Wireless Connectivity: An Intuitive and Fundamental Guide

Chapter 7: Coding for Reliable Communication

Petar Popovski

Connectivity Section

Department of Electronic Systems

petarp@es.aau.dk

Contributions to the slides:

Israel Leyva-Mayorga

Radoslaw Kotaba

Abolfazl Amiri

Alexandru-Sabin Bana

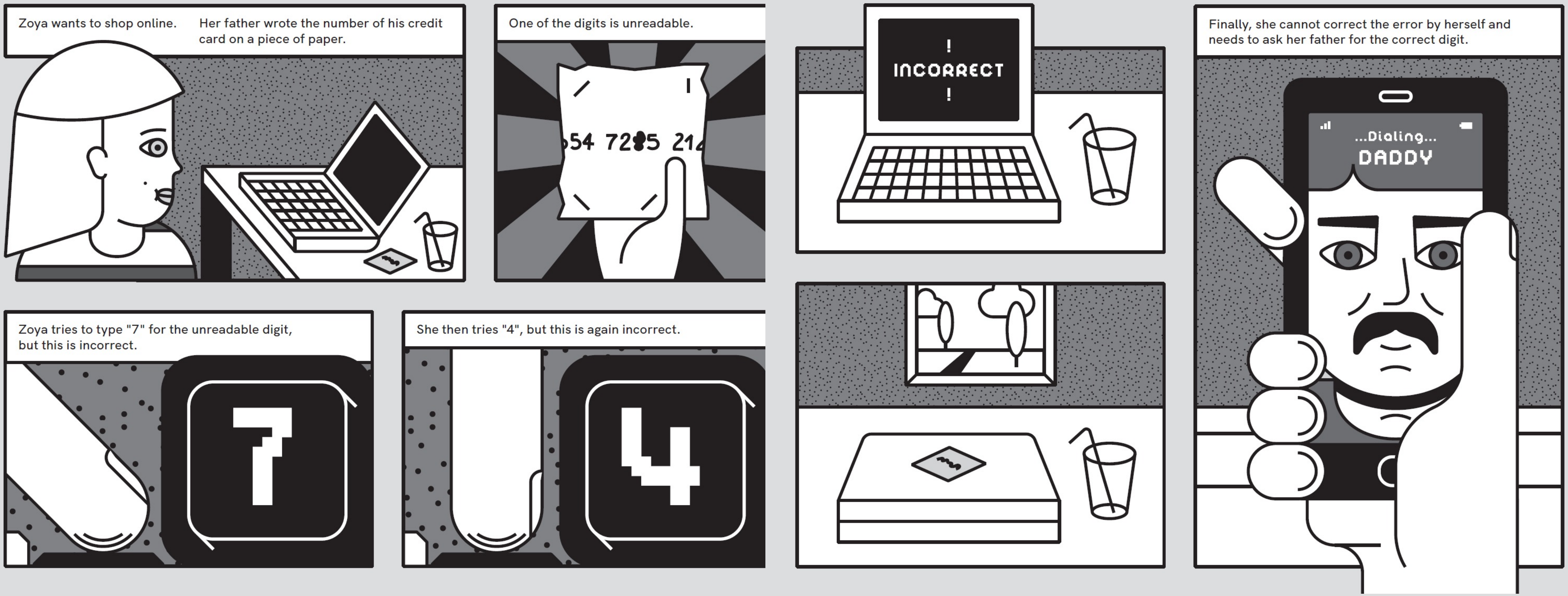
Robin J. WIlliams

AALBORG UNIVERSITY

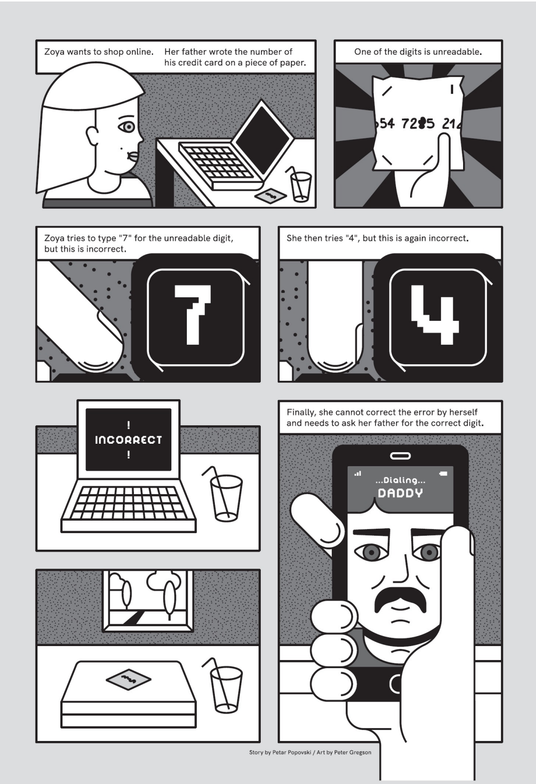
DENMARK

Modules

1. An easy introduction to the shared wireless medium
2. Random Access: How to Talk in Crowded Dark Room
3. Access Beyond the Collision Model
4. The Networking Cake: Layering and Slicing
5. Packets Under the Looking Glass: Symbols and Noise
6. A Mathematical View on a Communication Channel
7. **Coding for Reliable Communication**
8. Information-Theoretic View on Wireless Channel Capacity
9. Time and frequency in wireless communications
10. Space in wireless communications
11. Using Two, More, or a Massive Number of Antennas
12. Wireless Beyond a Link: Connections and Networks


Zoya wants to shop online.
Her father wrote the number of his credit card on a piece of paper.
One of the digits is unreadable.
INCORRECT
Zoya tries to type "7" for the unreadable digit, but this is incorrect.
7
She then tries "4", but this is again incorrect.
4
Finally, she cannot correct the error by herself and needs to ask her father for the correct digit.
...Dialing...
DADDY

Importance of coding for reliable communication



- Allowing every input sequence
  to be a valid message would make communication unreliable
- Coding is one of the foundations of communication: it allows to detect or correct errors
- Communication can be further improved by employing feedback and retransmissions

What will be learned in this chapter

- Basic coding ideas
- Detection and correction through coding
- Combining coding and modulation
- Role of feedback and retransmissions

Communication over unreliable channels

How to achieve reliable communication
with unreliable individual channel uses

- Use of packets containing multiple symbols
- Including **redundancy** for error **detection** and **correction**

Assume: **ideal** error check

The goodput is affected by the total probability of error and
the length of the overhead

$$G = \frac{b}{b+c}(1-p)^{b+c}$$

Coding and Binary Symmetric Channel (BSC)

(a) (b)

Simplest form:

repetition coding

Each bit is sent 3 times $y_1 y_2 y_3 \in \{000,001,010,011,100,101,110,111\}$

Majority voting $p_E = 3p^2(1-p) + p^3$

We obtain a **new BSC (r-channel)** with error probability p_E

Cost of **longer** symbol time (**triple** symbol time of *r*-channel)

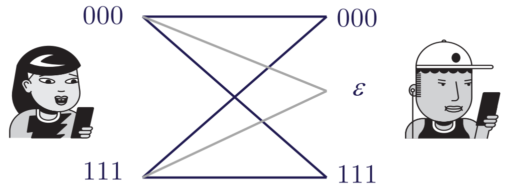
$$G_p = \frac{b}{l}(1-p)^l \quad ; \quad G_r = \frac{b}{3l}(1-p_E)^l \qquad G_r = G\,\frac{1}{3}\left(\frac{1-p_E}{1-p}\right)^l$$

Coding and Binary Symmetric Channel (BSC)

Goodput is increased if $\frac{1}{3}\left(\frac{1-p_E}{1-p}\right)^l > 1$

- This is not always the case
- In fact, repetition coding is rather inefficient, unless p is high
  - E.g. for $l = 300$, $p = 0.495$ we have $G_r = 1.15G$

Repetition coding with erasures



(c)

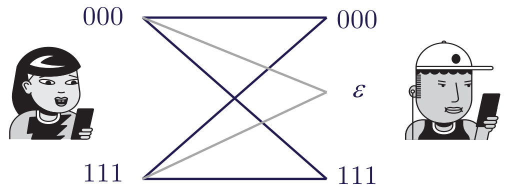
$p_S = (1-p)^3$
$p_E = p^3$
$p_{ERS} = 3p(1-p)^2 + 3p^2(1-p) = 3p(1-p)$

Assume: p is low → then p_E and p_{ERS} are also low

- Furthermore, let $p_{ERS} \sim \frac{1}{l}$, s.t. small number of erasures can happen
- We can use the error detection capabilities to recover the packet!
  - **Further** assumption: CRC is ideal

$$G_\epsilon \approx \frac{b}{3l}(1-p^3)^l = G\,\frac{1}{3}\left(\frac{1-p^3}{1-p}\right)^l$$

Repetition coding with erasures



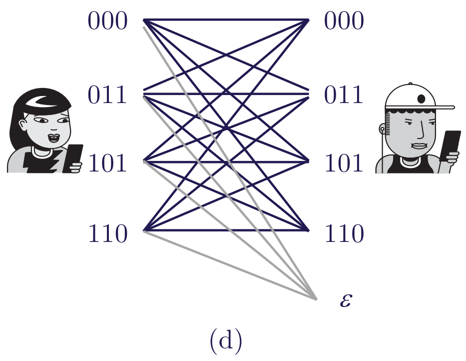
(c)

Some caveats

- To recover the packet despite the erasures we had to test different hypotheses i.e. "The erased bit was 0" and "The erased bit was "1"
  - The error correction is possible also with p-channel, but more cumbersome
  - If the number of erasures is n we might have to test all 2^n cases
- Furthermore, error detection in practice is not ideal
  - A specific combination of error flips may lead to another valid packet

Can we do even better (while still using 3 BSC channel uses per bit)?

Coding beyond repetition



Repetition coding restricts to **2 out of 8 possible** 3-BSC symbols

How about if we choose **4 out of 8** symbols

Nominal goodput $\frac{b}{\frac{3l}{2}} = \frac{2}{3}\frac{b}{l}$

Which 4 new symbols? ones with **maximum separation** (Hamming distance)

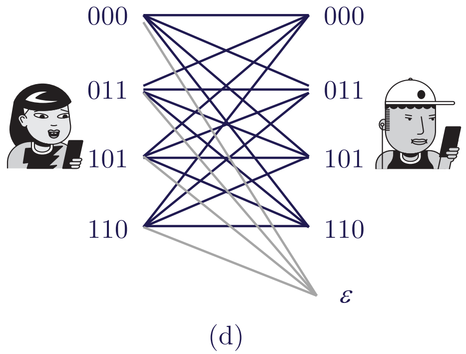
$$p_S = (1-p)^3 \;\; ; \;\; p_E = 3p^2(1-p) \;\; ; \;\; p_{ERS} = 3p(1-p)^2 + p^3$$

$$G_c = \frac{2}{3}\frac{b}{l}(1-p_E)^{\frac{l}{2}}$$

Packet duration: $\frac{3l}{2} > l$ i.e. there is still some redundancy

Coding objective:
minimize redundancy, while still meeting target error

Coding beyond repetition



If $G_c = \frac{2}{3}\frac{b}{l}(1-p_E)^{\frac{l}{2}}$ holds, then we can show that it strictly outperforms simple p-channel

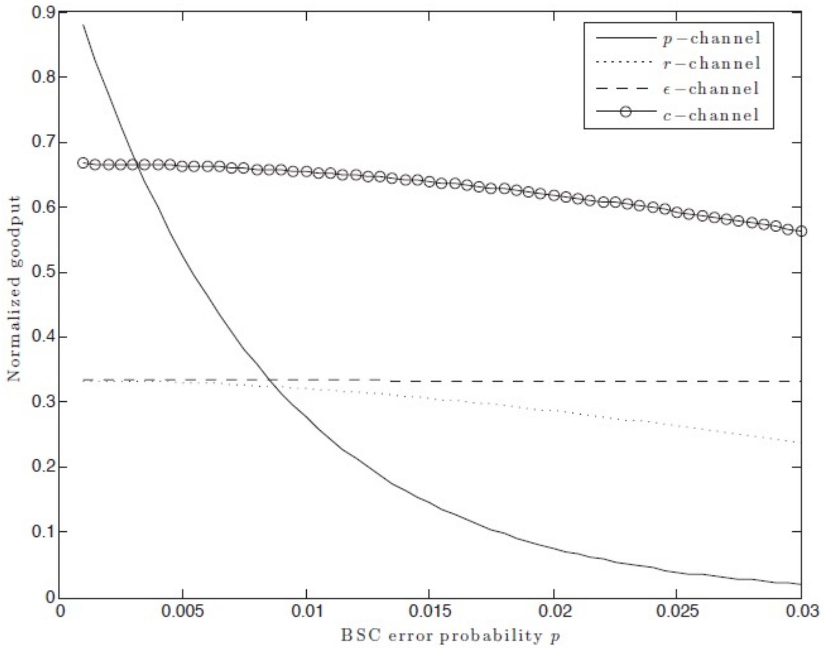
$$G_r = \frac{b}{3l}(1-3p^2(1-p)-p^3)^l < \frac{b}{3l}\big(1-3p^2(1-p)\big)^l$$

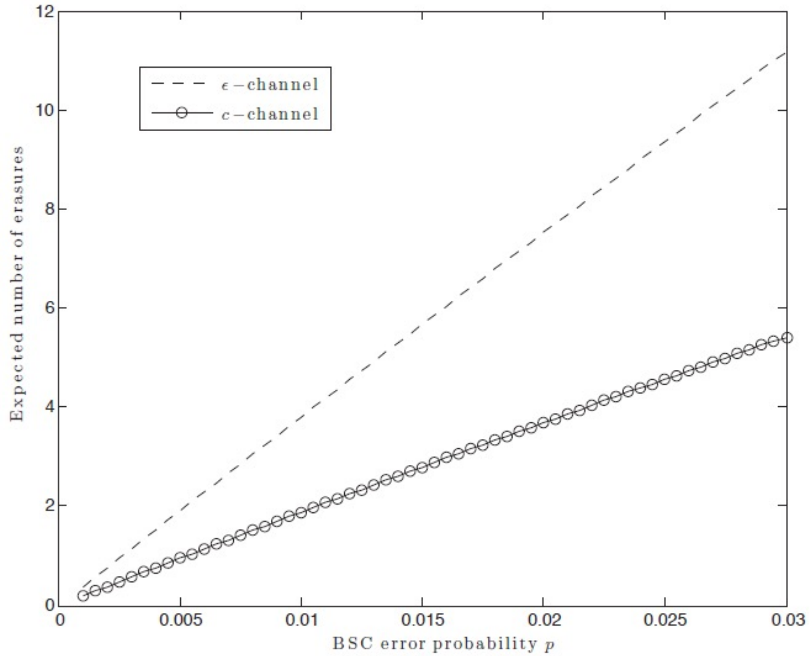
$$= \frac{b}{3l}\big(1-3p^2(1-p)\big)^{\frac{l}{2}} \cdot \underbrace{\frac{b}{3l}\big(1-3p^2(1-p)\big)^{\frac{l}{2}}}_{<1}$$

$$< \frac{b}{3l}\big(1-3p^2(1-p)\big)^{\frac{l}{2}} < G_c$$

Illustrative comparison of BSC channels

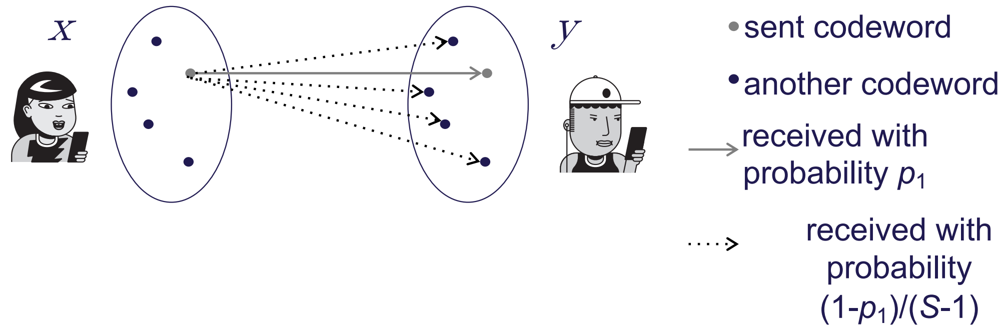
$l = 128\ bits$





Generalization of the coding idea

Unreliable **single** channel use



Uncoded transmission: map $\log_2 S$ bits into S symbols

- resulting in $l \log_2 S$ bits over l channel uses

Generalization of the coding idea

Unreliable **single** channel use

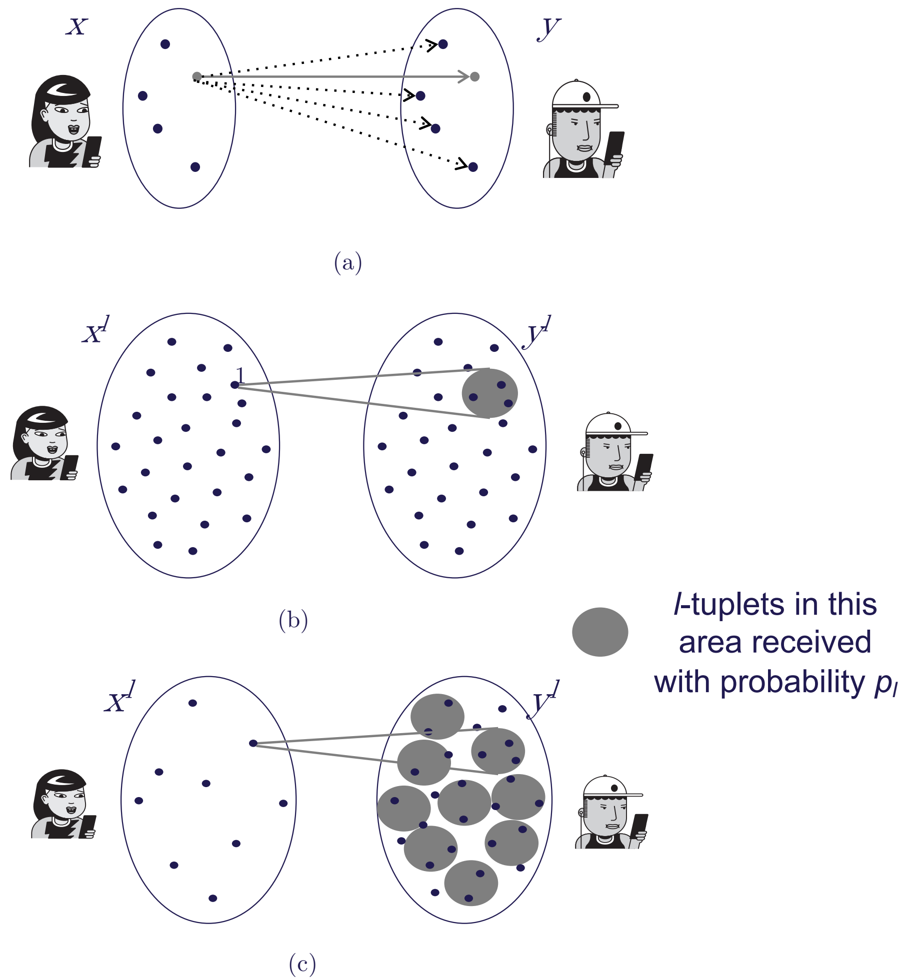
Group multiple uses to create an **expanded channel version**

Select subset of symbols representing codewords

Inherent tension:
nominal data rate vs. **reliability**

$$R = \frac{\log_2 M}{l}$$

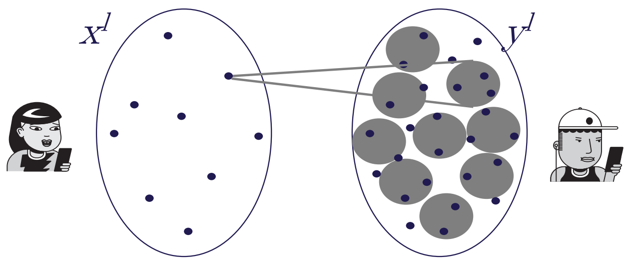
redundancy: $l - \log_S M$



Maximum likelihood (ML) decoding

Decision rule = **valid** codeword belonging to the **shaded** area

$$P(\mathbf{x}_i|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{x}_i) \cdot P(\mathbf{x}_i)}{P(\mathbf{y})} = \frac{P(\mathbf{y}|\mathbf{x}_i)}{P(\mathbf{y})}\frac{1}{M}$$



In memoryless channel: $P(\mathbf{y}|\mathbf{x}_i) = \prod_{j=1}^{l} P(y_j|x_{ij})$

- Equivalently, $\log P(\mathbf{y}|\mathbf{x}_i) = \sum_{j=1}^{l} P(y_j|x_{ij})$

Smaller Hamming distance $(\mathbf{y}, \mathbf{x})$ → higher log-likelihood

- This is related to noise clouds

Pick x that leads to **max of the possible likelihoods**

- $M \times S^l$ likelihoods to compute

Main objective: find **sufficiently separated** codewords
while keeping **reasonable** decoding **complexity**

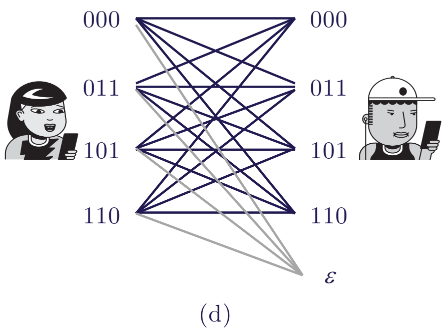
Linear block codes for the BSC

Block code: b bits supplied as a **message** $\mathbf{d} = (d_1, \ldots, d_b)$

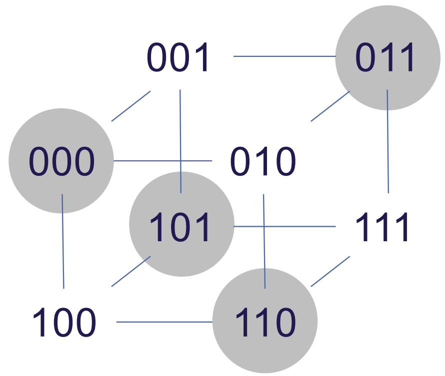
Each **message** is associated a **codeword** $\mathbf{x} = (x_1, \ldots, x_l)$

$l > b$ and both d_i and x_j assumed binary

We call this an $(\boldsymbol{l}, \boldsymbol{b})$**-code** with rate $R = \frac{b}{l}$ [bit/c.u.] and corresponds to *c*-channel with 2^b inputs and 2^l outputs



message $\mathbf{d}$	codeword $\mathbf{x}$
00	000
01	011
10	101
11	110



Linear block codes for the BSC

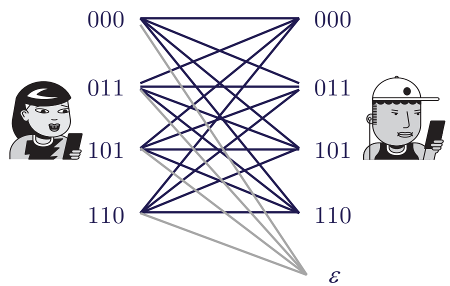
The code provided as example belongs to the class of *linear block codes*, where linearity is wrt. addition and multiplication of binary numbers, i.e., within the *GF*(2)

Encoding has particularly simple interpretation

$$\mathbf{x} = \mathbf{d} \cdot \mathbf{G}$$

Where $\boldsymbol{G} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ is the Generator matrix

In a good code **one bit** should influence **several symbols**, and **one symbol** should be influenced by **several bits**



(d)

message $\mathbf{d}$	codeword $\mathbf{x}$
00	000
01	011
10	101
11	110

Properties of linear block codes

Linearity: sum of codewords also a codeword

Hamming distance spectrum = histogram of distances

Minimal distances and **multiplicity of minimal distances**

- For linear codes, the Hamming distance spectrum for each codeword is identical

A good example, (7,4) Hamming code, given by:

systematic bits

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

With $\boldsymbol{G}$ there is associated a parity check matrix $\boldsymbol{H}$:

$\mathbf{x} \cdot \mathbf{H}^T = \mathbf{0}$

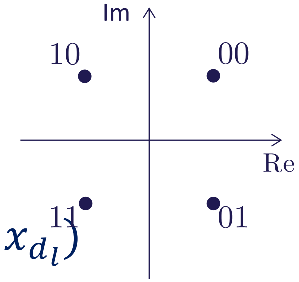
Compute *syndrome* $\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T = (\mathbf{x} \oplus \mathbf{e}) \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T$

- Can correct one-bit errors
    - However, the code cannot distinguish between single and multi-bit errors

Coded modulation as a layered subsystem

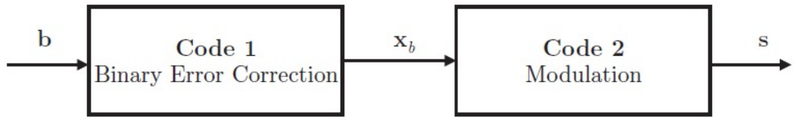
Shift to Gaussian channel → **Analog coding** (symbols are not binary anymore)

Analog coding = error control + modulation



A layered approach:

- A **message d** $= (d_1, \dots, d_b)$ is mapped to a **binary codeword** $\mathbf{x}_d = (x_{d_1}, x_{d_2}, \dots, x_{d_l})$ (introducing redundancy as $b > l$)
- It is then mapped to a **complex codeword s** $= (s_1, s_2, \dots, s_u)$, where s_i come from a set of M predefined complex values
    - Generally $M = 2^m$, then $l = m \cdot u$



This can be viewed as a concatenation of codes

Many configurations possible

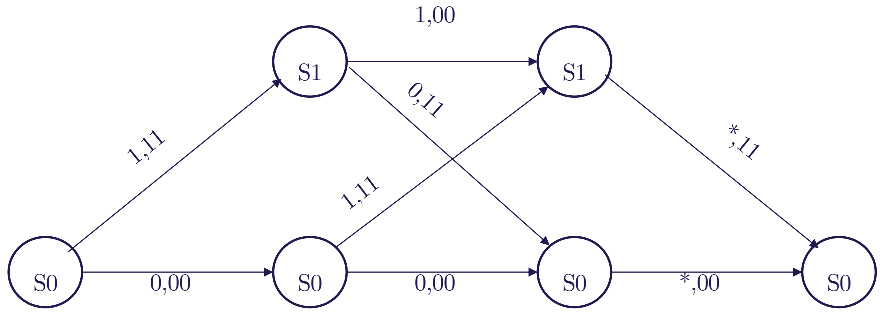
- BPSK, rate-½ FEC + QPSK, rate-¼ FEC + 16QAM → all result in 1 bit/symb

Coded modulation as a layered subsystem

What if coding and modulation are not separated, but rather done jointly?

Trellis coded modulation

message $\mathbf{d}$	codeword $\mathbf{x}_d$
00	000000
01	001111
10	111100
11	110011



Decoding the Trellis modulated signal can be done with **Viterbi** decoder, which implements the maximum likelihood decoding

Coded modulation as a layered subsystem

Strength related to **#states:**
more outputs get affected by a specific input bit – ***good code design***

Consider two bit sequences

$$\boldsymbol{d_1} = [\boldsymbol{c_1}\ 0001]$$
$$\boldsymbol{d_2} = [\boldsymbol{c_2}\ 0001]$$

Using the system that separates coding and modulation would yield $\boldsymbol{d_1} \rightarrow [\boldsymbol{S_1}\ s_1\ s_2]$ and $\boldsymbol{d_2} \rightarrow [\boldsymbol{S_2}\ s_1\ s_2]$ respectively.

Meanwhile, with trellis encoding the resulting codewords are more likely to have a form $\boldsymbol{d_1} \rightarrow [\boldsymbol{T_1}\ s_1\ s_2]$ and $\boldsymbol{d_2} \rightarrow [\boldsymbol{T_2}\ s_3\ s_4]$

- Same bit sequence can be encoded differently depending on the state

Retransmission in addition to coding

What if FEC fails?

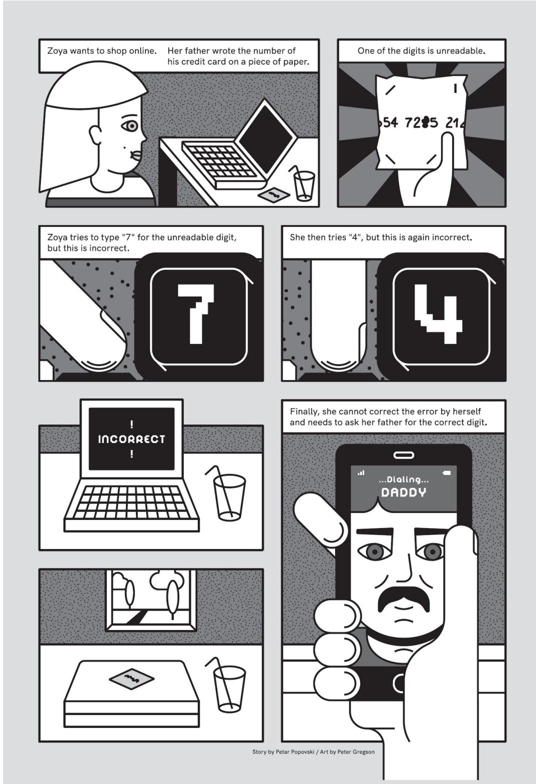
Decoded packet $\mathbf{d}_2 = \mathbf{d}_1 \oplus \mathbf{e}$

- Accidentally $\mathbf{d}_2$ satisfies the CRC
  - A simple 1-bit ACK does not help; Base station would have to send whole $\mathbf{d}_2$ back as *very rich ACK*

What if the error process produces $\mathbf{e}$ again?

- Received ACK is $\mathbf{d}_2 \oplus \mathbf{e} = \mathbf{d}_1 \oplus \mathbf{e} \oplus \mathbf{e} = \mathbf{d}_1$ satisfying error check

Impossible perfectly reliable transmission within **limited time**



Full packet retransmission

Assume ideal CRC again

Then eventually, the packet is received correctly

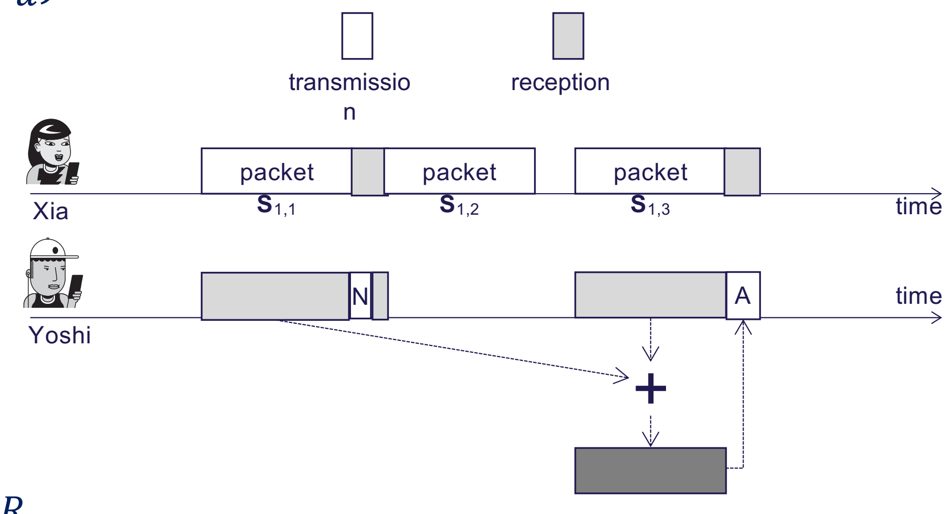
Hybrid ARQ (**HARQ**) with packet $\mathbf{s} = (s_1, s_2, \dots, s_u)$

- Basic:
  previous packet versions are discarded
- More efficient:
  combine previous packet samples

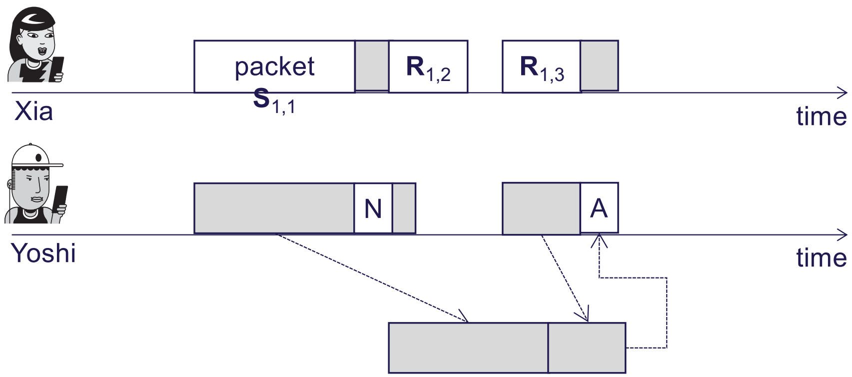
$$y_{i,1} = hs_i + n_{i,1} \;;\; y_{i,2} = hs_i + n_{i,2}$$

Chase combining (MRC):

$$y_i = y_{i,1} + y_{i,2}$$
$$= 2hs_i + n_{i,1} + n_{i,2} \qquad \rightarrow \frac{|2h|^2}{\sigma^2 + \sigma^2} = \frac{2|h|^2}{\sigma^2} = 2 \cdot SNR$$



Partial retransmission & incremental redundancy



Retransmitting **smaller set of symbols**

Yoshi signals with a *rich feedback*

How much redundancy is **further** required?

- Challenging how to signal this back

Simpler: retransmit systematic bits

- Yoshi may use the new ones, or use MRC on the retransmitted bits
- However, Xia needs to signal which bits were retransmitted

More practical: using linear codes and puncturing

- $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ sending one fragment at a time in case no ACK
- Could be coupled with adaptation of power

Recall AMC – incremental redundancy can act as a bridge between the “stairs”

Outlook and takeaways

- Using the channel multiple times allows us to construct another channel with more possible inputs and outputs
  - Restricting the subset of possible inputs is a source of redundancy, but allows to introduce error detecting and correcting capabilities
- Practical systems employ coding and modulation to enable reliable and efficient communication
  - Further gains can be achieved by adopting the cross-layer design
- Feedback and retransmissions are another set of “tools” to improve the communication reliability