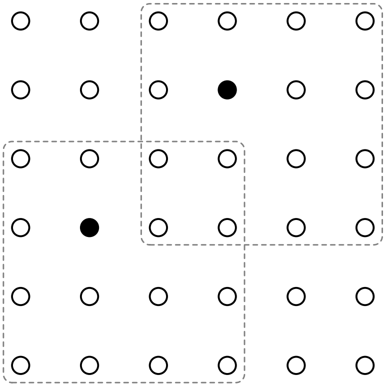
- One cannot do better than this (asymptotically):

  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

- One cannot do better than this (asymptotically):

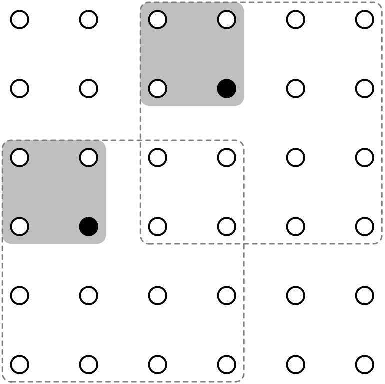
  Every code detecting shifts from $\{-K_1, \ldots, 0, \ldots, K_2\}$ is a code correcting shifts from $\{-K_1, \ldots, 0\}$



Zero-error-detection capacity

- One cannot do better than this (asymptotically):

  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

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

Discrete-time queues: Model description

- 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 $\{0, 1, \dots, K\}$;

Discrete-time queues: Model description

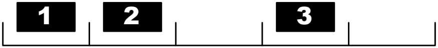
- DTQP—Discrete-Time Queue with bounded Processing times

- 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 $\{0, 1, \ldots, K\}$;
  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: 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 for 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 for a multiplicative constant!)
- We have to incorporate this fact in the definition of the code rate:

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 for a multiplicative constant!)
- We have to incorporate this fact in the definition of the code rate:

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

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

Optimal zero-error codes: Geometric approach

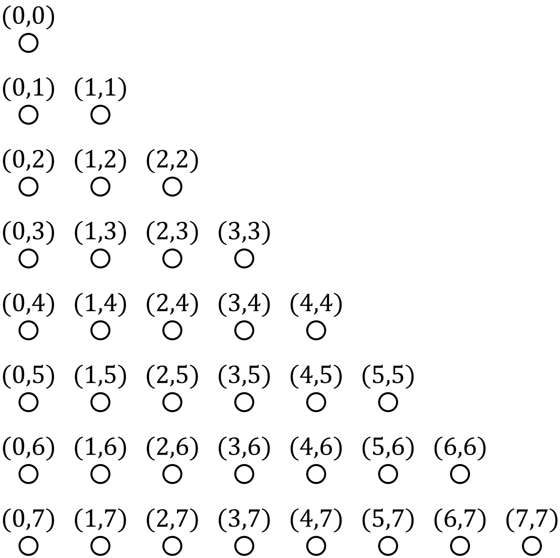
Optimal zero-error codes: Geometric approach

- Note: We can again focus on the constant-weight case

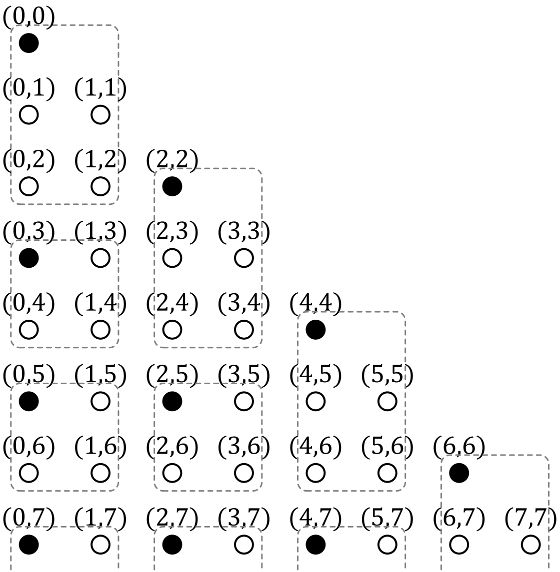
Optimal zero-error codes: Geometric approach

- Note: We can again focus on the constant-weight case
- Example: $n = 9$, $W = 2$, $K = 1$

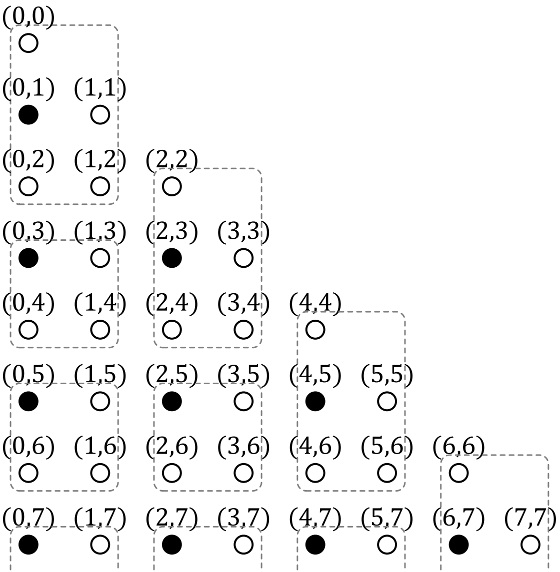
Optimal zero-error codes: Geometric approach



Optimal zero-error codes: Geometric approach



Optimal zero-error codes: Geometric approach



Optimal zero-error codes

- Optimal solution:

Optimal zero-error codes

- Optimal solution:
  - Forbid the sequences whose 1's are too close to affect each other (i.e., with less than K zeros in between)

Optimal zero-error codes

- Optimal solution:
  - Forbid the sequences whose 1's are too close to affect each other (i.e., with less than K zeros in between)
  - The effect of the channel on the remaining sequences is the same as in the Shift Channel, so construct an optimal code for it, as before

Optimal zero-error codes

- Optimal solution:
    - Forbid the sequences whose 1's are too close to affect each other (i.e., with less than K zeros in between)
    - The effect of the channel on the remaining sequences is the same as in the Shift Channel, so construct an optimal code for it, as before
- The size of the resulting codes is

$$M^{\mathrm{Q}}(n, W) = \binom{\frac{n+K}{K+1}}{W}$$

Optimal zero-error codes

- Optimal solution:
  - Forbid the sequences whose 1's are too close to affect each other (i.e., with less than K zeros in between)
  - The effect of the channel on the remaining sequences is the same as in the Shift Channel, so construct an optimal code for it, as before
- The size of the resulting codes is

$$M^{\mathrm{Q}}(n, W) = \binom{\frac{n+K}{K+1}}{W}$$

- The capacity:

$$C_0^{\mathrm{Q}} = \frac{1}{K+1}$$

Generalizations

Generalizations

- P types of packets, no reordering (FIFO service)

Generalizations

- P types of packets, no reordering (FIFO service)

Theorem

Zero-error capacity of the DTQP with P types of packets is

$$C_0^{\mathrm{Q}} = \max\left\{\frac{\log(P+1)}{K+1}\,,\ \frac{\log P}{\bar{\kappa}+1}\right\}.$$

Generalizations

- P types of packets, no reordering (FIFO service)

Theorem

Zero-error capacity of the DTQP with P types of packets is

$$C_0^{\mathrm{Q}} = \max\left\{\frac{\log(P+1)}{K+1}, \frac{\log P}{\bar{\kappa}+1}\right\}.$$

- Even though we are analyzing the zero-error case, the capacity depends on the channel statistics through $\bar{\kappa} = \sum_{k=0}^{K} k \cdot p(k)$

And Finally...

Thank you for your attention!