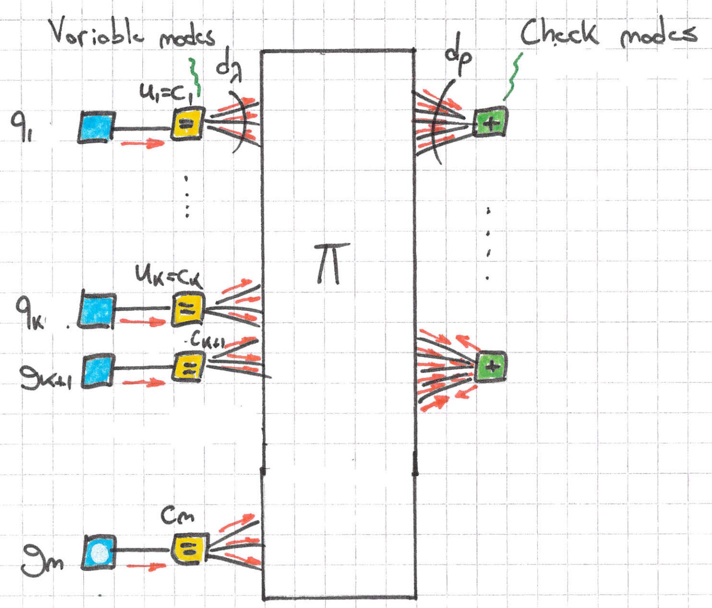
# C. LDPC Codes

$$f(\underline{u},\underline{c}) = \mathcal{M}(\underline{u},\underline{c}) \prod_\ell p(u_\ell) \prod_\ell \cancel{\frac{1}{\sqrt{2\pi\sigma_w^2}}} e^{-\frac{1}{2\sigma_w^2}(r_\ell - M(c_\ell))^2}$$

## Systematic Regular LDPC




**Initialization** $\quad \mu_{c_i \to \pi} = 1$ (Check nodes)

**Step 1** The node $\boxed{=}$ compute and output a message for each incident edge

**Step 2** The node $\boxed{+}$ compute and output a message for each incident edge

**Step 3** Iterate

**Stop** When all the check equations are satisfied or/and after a # of iterations

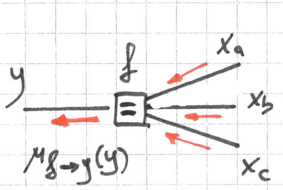
**Log Messages**

$$g_\ell(u_\ell) = \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{1}{2\pi\sigma_w^2}(r_\ell - M(c_\ell))^2} p(u_\ell)$$

$$g_\ell(c_\ell) = \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{1}{2\pi\sigma_w^2}(r_\ell - M(c_\ell))^2}$$

- Systematic LDPC $\quad c_1 = u_1, \cdots c_k = u_k$

**Variable nodes**




$$f(y, x_a, x_b, x_c) = \delta_{yx_a}\delta_{yx_b}\delta_{yx_c}$$

N

$$\underbrace{\mu_{f\to y}(y)}_{\mu_y(y)} = \sum_{x_a, x_b, x_c} \delta_{yx_a}\delta_{yx_b}\delta_{yx_c} \underbrace{\mu_{x_a\to f}(x_a)}_{\mu_a(x_a)} \underbrace{\mu_{x_b\to f}(x_b)}_{\mu_b(x_b)} \underbrace{\mu_{x_c\to f}(x_c)}_{\mu_c(x_c)}$$

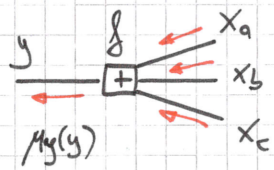
$$= \mu_a(y)\mu_b(y)\mu_c(y)$$

**Notation**

$\mu_{f\to y}(y) = \mu_y(y)$

$\mu_{x_a\to f}(x_a) = \mu_a(x_a)$

## Factor nodes




$$f(y, x_a, x_b, x_c) = \delta_{y+x_a+x_b+x_c,\,0}$$
$$= \delta_{y,\, x_a+x_b+x_c}$$
$$= \delta_{x_a,\, y+x_b+x_c}$$

$$\mu_y(y) = \sum_{x_a, x_b, x_c} \delta_{x_a,\, y+x_b+x_c}\, \mu_a(x_a)\, \mu_b(x_b)\, \mu_c(x_c)$$
$$= \sum_{x_b, x_c} \mu_a(y+x_b+x_c)\, \mu_b(x_b)\, \mu_c(x_c)$$

## Logarithmic version

**Log likelihood ratio** $\quad LLR_y = \ln \dfrac{\mu_y(0)}{\mu_y(1)}$

$$\mu_y(1) = 1 - \mu_y(0) \longrightarrow e^{LLR_y} = \frac{\mu_y(0)}{1 - \mu_y(0)}$$

$$e^{LLR_y} - \mu_y(0)\, e^{LLR_y} = \mu_y(0)$$

Using LLR need to be store/transmit 1 value instead of 2

**Inversion role**
$$\begin{cases} \mu_y(0) = \dfrac{e^{LLR_y}}{1 + e^{LLR_y}} \\[2ex] \mu_y(1) = \dfrac{1}{1 + e^{LLR_y}} \end{cases}$$

## Leaf messages

$$g_\ell(u_\ell) = \frac{1}{\sqrt{2\pi\sigma_w^2}}\, e^{-\frac{1}{2\sigma_w^2}(r_\ell - M(c_\ell))^2}$$

- $$LLR_{g_\ell \to u_\ell} = \ln \frac{\frac{1}{\sqrt{2\pi\sigma_w^2}}\, e^{-\frac{1}{2\sigma_w^2}(r_\ell+1)^2}}{\frac{1}{\sqrt{2\pi\sigma_w^2}}\, e^{-\frac{1}{2\sigma_w^2}(r_\ell-1)^2}}$$

$$= \frac{1}{2\sigma_w^2}\left(-r_\ell^2 - 1 - 2r_\ell + r_\ell^2 + 1 - 2r_\ell\right)$$

$$= \boxed{-\frac{2r_\ell}{\sigma_w^2}}$$

## Variable nodes

$$\mu_y(y) = \prod_i \mu_i(y)$$ incoming messages

- $$LLR_y = \ln \frac{\mu_a(0)\,\mu_b(0)\,\mu_c(0)}{\mu_a(1)\,\mu_b(1)\,\mu_c(1)}$$

$$\boxed{LLR_y = \sum_i LLR_i}$$

## Check nodes

$$\mu_y(y) = \sum_{x_a, x_b, x_c \dots} \delta_{y,\, x_a + x_b + x_c \dots}\, \mu_a(x_a)\,\mu_b(x_b)\,\mu_c(x_c) \dots$$

even # of ones in $\underline{x}$

- $$LLR_y = \ln \frac{\sum_{x_a x_b x_c \dots} \delta_{0,\, x_a + x_b + x_c + \dots}\, \overbrace{\mu_a(x_a)\,\mu_b(x_b)\,\mu_c(x_c)}}{\sum_{x_a x_b x_c \dots} \delta_{1,\, x_a + x_b + x_c \dots}\, \underbrace{\mu_a(x_a)\,\mu_b(x_b)\,\mu_c(x_c)}}$$

odd # of ones in $\underline{x}$

## Example with 2 variables $x_a$ $x_b$

$$LLR_y = \ln \frac{\mu_a(0)\,\mu_b(0) + \mu_a(1)\,\mu_b(1)}{\mu_a(0)\,\mu_b(1) + \mu_a(1)\,\mu_b(0)} \Big/ \frac{\mu_a(1)\,\mu_b(1)}{\mu_a(1)\,\mu_b(1)}$$

$$= \ln \frac{\frac{\mu_a(0)\,\mu_b(0)}{\mu_a(1)\,\mu_b(1)} + 1}{\frac{\mu_a(0)}{\mu_a(1)} + \frac{\mu_b(0)}{\mu_b(1)}}$$

$$= \ln \frac{e^{LLR_a} e^{LLR_b} + 1}{e^{LLR_a} + e^{LLR_b}}$$

We can use this function

$$\phi(x) = \frac{e^x - 1}{e^x + 1} = \tanh\left(\frac{x}{2}\right)$$

$$\phi^{-1}(x) = \ln\left(\frac{1+x}{1-x}\right)$$

Odd function

$$\phi(LLR_y) = \frac{\dfrac{e^{LLR_a} e^{LLR_b} + 1}{e^{LLR_a} + e^{LLR_b}} - 1}{\dfrac{e^{LLR_a} e^{LLR_b} + 1}{e^{LLR_a} + e^{LLR_b}} + 1}$$

$$= \frac{e^{LLR_a} e^{LLR_b} - e^{LLR_a} - e^{LLR_b} + 1}{e^{LLR_a} e^{LLR_b} + e^{LLR_a} + e^{LLR_b} + 1}$$

$$= \frac{(e^{LLR_a} - 1)(e^{LLR_b} - 1)}{(e^{LLR_a} + 1)(e^{LLR_b} + 1)}$$

$$= \phi(LLR_a)\,\phi(LLR_b)$$

This is a general results that is valid with >2 variables

$$LLR_y = \phi^{-1}\left(\prod_i \phi(LLR_i)\right)$$

- $\phi$, $\phi^{-1}$ can be implemented with:

  Lookup table

  polynomial approximation

Recall that $\phi$ is odd $\left(f(-x) = -f(x)\right)$

$$LLR_i = \text{sign}(LLR_i) \cdot |LLR_i|$$

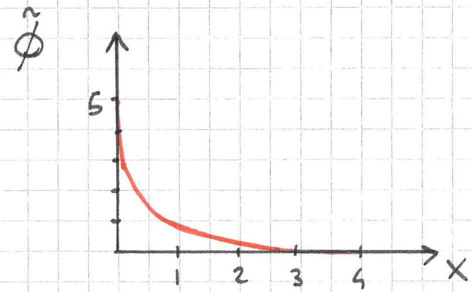
$$\boxed{LLR_y = \left(\prod_i \text{sign}(LLR_i)\right) \phi^{-1}\left(\prod_i \phi\left(|LLR_i|\right)\right)}$$

## Alternative

$$\tilde{\phi}(x) = -\ln \phi(x) \qquad x > 0$$

$$\tilde{\phi}^{-1}(x) = \tilde{\phi}(x)$$

$$\boxed{LLR_y = \prod_i \operatorname{sign}(LLR_i)\, \tilde{\phi}\left( \sum_i \tilde{\phi}(|LLR_i|) \right)}$$



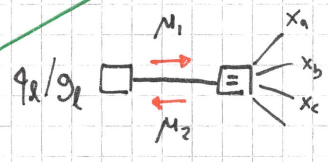

$$LLR_y = \ln \frac{M_y(0)}{M_y(1)}$$

$\downarrow$

- $M_y(0) = \dfrac{e^{LLR_y}}{1 + e^{LLR_y}}$ $\qquad M_y(0) \propto e^{LLR_y}$
- $M_y(1) = \dfrac{1}{1 + e^{LLR_y}}$ $\qquad M_y(1) \propto 1$

$$\hat{u}_\ell = \arg\max_{u_\ell} M_1(u_\ell) M_2(u_\ell)$$

$$= \arg\max_{u_\ell} h(u_\ell)$$




$M_1 = q_\ell(u_\ell)$ or $g_\ell(u_\ell)$
that is the info from the channel $(r_\ell)$

$M_2$ = Extrinsic info

$M_2 = M_a M_b M_c \cdots$

$\Rightarrow M_1 M_2 =$ Product of all the messages entering $\boxed{=}$

$$h(u_\ell) = \begin{cases} e^{LLR_1 + LLR_2} & u_\ell = 0 \\ 1 & u_\ell = 1 \end{cases}$$

$$\boxed{\hat{u}_\ell = \begin{cases} 0 & LLR_1 + LLR_2 > 0 \\ 1 & \text{otherwise} \end{cases}}$$

The same formula is used to compute $\hat{c}_\ell$ needed to check the stop condition

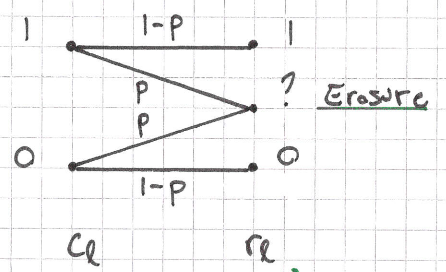
$H\hat{c} = 0$ if satisfied the decoding is correct

I can use this as a condition on when should I stop iterate

9-05-2012

## III.4 Performance analysis of LDPC codes

### Binary erasure channel (BEC)




hard decoding of the received symbol

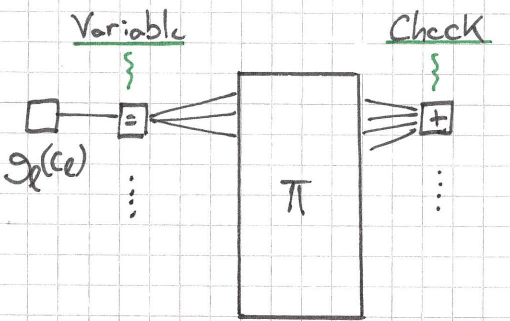
In this channel there are

**No ERRORS**

If we get an erasure ? we are not confident with that value ~ it can be correct or wrong

- BEC can be The Internet → When you receive a pkt if CRC is not valid you know that there is an error
  if CRC is valid the pkt doesn't contain errors

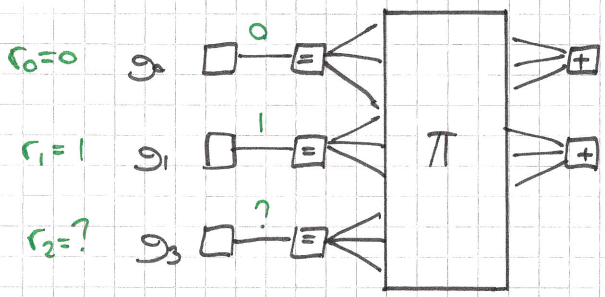
Results obtained on the BEC channel can be extended to the AWGN channel




$$g_\ell(c_\ell) = p(r_\ell | c_\ell)$$

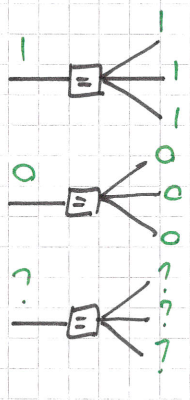
$$LLR_{g_\ell} = \ln \frac{g_\ell(0)}{g_\ell(1)} = \ln \frac{p(r_\ell|0)}{p(r_\ell|1)} = \begin{cases} \ln \frac{0}{1-p} = -\infty & r_\ell = 1 \\ \ln \frac{1-p}{0} = +\infty & r_\ell = 0 \\ \ln \frac{p}{p} = 0 & r_\ell = ? \end{cases}$$

# Decoding procedure




Instead of putting message on the arc variable we can put directly the value

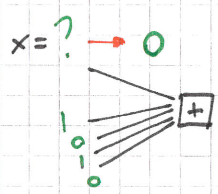
## Rule 1 (Variable nodes)




If at least one of the incident edges is not a ? associate its value to all edges

We can put the value because if the symbol is not ? it is correct for sure

## Rule 2 (Check nodes)




If all but one of the incident variables are not ? then set the erasure to the sum mod 2 of the incident variables

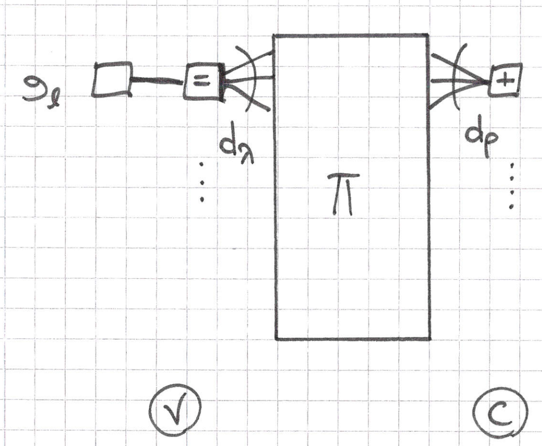
Iterate untill all the erasure become a symbol at this point the message is successfully decoded

# Performance

## Assumption

- Regular LDPC $\left\langle \begin{matrix} d_\lambda \\ d_\rho \end{matrix} \right.$
- Large block length $\left.\begin{matrix} \\ \\ \end{matrix}\right\}$ Message independence
- Random interleaver

- $q_{v \to c}$ probability that the output of a variable node is an ?
  (% of ? on variable nodes)
- $q_{c \to v}$ probability that the output of a check node is an ?




## Initialization

$$q_{c \to v} = 1$$

**Note**

$p =$ probability of getting an erasure
($P_{bit}$ of the channel)

## Variable nodes update

$$q_{v \to c} = p \cdot \left(q_{c \to v}\right)^{d_\lambda - 1} = f_1\left(q_{c \to v}\right)$$

All the incident variable must be erasure to let the variable be an erasure

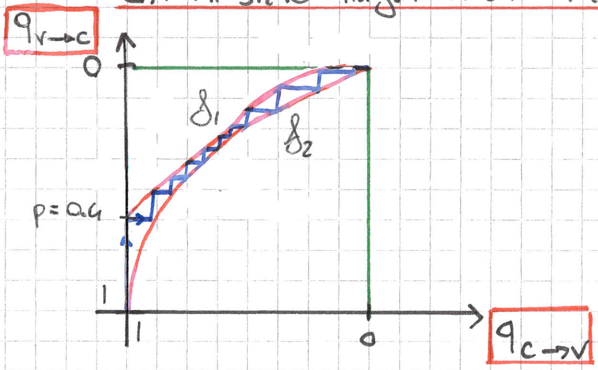
## Check nodes update

$$1 - q_{c \to v} = \left(1 - q_{v \to c}\right)^{d_\rho - 1}$$

$$q_{c \to v} = 1 - \left(1 - q_{v \to c}\right)^{d_\rho - 1} = f_2\left(q_{v \to c}\right)$$

$d_\rho - 1$ incident variable must be different from the erasure to let the output not be an erasure.

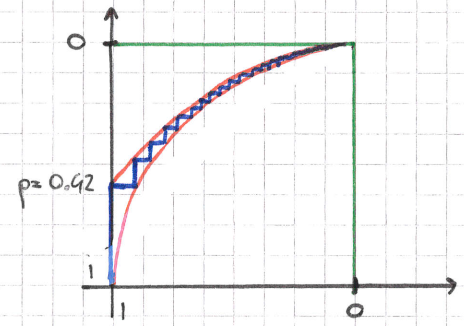
# Extrinsic Information Transfer (EXIT) Chart




It converges to the point (0,0) where the probability of having an erasure is zero.

- It takes about 20 iterations

with $p=0.4$
$d_\lambda = 3$
$d_p = 6$




- Takes too much to converge (it does not converge)


- Converge to a point different from (0,0)

$\boxed{p^* = 0.429}$

|| $p < p^*$ only in this case the convergence is possible

In the fast region:

$X = q_{v\to c},\ q_{c\to v} \simeq 0$

## Convergence speed

$$f_2(f_1(x)) = 1-(1-px^{d_\lambda-1})^{d_p-1}$$

$$f_2(x) = 1-(1-x)^{d_p-1} \simeq (d_p-1)x$$

$\Rightarrow$

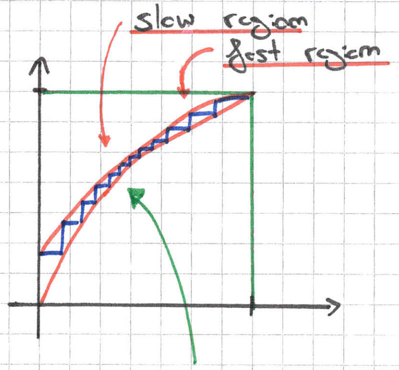
$$f_1(x) = p\,x^{d_\lambda-1}$$

$$f_2(f_1(x)) \simeq (d_p-1)px^{d_\lambda-1}$$

Gives the new X two steps later

No convergence $\left\{\begin{array}{l} \cancel{d_\lambda} = 1 \\ \cancel{d_\lambda} = 2 \end{array}\right.$

Very fast convergence $\left\{\begin{array}{l} \checkmark\ d_\lambda = 3 \\ \checkmark\ d_\lambda > 3 \end{array}\right.$




Very Narrow tunnel when $p \simeq p^*$ (near Shannon limits)

## Irregular LDPC Codes

- $f_1$ and $f_2$ can take any form

$\Rightarrow$ we can shape $f_1$ and $f_2$ to achieve better performance