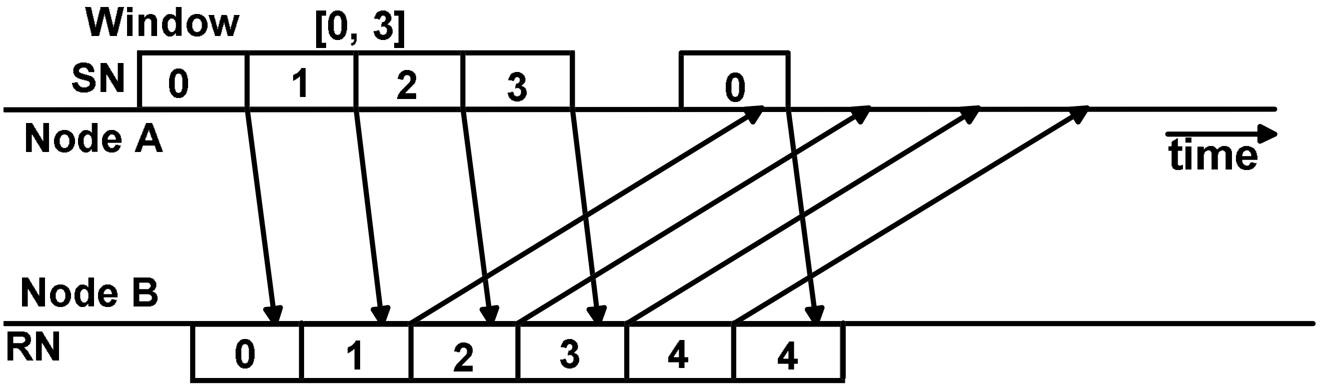
Range of RN received by B

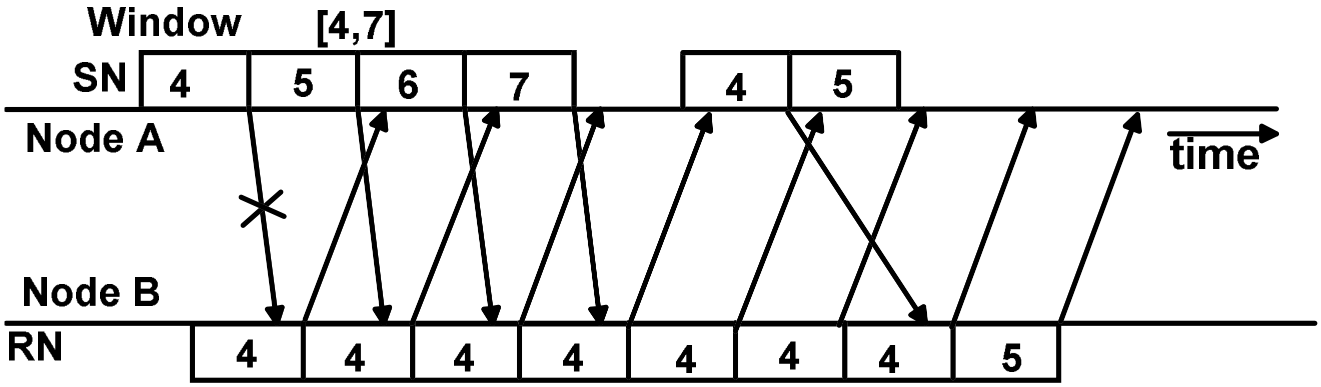
$SN_B \geq RN_B - r$

n=4



$SN_B \leq RN_B + n -$

n=4



$RN_B - n \leq SN_B \leq RN_B + n -$

Modulus m=2n for selective repeat

n=4

SN_B 0 1 2 3 4 5 6 7

RN_B 4

m=n+1=5
(go back n)

SN_B mod 5 0 1 2 3 4 0 1 2

RN_B mod 5 4

m=4
SN_B mod 4 $\underline{0}$ 1 2 3 $\underline{0}$ 1 2 3

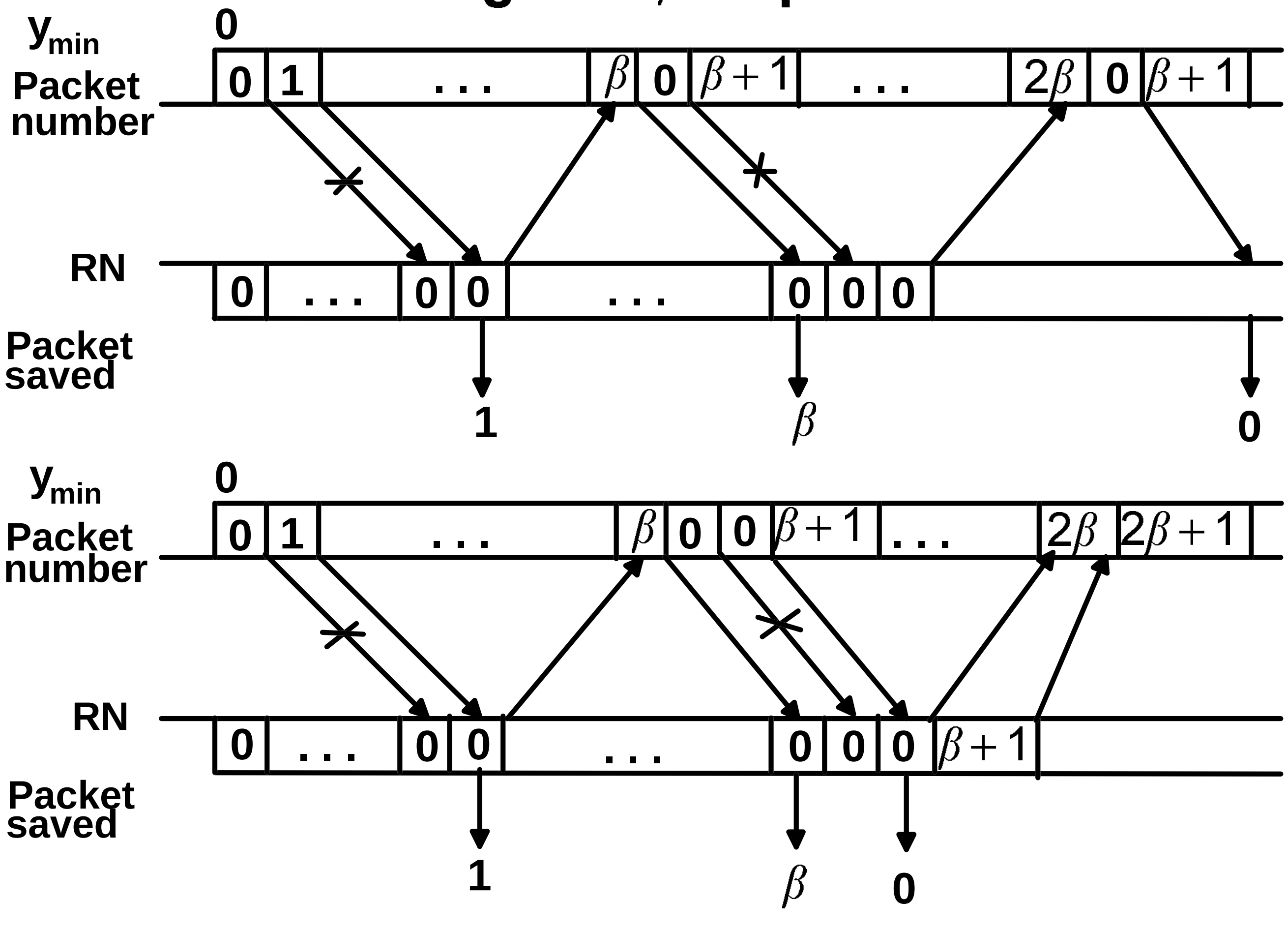
RN_B mod 4 0

m=0, 2n=8
(selective repeat)

SN_B mod 8 $\overset{\text{discarded}}{\underline{0\ 1\ 2\ 3}}\ \underset{\text{accepted}}{\underline{4}}\ \overset{\text{stored}}{\underline{5\ 6\ 7}}$

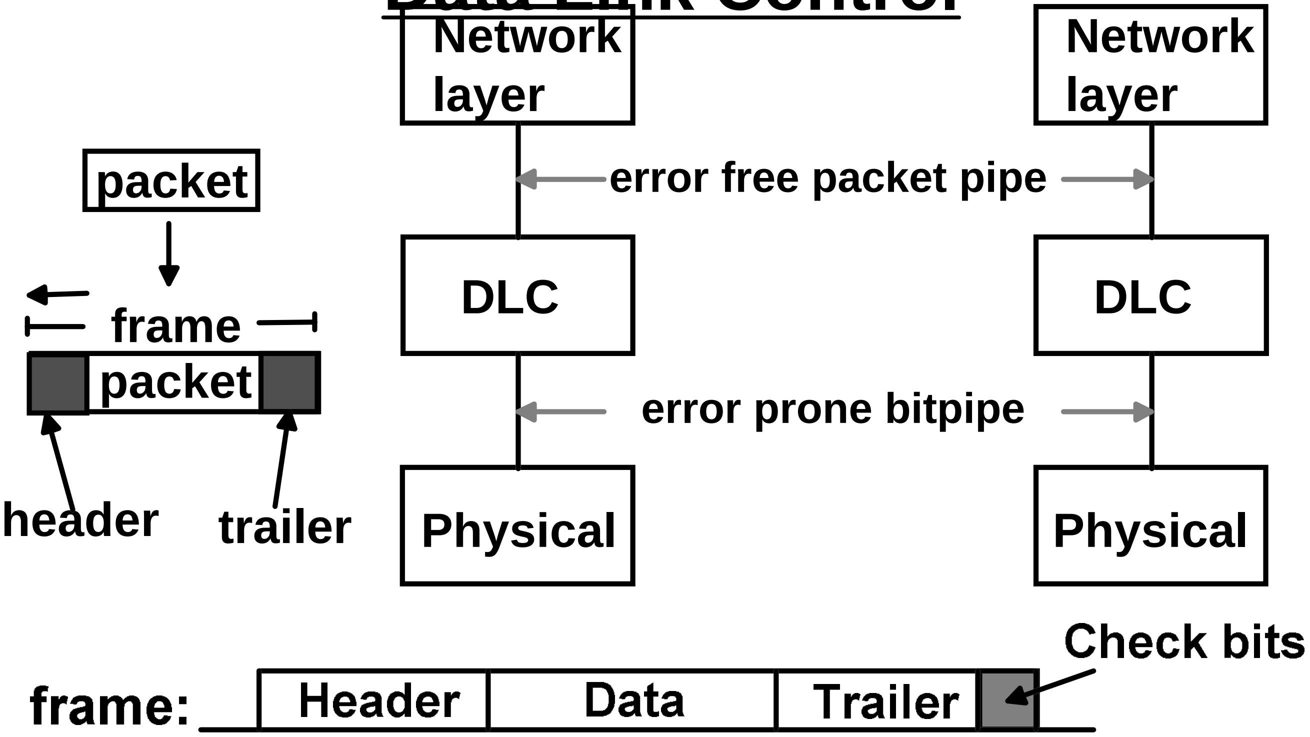
RN_B mod 8 4

Selective repeat ARQ with $n = 2\beta + 2$ and receiver storage for $\beta + 1$ packets.



Variability of delay is reduced

Data Link Control



Issues:

Detect bit errors, recover from errors by retransmission or by error correction.

Framing: determine, at the receiving DLC, the beginning and end of each frame.

Error detection of a string of bits

- **Simplest: add a parity check bit.** **(modulo 2 sum of**s_1 **to** s_K)

string	parity
1011010	**0**
$s_{k-1} \cdots s_0 s_1$	c

will catch any odd number of errors.

- **Improvement: add more parity bits.** **Ex. Horizontal & Vertical Parity Check**

$$\begin{array}{ccccccc|c} 1&0&1&0&0&1&0&1\\ 1&1&0&1&0&1&0&0\\ 0&1&1&0&1&1&0&0\\ \hline 0&0&0&1&1&1&0&1 \end{array}$$

horizontal check

vertical check

$$\begin{array}{ccccccc|c} 1&0&\cancel{1}&0&\cancel{0}&1&0&1\\ 1&1&0&1&0&1&0&0\\ 0&1&\cancel{1}&0&\cancel{1}&1&0&0\\ \hline 0&0&0&1&1&1&0&1 \end{array}$$

Will catch any pattern of 3 errors and any odd number of errors.

Effectiveness of a check code

1. Minimum distance: the minimum number of errors that may pass undetected.

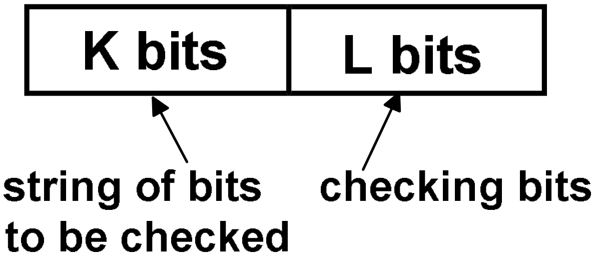
- **Single parity check: 2.**
- **Horizontal & vertical parity check: 4.**

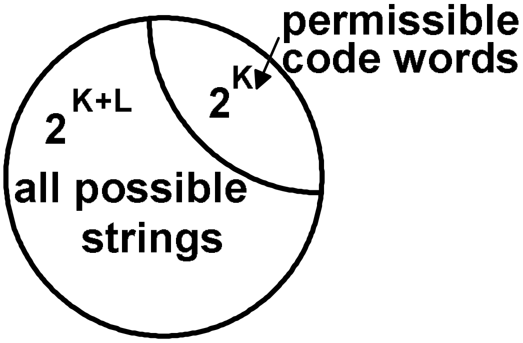
2. Burst detecting capability.

- **The length of a burst of errors is the number of bits from the first to the last error.**
- **Single parity check: 1.**
- **Horizontal & vertical parity check: the length of a row+1.**

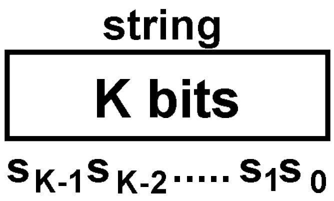
3. Probability that a random string will be accepted as error-free.

In general:





Cyclic Redundancy Check (CRC) Code



$$S(x) = s_{K-1}x^{K-1} + s_{K-2}x^{K-2} + \ldots + s_1 x + s_0$$

$C(x)$ **is the remainder of** $S(x) \cdot x^L$ **divided by a generator polynomial** $g(x)$ **.**

$$C(x) = \textbf{Remainder}\left\{\frac{S(x)\cdot x^L}{g(x)}\right\}$$

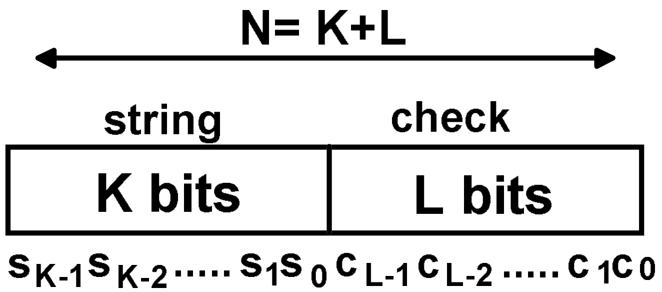
Example: string $= 101111$**, use** $L = 2$ **check bits with** $g(x) = x^2 + x$

$$
\begin{array}{l}
x^7 + \qquad x^5 + x^4 + x^3 + x^2 \\
x^7 + x^6 \\
\hline
x^6 + x^5 + x^4 + x^3 + x^2 \\
x^6 + x^5 \\
\hline
x^4 + x^3 + x^2 \\
x^4 + x^3 \\
\hline
x^2 \\
x^2 + x \\
\hline
\end{array}
\qquad
\begin{array}{|l}
x^2 + x \\
\hline
x^5 + x^4 + x^2 + 1
\end{array}
$$

Finally : $x \leftarrow$ **remainder** $\rightarrow C(x) = x$

$101111\overbrace{10}^{\textbf{CRC}}$ $\qquad c_1 = 1, c_0 = 0$

Cyclic Redundancy Check (CRC) Code



N= K+L

string

check

K bits

L bits

$s_{K-1}s_{K-2}\ldots\ldots s_1 s_0\, c_{L-1}c_{L-2}\ldots\ldots c_1 c_0$

Represent data bits and check bits by binary polynomials:

$$S(x)\cdot x^L + C(x) = s_{K-1}x^{K+L-1} + s_{K-2}x^{K+L-2} + \ldots + s_1 x^{L+1} + s_0 x^L$$
$$+ c_{L-1}x^{L-1} + c_{L-2}x^{L-2} + \ldots + c_1 x + c_0$$

e.g. data string $= 101111 \to S(x) = x^5 + 0\cdot x^4 + x^3 + x^2 + x + 1$
$(K=6)$

Entire packet: $S(x)\cdot x^L + C(x)$

	string	**check**
e.g.	101101	10
	$(K=6)$	$(L=2)$

Polynomial corresponding to entire packet:

$$S(x)\cdot x^L + C(x) = \underbrace{x^7 + 0\cdot x^6 + x^5 + x^4 + x^3 + x^2}_{K=6} + \underbrace{x+0}_{L=2}$$

is divisible by g(x).

$$S(x) \cdot x^L = Q(x) \cdot g(x) + C(x)$$

$$S(x) \cdot x^L - C(x) = S(x) \cdot x^L + C(x) = Q(x) \cdot g(x)$$

- **The polynomial corresponding to the frame sent by the transmitting DLC module is divisible by the generator polynomial g(x).**

- **When the receiving DLC module receives a frame, it checks whether the polynomial corresponding to the received frame is divisible by g(x).**
  - ❖ **If the remainder $R(x) = 0$ then it decides that no error occurred.**
  - ❖ **If the remainder $R(x) \neq 0$ then it detects an error.**

False acceptance

data string

SC: 1 0 ~~1~~ 1 1 1 1 ~~0~~

M: 1 0 0 1 1 1 1 1

error: 0 0 1 0 0 0 0 1 $\rightarrow E(x) = x^5 + 1$

$e^{K+L-1} \cdot \cdot \cdot \cdot \cdot e_2 e_1 e_0$ **(error polynomial)**

$$E(x) = e^{K+L-1} x^{K+L-1} + \cdot \cdot \cdot \cdot \cdot + e_1 x + e_0$$

sent out: $S(x) \cdot x^L + C(x) \rightarrow$ **divisible by** $g(x)$

received: $S(x) \cdot x^L + C(x) + E(x) \overset{def}{=} M(x)$

Receiver finds: $R(x) \overset{def}{=}$ **remainder** $\left\{ \frac{M(x)}{g(x)} \right\}$

$= $ **remainder** $\left\{ \frac{E(x)}{g(x)} \right\}$

False acceptance when
$g(x)$ **divides** $E(x)$ **and** $E(x) \neq 0$

e.g.

if $g(x) = x^2 + x$

$E(x) = x^2 + x, x^3 + x^2$

Properties of the CRC codes

In practice $g(x)$**is chosen as**

$$g(x) = \underbrace{(x^{L-1} + \cdots\cdots + 1)}_{\text{"}primitive \quad polynomial\text{"}} \cdot (x+1)$$

$$= x^L + g_{L-1}x^{L-1} + \cdots\cdots + g_1 x + 1$$

- **Minimum distance = 4**.
- **Burst-detecting capability = L.**
- **Random strings accepted as code words: prob.**2^{-L}

Minimum distance = 4.

- **Any odd number of errors detected:**

  If E(x) is divisible by g(x)

  then $E(x) = q(x) \cdot g(x)$

  An odd number of errors

  e.g. $E(x) = x^5 + x^3 + x^2, \quad E(1) = 1$

  But $g(1) = 0$ **since** $(x + 1) = 0$

  $E(x) \neq q(x) \cdot g(x)$

- **A primitive polynomial does not divide**

  $x^n + 1$ **if** $n < 2^L - 1$

  2 errors: $E(x) = x^i + x^j = x^j(x^{x-j} + 1)$

  $i - j < K + L$ ($< 2^L$ **in practice**)

  K+L: length of frame

Burst-detecting capability = L

$$E(x) = x^{m+b-1} + a_2 x^{m+b-2} + \cdot\cdot\cdot\cdot\cdot + a_{b-1} x^{m+1} + x^m$$

$$= x^m (x^{b-1} + a_2 x^{b-2} + \cdot\cdot\cdot\cdot\cdot + a_{b-1} x + 1)$$

not divisible by $x^L + g_{L-1} x^{L-1} + \cdot\cdot\cdot\cdot\cdot + g_1 x + 1$

if $b - 1 < L$

A burst of errors with length$b \leq L$ **are detected.**

.