IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Design of Energy-Efficient LDPC Codes and Decoders

Elsa Dupraz

16/04/2019

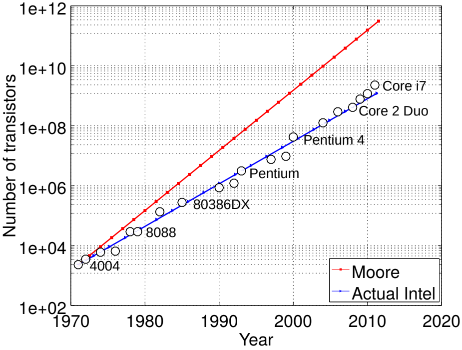
Acknowledgments

- ANR JCJC project EF-FECtive (January 2018 - December 2020)

EF-FECtive

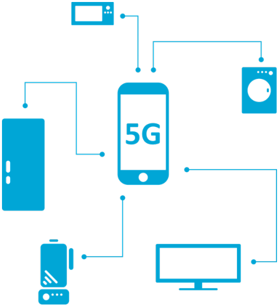
- Fangping Ye, Mohamed Yaoumi, Zeina Mheich
- François Leduc-Primeau, David Declercq, Valentin Savin, Bane Vasic, Lav Varshney, Emanuel Popovici, Frederic Guilloud ...

- In 1965, Moore predicted that the number of transistors on processors was going to double every 2 years



- What about energy consumption ?

In the 5G standardization process

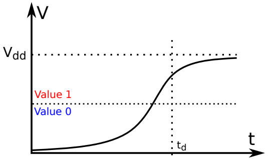


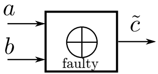
- Huge increase of number of users, terminals, etc.
- Need to improve environmental footprint, battery lifetime

- Hardware energy consumption has become a major issue

- Energy consumption can be reduced by
  - Aggressive voltage scaling
  - Increased sampling frequency

- Problem : this may introduce faults in the computation operations





$$p_{\text{xor}} = P(\tilde{c} \neq a \oplus b)$$

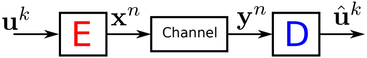
- In this talk, focus on channel coding

- Noisy vs Faulty
- Family of error-correction codes : LDPC codes

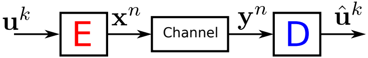
- In this talk, focus on channel coding

- Noisy vs Faulty
- Family of error-correction codes : LDPC codes

- Objectives
  - Study the effect of faults in LDPC decoders
  - Design fault-tolerant LDPC decoders



- Channel : $P(Y|X)$



- Channel : $P(Y|X)$

Encoding

$\mathbf{u}^k$: information sequence (k)
G : generator matrix ($n \times k$)

$$\mathbf{x}^n = G\mathbf{u}^k$$

Decoding

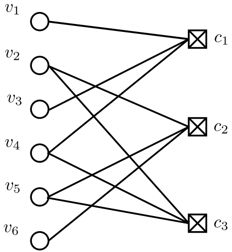
$\mathbf{x}^n$: codeword (n)
H ($n \times m$) : parity check matrix

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

LDPC codes : H sparse, optimized for good perf.

- H is a sparse parity check matrix, $H^T \mathbf{x}^n = 0$.

$$H^T = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$



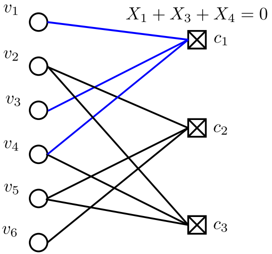
- $v_1, v_2, \cdots, v_n$: Variable Nodes (VN), degrees d_{v_i}
  $c_1, c_2, \cdots, c_m$: Check Nodes (CN), degrees d_{c_j}

- Regular codes : constant degrees d_v, d_c

$$R = \frac{k}{n} = 1 - \frac{d_v}{d_c}$$

- H is a sparse parity check matrix, $H^T \mathbf{x}^6 = \mathbf{0}^3$

$$H^T = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$



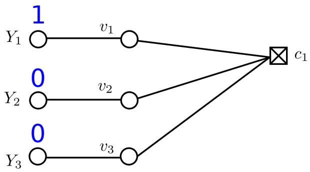
- $v_1, v_2, \cdots, v_n$: Variable Nodes (VN), degrees d_{v_i}
  $c_1, c_2, \cdots, c_m$: Check Nodes (CN), degrees d_{c_j}

- Regular codes : constant degrees d_v, d_c

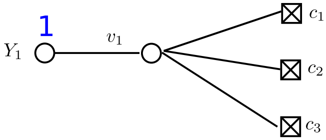
$$R = \frac{k}{n} = 1 - \frac{d_v}{d_c}$$

Ex : Gallager decoder, hard-decision decoder

CN c_1

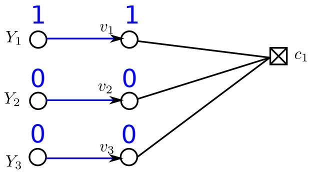


VN v_1

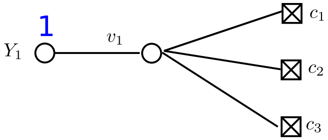


Ex : Gallager decoder, hard-decision decoder

CN c_1

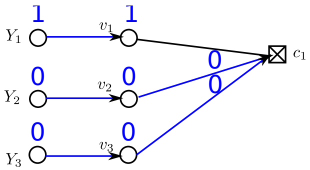


VN v_1

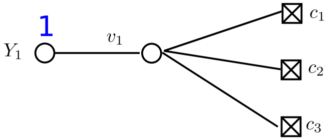


Ex : Gallager decoder, hard-decision decoder

CN c_1

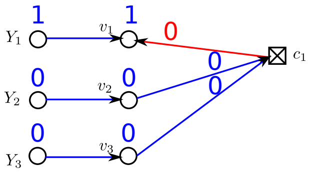


VN v_1

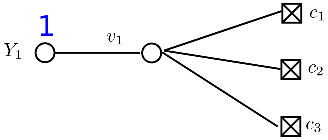


Ex : Gallager decoder, hard-decision decoder

CN c_1

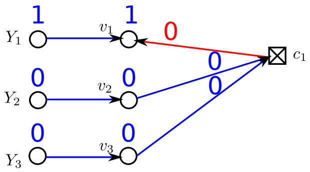


VN v_1

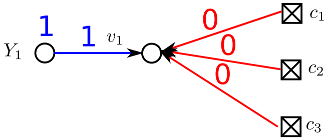


Ex : Gallager decoder, hard-decision decoder

CN c_1

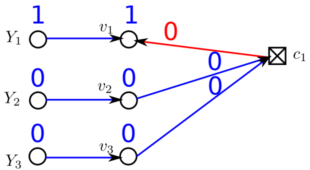


VN v_1

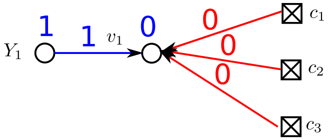


Ex : Gallager decoder, hard-decision decoder

CN c_1

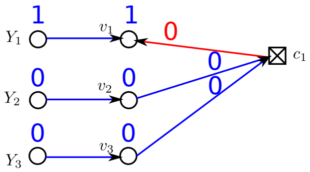


VN v_1

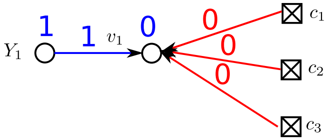


Ex : Gallager decoder, hard-decision decoder

CN c_1

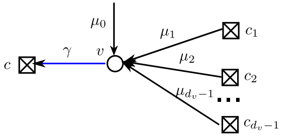


VN v_1

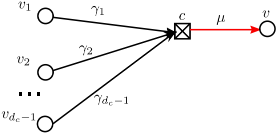


- Hard-decision decoders : binary messages
- Soft-decision decoders : LLR messages, *e.g.*, $\log \frac{P(X=0|y)}{P(X=1|y)}$

- VN update function : $\gamma^{(\ell)} = \Phi_v(\mu_0^{(\ell)}, \mu_1^{(\ell)}, \cdots, \mu_{d_v-1}^{(\ell)})$



- CN update function : $\mu^{(\ell+1)} = \Phi_c(\gamma_1^{(\ell)}, \cdots, \gamma_{d_c-1}^{(\ell)})$



- VN update function : $\gamma^{(\ell)} = \Phi_v(\mu_0^{(\ell)}, \mu_1^{(\ell)}, \cdots, \mu_{d_v-1}^{(\ell)})$
- CN update function : $\mu^{(\ell+1)} = \Phi_c(\gamma_1^{(\ell)}, \cdots, \gamma_{d_c-1}^{(\ell)})$

- VN update function : $\gamma^{(\ell)} = \Phi_v(\mu_0^{(\ell)}, \mu_1^{(\ell)}, \cdots, \mu_{d_v-1}^{(\ell)})$
- CN update function : $\mu^{(\ell+1)} = \Phi_c(\gamma_1^{(\ell)}, \cdots, \gamma_{d_c-1}^{(\ell)})$
- APP computation : $\alpha^{(\ell)} = \Phi_a(\mu_0^{(\ell)}, \mu_1^{(\ell)}, \cdots, \mu_{d_v}^{(\ell)})$
  Decide $\hat{X} = 0$ if $\alpha^{(\ell)} > 0$

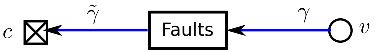
- VN update function : $\gamma^{(\ell)} = \Phi_v(\mu_0^{(\ell)}, \mu_1^{(\ell)}, \cdots, \mu_{d_v-1}^{(\ell)})$
- CN update function : $\mu^{(\ell+1)} = \Phi_c(\gamma_1^{(\ell)}, \cdots, \gamma_{d_c-1}^{(\ell)})$

- APP computation : $\alpha^{(\ell)} = \Phi_a(\mu_0^{(\ell)}, \mu_1^{(\ell)}, \cdots, \mu_{d_v}^{(\ell)})$
  Decide $\hat{X} = 0$ if $\alpha^{(\ell)} > 0$

- Hard-decision decoders : binary messages
- Soft-decision decoders : LLR messages, *e.g.*, $\mu_0 = \log \frac{P(X=0|y)}{P(X=1|y)}$

- Faulty VN update function :

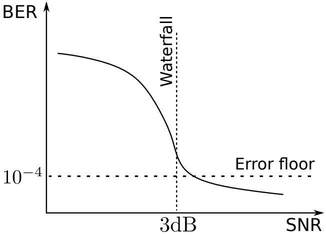
$$\gamma^{(\ell)} = \Phi_v(\mu_0^{(\ell)}, \tilde{\mu}_1^{(\ell)}, \cdots, \tilde{\mu}_{d_v-1}^{(\ell)}), \quad P(\tilde{\gamma}^{(\ell)} | \gamma^{(\ell)})$$



- Faulty CN update function :

$$\mu^{(\ell+1)} = \Phi_c(\tilde{\gamma}_1^{(\ell)}, \cdots, \tilde{\gamma}_{d_c-1}^{(\ell)}), \quad P(\tilde{\mu}^{(\ell)} | \mu^{(\ell)})$$

- Two areas of performance :



- Error Floor : Avoid short cycles in the code (PEG algorithm)
- Waterfall : Optimize the code threshold (density evolution)

Symmetry conditions [Richardson01], [Varshney11]

- Channel : $P(Y|X=0) = P(-Y|X=1)$
- VN function : $\Phi_v(-\mu_0, -\mu_1, \cdots, -\mu_{d_v-1}) = -\Phi_v(\mu_0, \mu_1, \cdots, \mu_{d_v-1})$
- CN function : $\Phi_c(b_1\gamma_1, \cdots, b_{d_c-1}\gamma_{d_c-1}) = \left(\sum_i b_i\right)\Phi_c(\gamma_1, \cdots, \gamma_{d_c-1})$
- Fault model : $P(-\tilde{\mu}|\mu) = P(\tilde{\mu}|-\mu)$

Symmetry conditions [Richardson01], [Varshney11]

- Channel : $P(Y|X=0) = P(-Y|X=1)$
- VN function : $\Phi_v(-\mu_0, -\mu_1, \cdots, -\mu_{d_v-1}) = -\Phi_v(\mu_0, \mu_1, \cdots, \mu_{d_v-1})$
- CN function : $\Phi_c(b_1\gamma_1, \cdots, b_{d_c-1}\gamma_{d_c-1}) = (\sum_i b_i)\,\Phi_c(\gamma_1, \cdots, \gamma_{d_c-1})$
- Fault model : $P(-\tilde{\mu}|\mu) = P(\tilde{\mu}|-\mu)$

Examples

- BSC : $\alpha = P(Y=0|X=1) = P(Y=1|X=0)$
- $\Phi_v(\mu_0, \mu_1, \cdots, \mu_{d_v-1}) = \sum \mu_i$

Symmetry conditions [Richardson01], [Varshney11]

- Channel : $P(Y|X=0) = P(-Y|X=1)$
- VN function : $\Phi_v(-\mu_0, -\mu_1, \cdots, -\mu_{d_v-1}) = -\Phi_v(\mu_0, \mu_1, \cdots, \mu_{d_v-1})$
- CN function : $\Phi_c(b_1\gamma_1, \cdots, b_{d_{c-1}}\gamma_{d_{c-1}}) = (\sum_i b_i)\, \Phi_c(\gamma_1, \cdots, \gamma_{d_{c-1}})$
- Fault model : $P(-\tilde{\mu}|\mu) = P(\tilde{\mu}|-\mu)$

Symmetry conditions [Richardson01], [Varshney11]

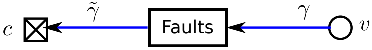
- Channel : $P(Y|X=0) = P(-Y|X=1)$
- VN function : $\Phi_v(-\mu_0, -\mu_1, \cdots, -\mu_{d_v-1}) = -\Phi_v(\mu_0, \mu_1, \cdots, \mu_{d_v-1})$
- CN function : $\Phi_c(b_1\gamma_1, \cdots, b_{d_c-1}\gamma_{d_c-1}) = (\sum_i b_i)\, \Phi_c(\gamma_1, \cdots, \gamma_{d_c-1})$
- Fault model : $P(-\tilde{\mu}|\mu) = P(\tilde{\mu}| - \mu)$

All-zero codeword assumption [Richardson01], [Varshney11]

- The decoder performance does not depend on the codeword $\mathbf{x}^n$
- All-zero codeword assumption : $\mathbf{x}^n = 0$

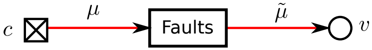
- VN messages probability distributions

$$P(\gamma^{(\ell)}|X=0),\ P(\tilde{\gamma}^{(\ell)}|X=0)$$



- CN messages probability distributions

$$P(\mu^{(\ell)}|X=0),\ P(\mu^{(\ell)}|X=0)$$



- **Message error probabilities** (recall LLR : $\log \frac{P(X=0|y)}{P(X=1|y)}$)

$$P_e^{n,(\ell)}(\alpha) = P(\gamma^{(\ell)} < 0 | X = 0)$$

$$\tilde{P}_e^{n,(\ell)}(\alpha, \epsilon) = P(\tilde{\gamma}^{(\ell)} < 0 | X = 0)$$

[Richardson01], [Varshney11], [Huang14], [Ngassa15], [Leduc18], etc.

- Noiseless threshold : worst channel parameter α for which

$$\lim_{n,\ell\to\infty} P_e^{n,(\ell)}(\alpha) = 0$$

[Richardson01]

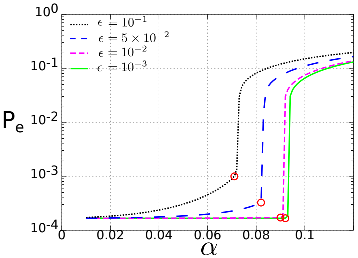
- Faulty threshold : worst channel parameter α for which

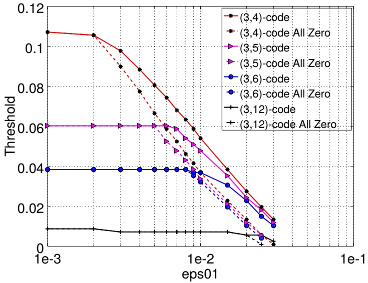
$$\lim_{n,\ell\to\infty} \tilde{P}_e^{n,(\ell)}(\alpha,\epsilon) < \eta$$

[Varshney11], [Dupraz15]

- Ex : Binary Symmetric Channel

$$\alpha = P(Y = 0|X = 1) = P(Y = 1|X = 0)$$





- Min-sum decoder with LLR messages quantized on q bits
- i.i.d. fault model, $P(\tilde{\mu}|\mu)$, $P(\tilde{\gamma}|\gamma)$

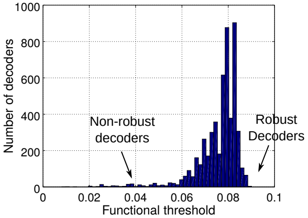
- Min-sum decoder with LLR messages quantized on q bits
- i.i.d. fault model, $P(\tilde{\mu}|\mu)$, $P(\tilde{\gamma}|\gamma)$

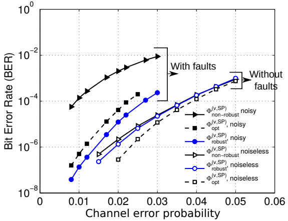
Method [Dupraz15], [Nguyen-Li16]

- A wide range of quantization functions
- Performance evaluation with faulty Density Evolution
- Optimization of the quantization function for fault-tolerance

Optimization results

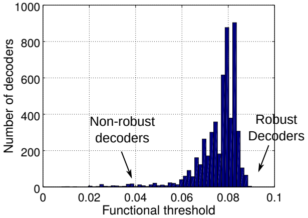
Histogram of thresholds for 5192 decoders
(different sets of quantization parameters)

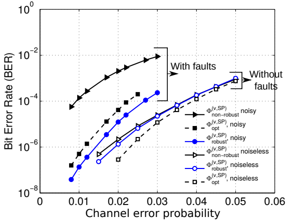




- **Optimization results**

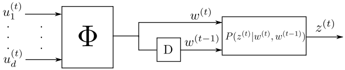
Histogram of thresholds for 5192 decoders
(different sets of quantization parameters)



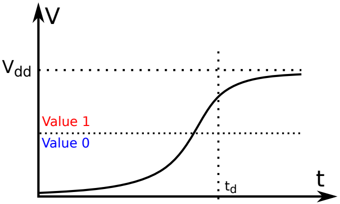


- Conclusion : Careful quantizer design is sufficient to ensure fault-tolerance

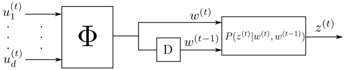
- Timing errors in the decoder [Brkic15]



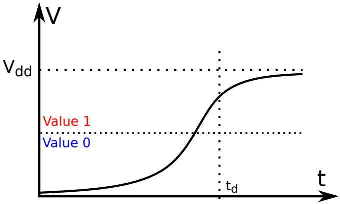
$$\begin{cases} P(z^{(t)} = w^{(t)} \,|w^{(t)}, w^{(t-1)}) & = 1 - \varepsilon, \\ P(z^{(t)} = w^{(t-1)}|w^{(t)}, w^{(t-1)}) & = \varepsilon. \end{cases}$$



- Timing errors in the decoder [Brkic15]



$$\begin{cases} P(z^{(t)} = w^{(t)} \;|w^{(t)}, w^{(t-1)}) & = 1 - \varepsilon, \\ P(z^{(t)} = w^{(t-1)}|w^{(t)}, w^{(t-1)}) & = \varepsilon. \end{cases}$$

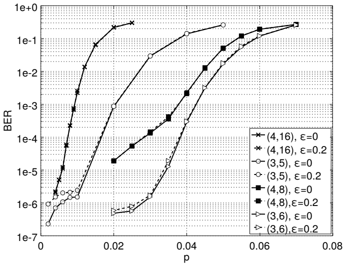


Main result [Dupraz17]

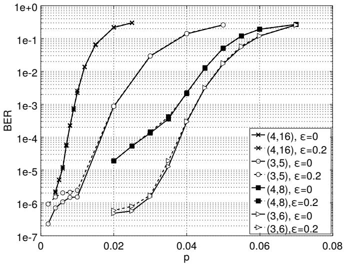
If $\lim_{\ell\to\infty} P_e^{(\ell)}(\alpha)$ exists, then $\forall\epsilon$,

$$\lim_{\ell\to\infty} \tilde{P}_e^{(\ell)}(\alpha, \epsilon) = \lim_{\ell\to\infty} P_e^{(\ell)}(\alpha)$$

- $L = 100$ iterations



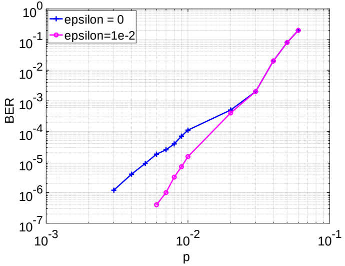
- $L = 100$ iterations



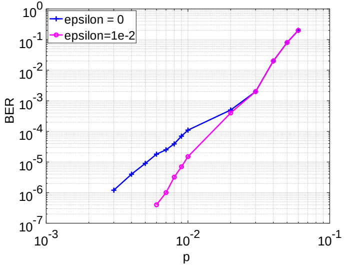
- Conclusion : Timing errors do not affect the decoder performance

- Hard-decision decoders [Sundararajan14],[Vasic15]
- Fault model $P(\tilde{\mu} = 1|\mu = 0) = P(\tilde{\mu} = 0|\mu = 1) = \epsilon$

- Hard-decision decoders [Sundararajan14],[Vasic15]
- Fault model $P(\tilde{\mu} = 1|\mu = 0) = P(\tilde{\mu} = 0|\mu = 1) = \epsilon$
- LS(155,64)-code



- Hard-decision decoders [Sundararajan14],[Vasic15]
- Fault model $P(\tilde{\mu} = 1|\mu = 0) = P(\tilde{\mu} = 0|\mu = 1) = \epsilon$
- LS(155,64)-code



- Conclusion : Faults in the decoder sometimes improve the decoder performance

Conclusions

- Density Evolution permits to analyze the performance of faulty LDPC decoders
- The robustness to faults depends on the decoder and on the fault model

Other existing works

- LDPC encoders [Hachem13],[Yang14],[Dupraz16]
- LDPC decoders for faulty computation [Grandhi16],[Yang16]
- LDPC decoders in faulty memories [Chilappagari07],[Vasic07]
- Other families of error-correction codes [Balatsoukas18]
- Machine Learning Algorithms under faulty hardware [Yang16], [Leduc18] ,[Dupraz19]

Ongoing works and Perspectives

- Energy optimization of LDPC codes and decoders [Yaoumi19]
- Realistic energy-vs-faults models
- Practical implementations
- Energy-efficient Machine Learning algorithms