Lecture 5

Digital television

The forward error correction (FEC)

- Need for error correction
- Overall picture
- The different FEC methods in DVB:
  - PRBS energy dispersal (bit level coding)
  - Reed Solomon outer coder (byte/block level)
  - Outer interleaver (byte level)
  - Convolution inner coder (Viterbi decoder) (bit level)

DVB forward error correction

The Transport stream should be transmitted over a Quasi Error Free (QEF) channel

-> (Bit Error Ratio) BER $< 10^{-10}$

1 error on 10^{10} bits

Compare:

1. TCP – Error free channel by resending packes

Resending the date 3 times and voting on the correct?

Basis for error correction

Hamming distance of *x* and *y*

$$d(x,y) := | i \mid 1 \le i \le n, x_i \ne y_i \} |$$

i.e. the number of bits that are different in *x* and *y*.

x = 0111**0**0**0**1110
y = 0111**1**0**1**1110

d(x,y) = 2

Hence, we try to find a code *C* that maximizes the distance between any two codewords in *C*

Example

Parity check

Parity check bit added to codeword

```
0111   1
0101   0
```

The minimum hamming distance between any two codeword in a parity check code is 2.

Hence, the codeword length is 5 bits instead of 4 bits, but a single bit error can now be detected.

Error correction

Parity checking enables error detection. It is also possible to construct codes, that can perform error correction.

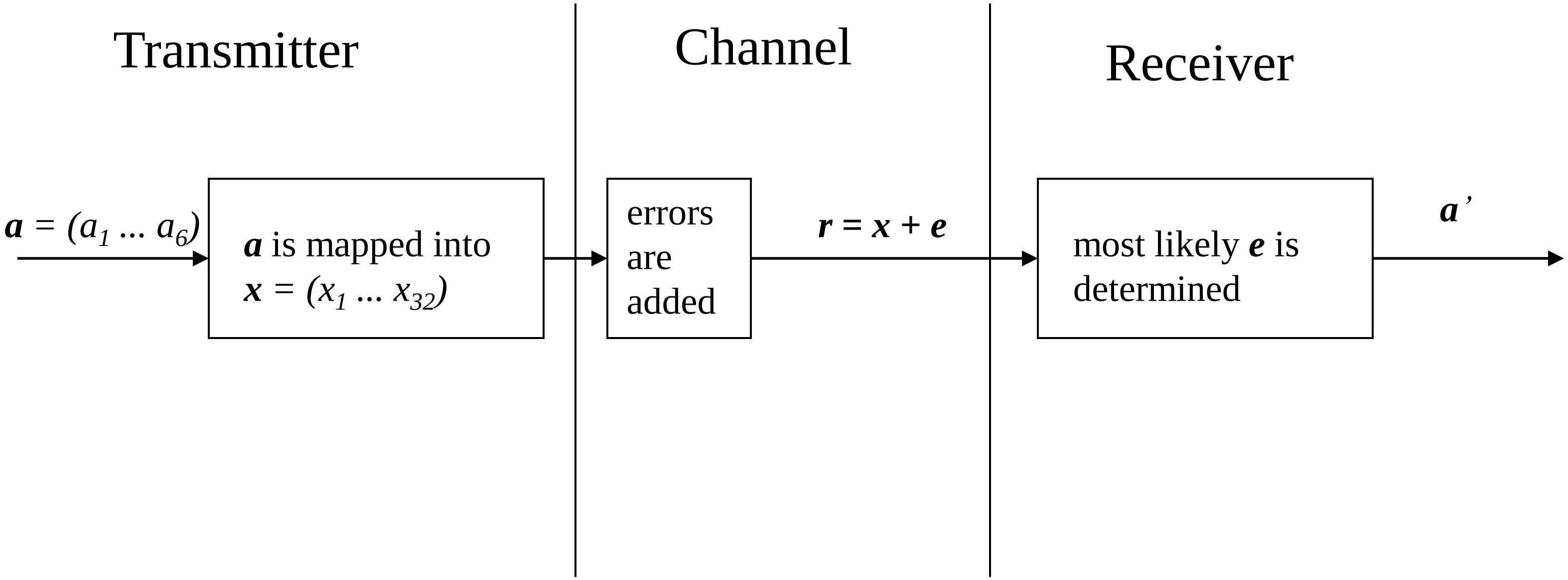
Idea: construct a code, where the Hamming distance between any two codewords is as large as possible.

For any received word *r*, compare the received word with all codewords in the code, select the codeword *x* where

$$d(x^i, r) < a$$

Code	Received	Distances	Decoded
01001	01111	2	01110
01110		1	
10011		3	
11000		4	

Forward error correction



Coding gain

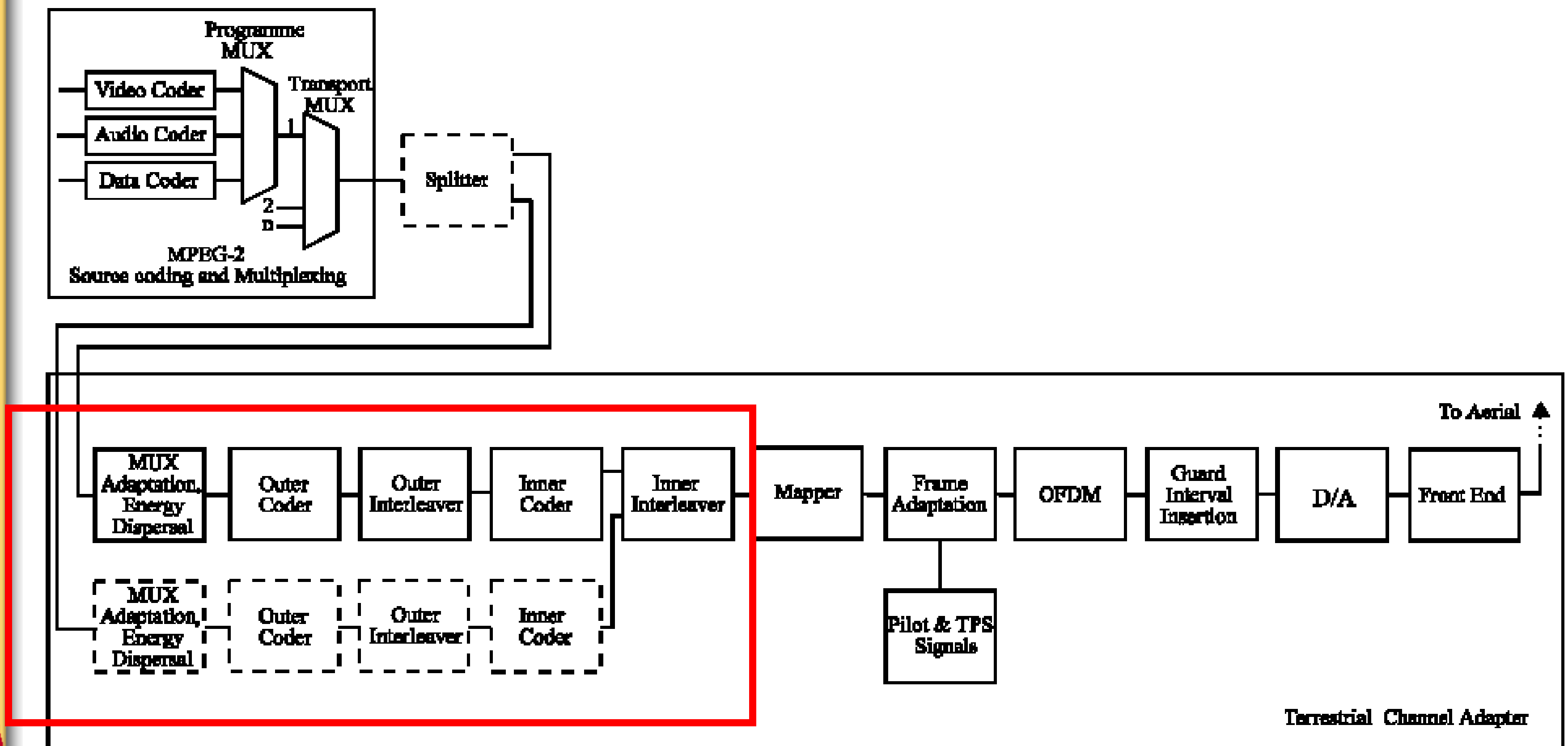
B – number of bits per second we would like to transmit

W – power used for transmitting in noisy channel

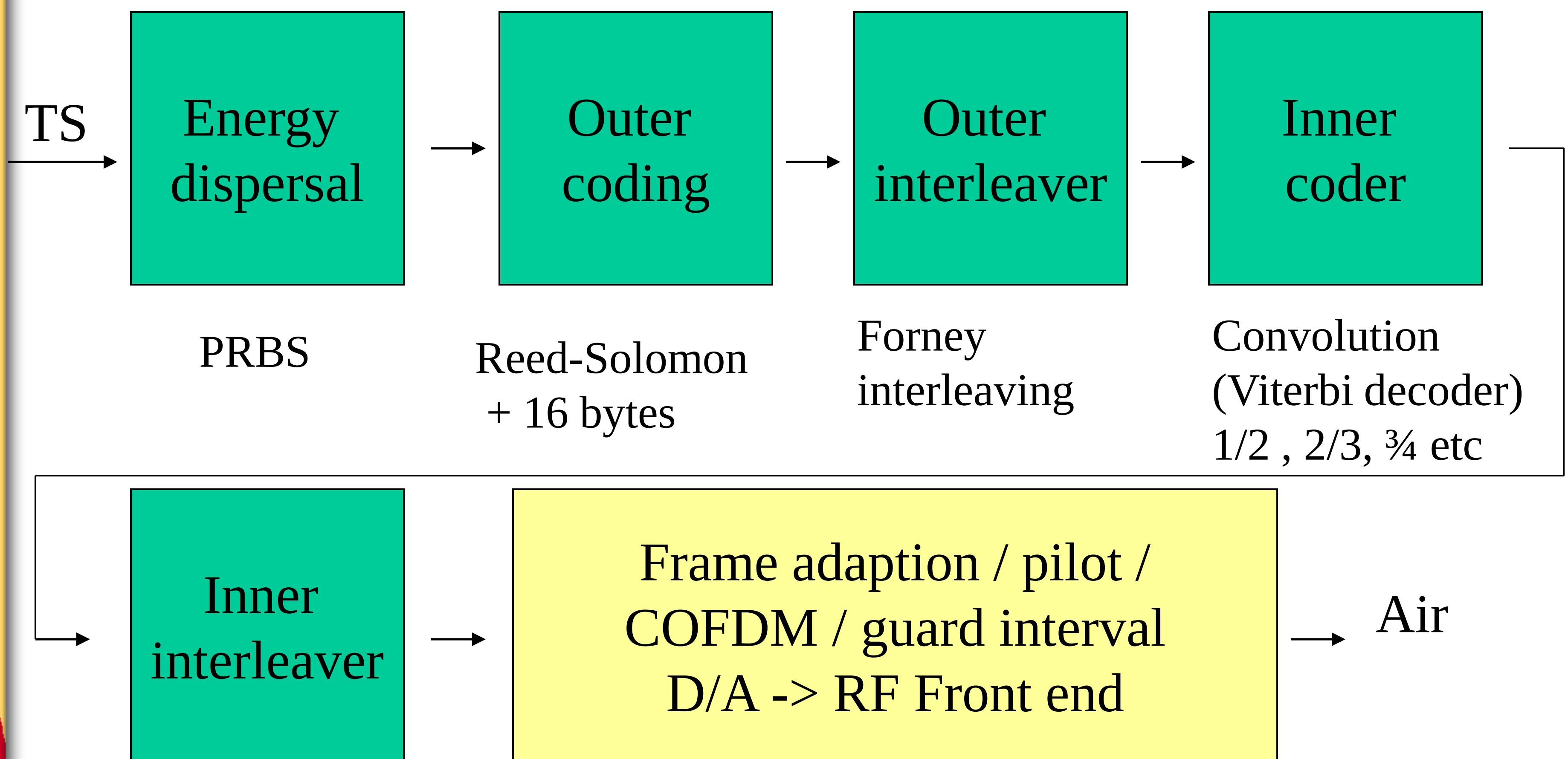
R – code rate = k/n k-original word length, n-word length

Coding only makes sense if the error rate using coding is less than without coding = *coding gain*

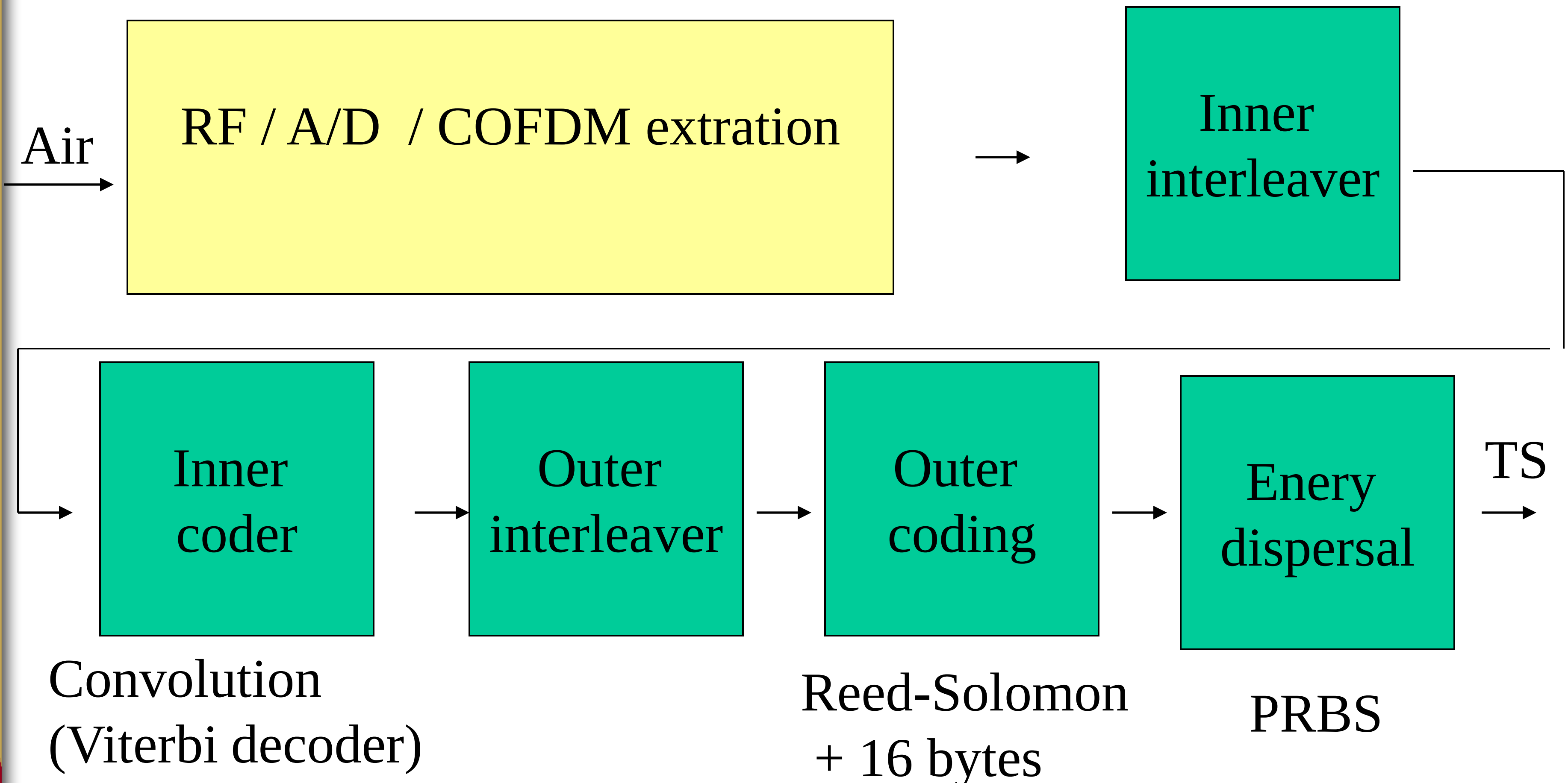
DVB overall transmission sequence

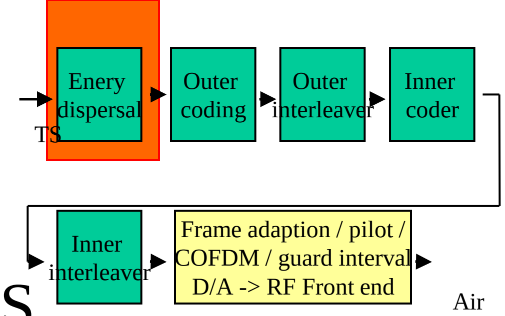


FEC overall transmission



FEC receiver end



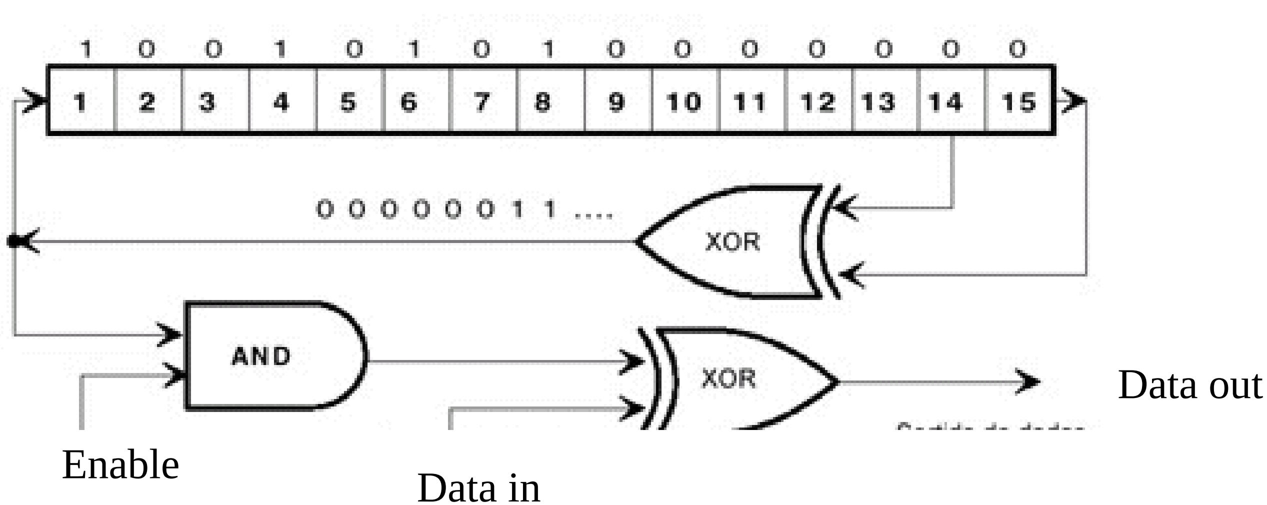


Energy dispersal

Pseudo Random Binary Sequence – PRBS

Polynomial generating the sequence:

$$1 + X^{14} + X^{15}$$

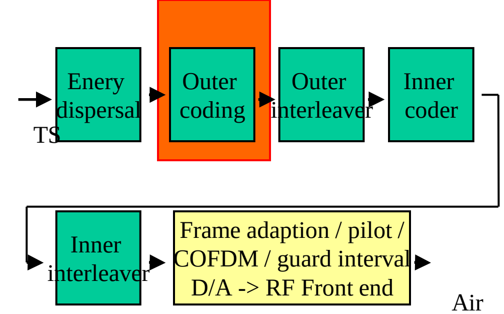


Energy dispersal example

Test: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1

PRBS: 0 0 0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 0 1 1 0
0 0 0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 0 1 1 0 0 0 0 0
0 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 0 1 0
1 0 1 0 0 1 1 1

Reed solomon



Block level code (works on bytes)

- Adds extra bytes for error correction
- Need for a block syncronization

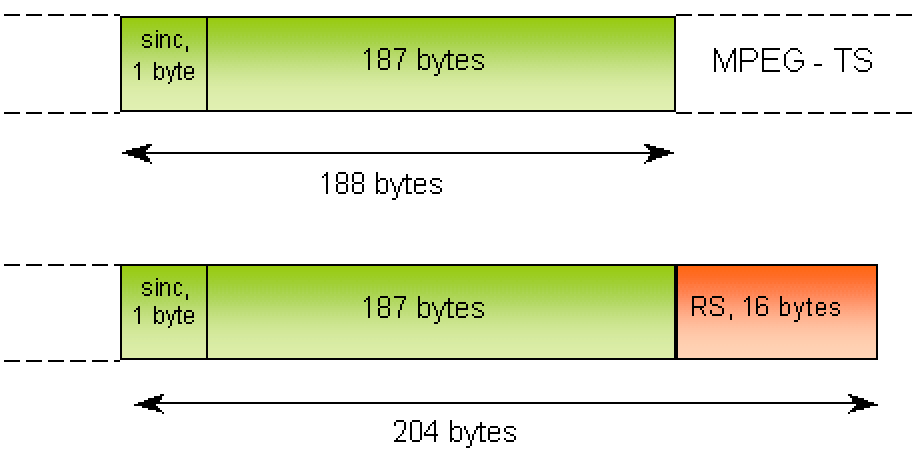
Characteristics: n, k, t

n = 204 Final transport packet length

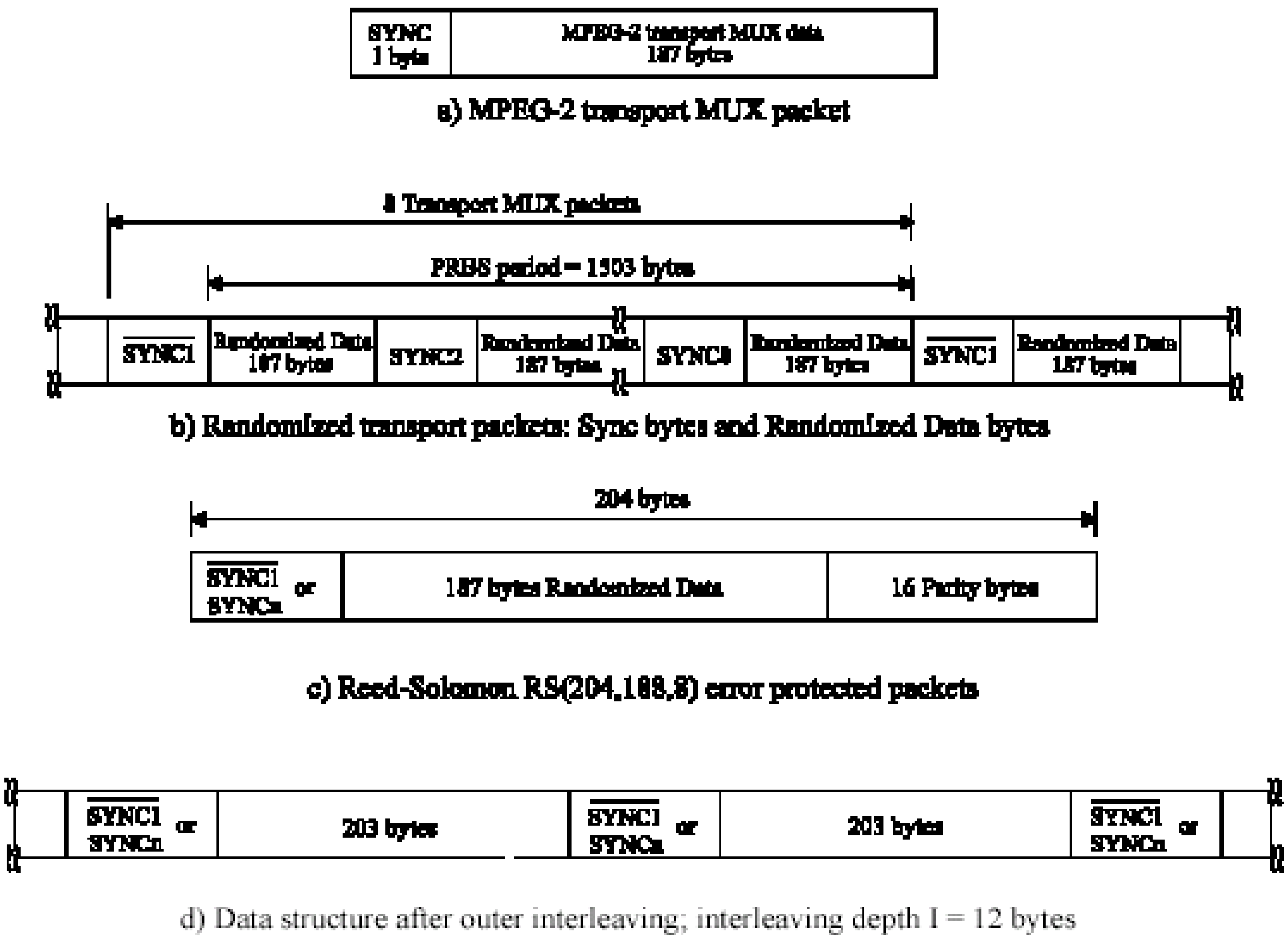
k = 188 Original correctable bytes

t = 8 Number of correctable bytes

-> RS(204, 188)



Reed Solomon



Reed Solomon mathematics

Finite (Galois) Field Arithmetic

•Reed-Solomon codes are based on a specialist area of mathematics known as Galois fields or finite fields.

•A finite field has the property that arithmetic operations (+,-,x,/ etc.) on field elements always have a result in the field.

•A Reed-Solomon encoder or decoder needs to carry out these arithmetic operations. These operations require special hardware or software functions to implement.

Reed Solomon if error?

When a codeword is decoded, there are three possible outcomes:

1. If $2s + r < 2t$ (s errors, r erasures) then the original transmitted code word **will always be recovered**,

OTHERWISE

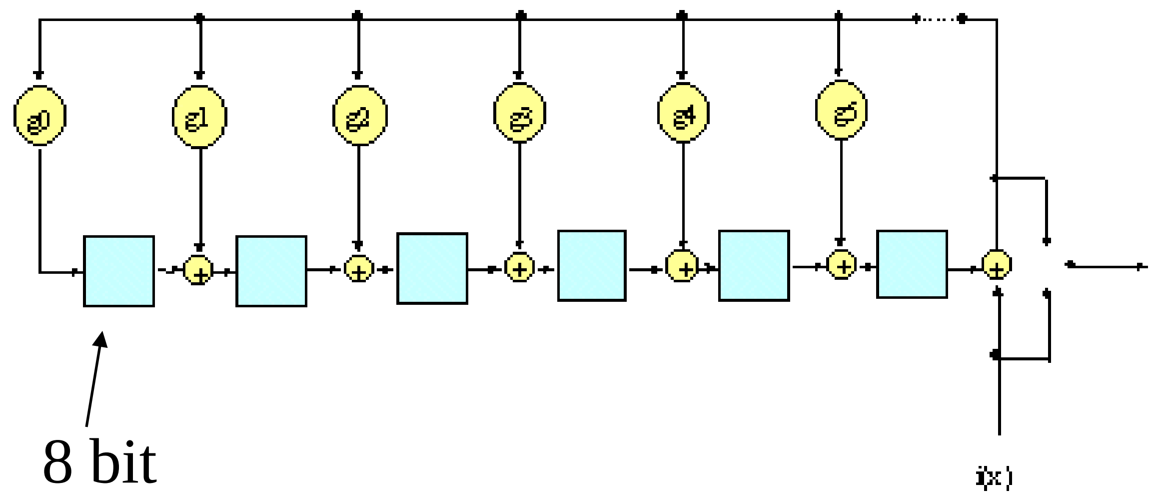
2. The decoder will detect that it cannot recover the original code word and indicate this fact.

OR

3. The decoder will mis-decode and recover an incorrect code word without any indication.

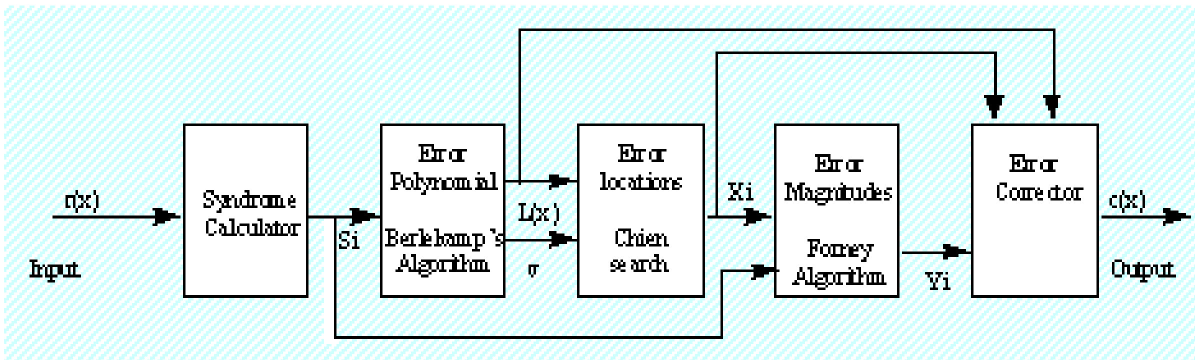
The probability of each of the three possibilities depends on the particular Reed-Solomon code and on the number and distribution of errors.

Reed Solomon encoder architecture



Finite field arithmetics

Reed Solomon decoder architecture



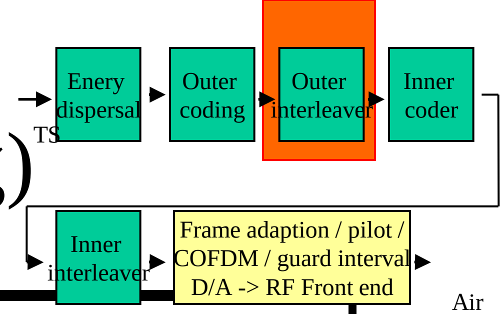
r(x)	Received codeword
Si	Syndromes
L(x)	Error locator polynomial
Xi	Error locations
Yi	Error magnitudes
c(x)	Recovered code word
v	Number of errors

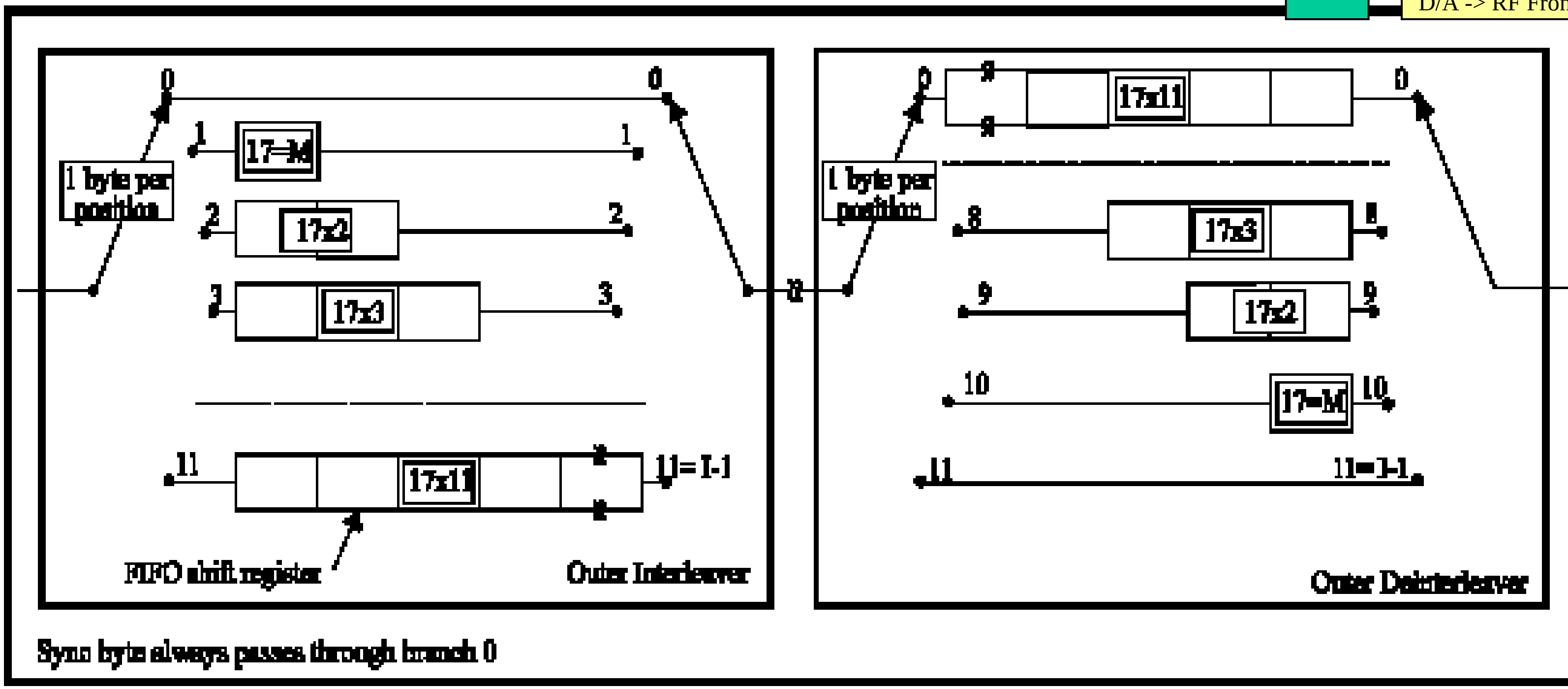
Reed Solomon in software

166 Mhz Pentium

Code	Data rate
RS(255,251)	12 Mbps
RS(255,239)	2.7 Mbps
RS(255,223)	1.1 Mbps

Outer interleaving (and deinterleaving)





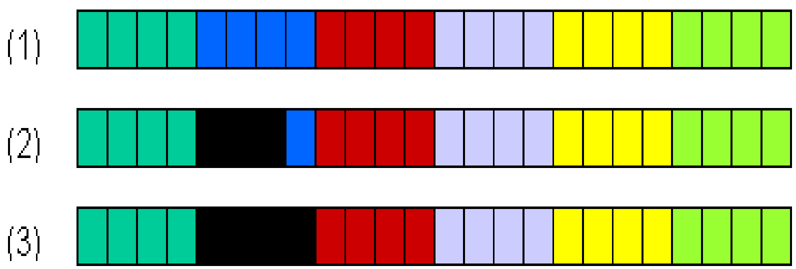
In DVB: Number of branches $i = 12$, $\{0,....,11\}$

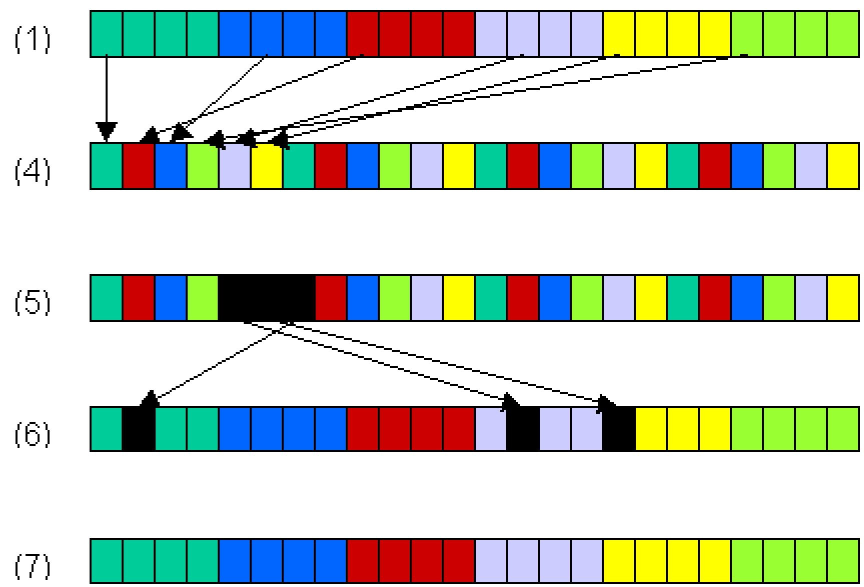
Length of packet to be protected $L = 204$

$M = (L/i) * j$ $(=17*j)$, j branch number index

Interleaving

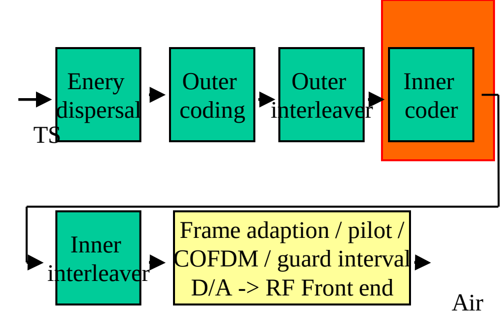
No interleaving





Viterbi + Reed Solomon

Convolution Codes

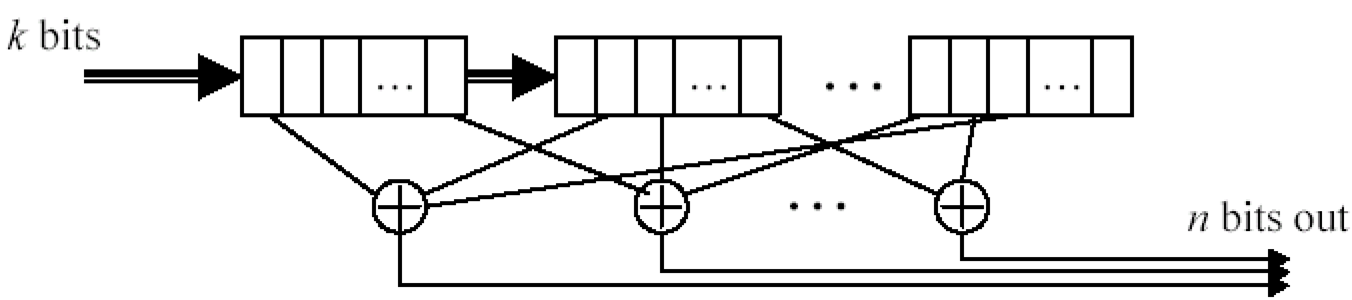


- Code words are generated by c(D) = m(D)g(D)

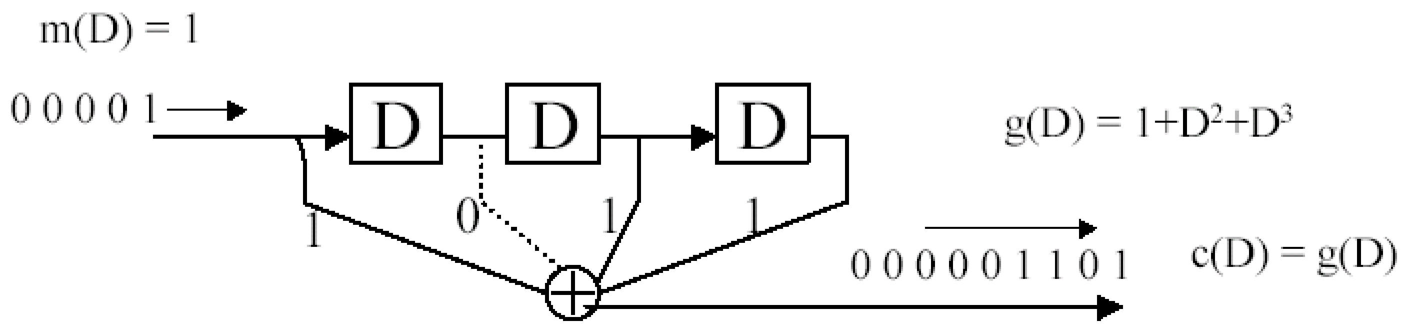
– D is one unit delay of a shift register circuit
– g(D) is realized with a linear finite-state shift register
– The degree of m(D) can be infinite. So can be that of c(D).

- Rate *k/n* code,

– *k* information bits get shifted in at each D,
– goes thru K units of delay, (K is called the *constraint length*)
– generates *n* coded bits output at each delay

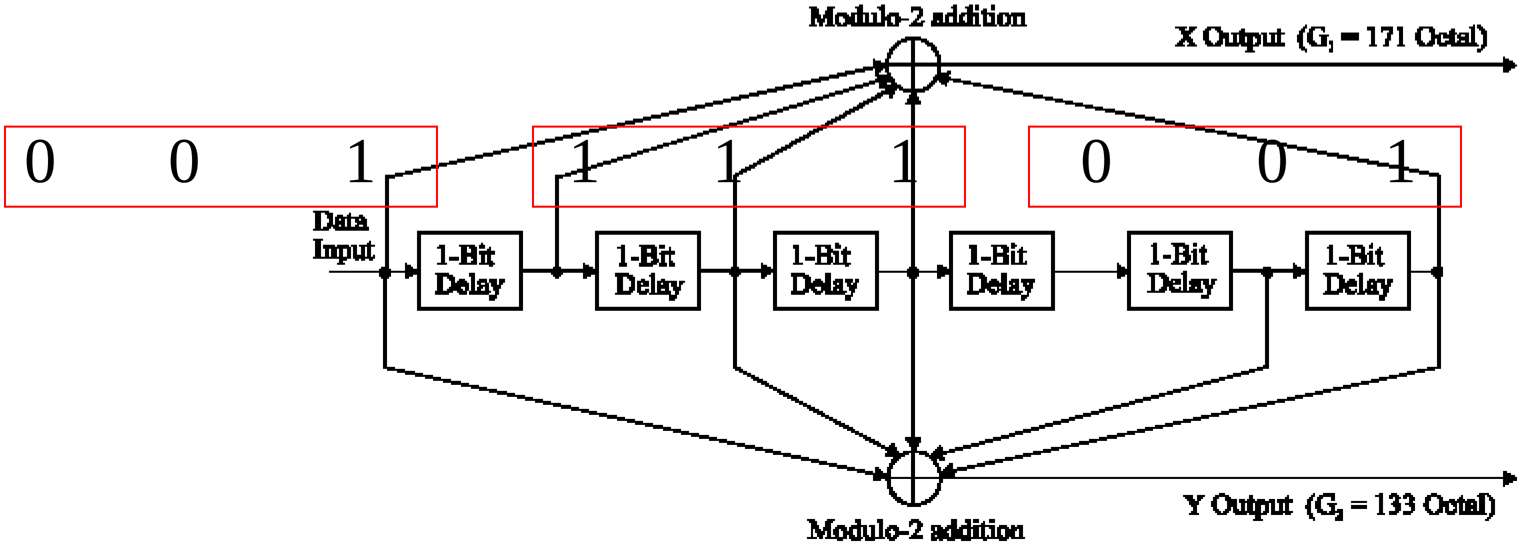


Convolution



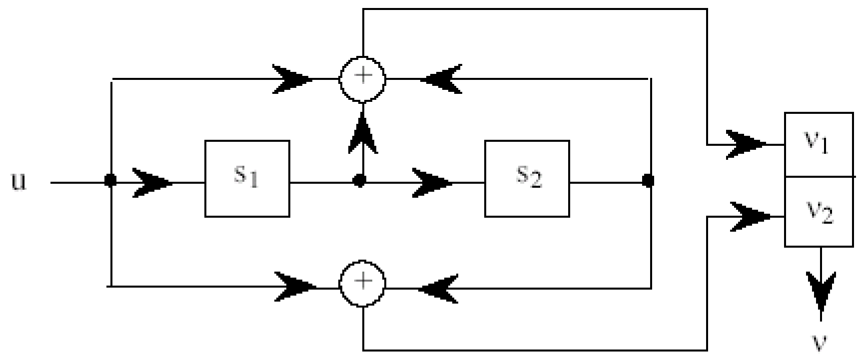
- The convolution is
  - – Shifting the input sequence
  - – Multiplying the shifted input term by term with the filter coefficients
  - – Add up the product terms
  - – Shift the input sequence by one and repeat
- Here the input and the filter coefficients are binary and the summation is module 2 addition

Convolution coding in DVB

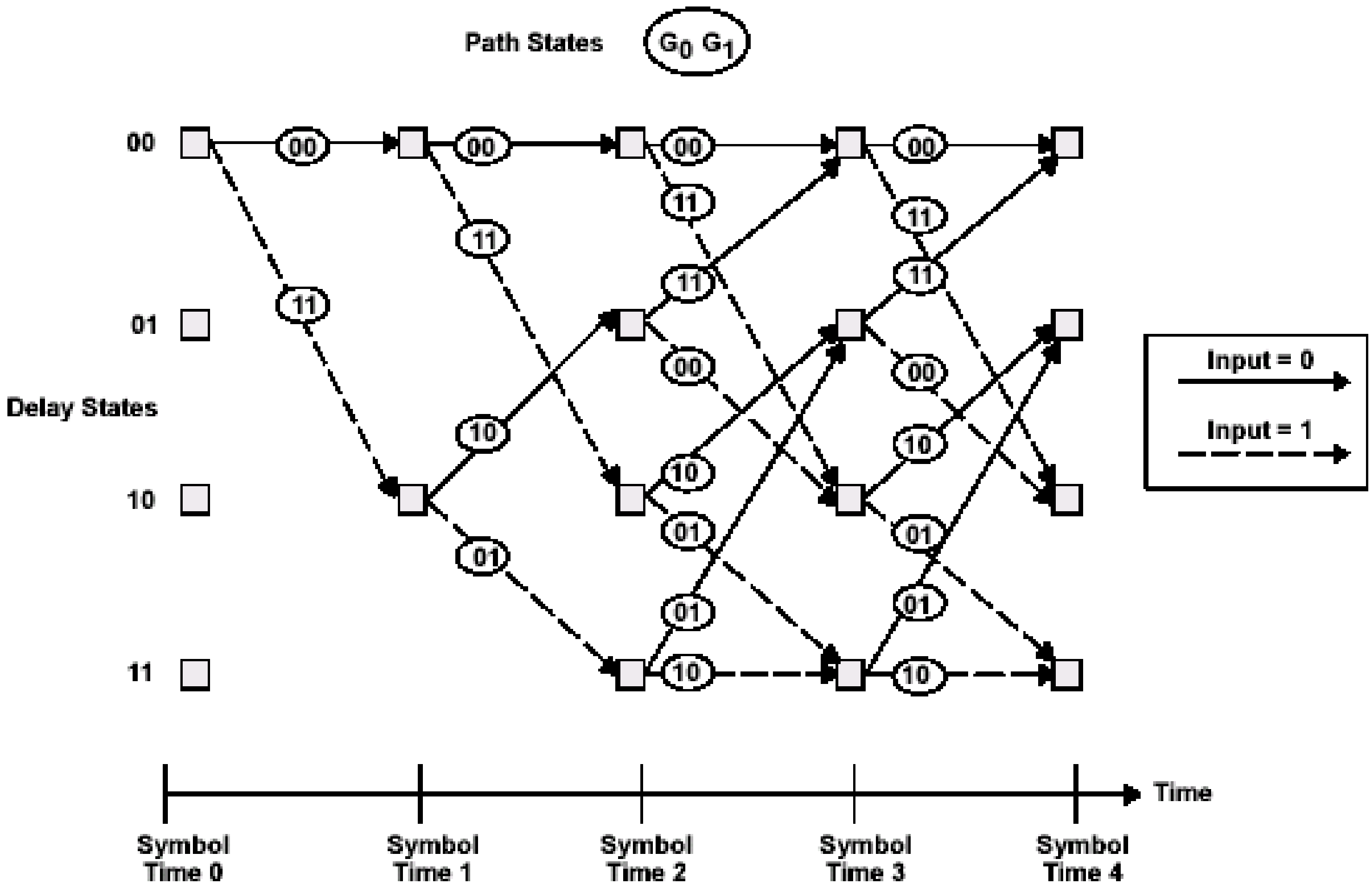


Decoding the convolution code

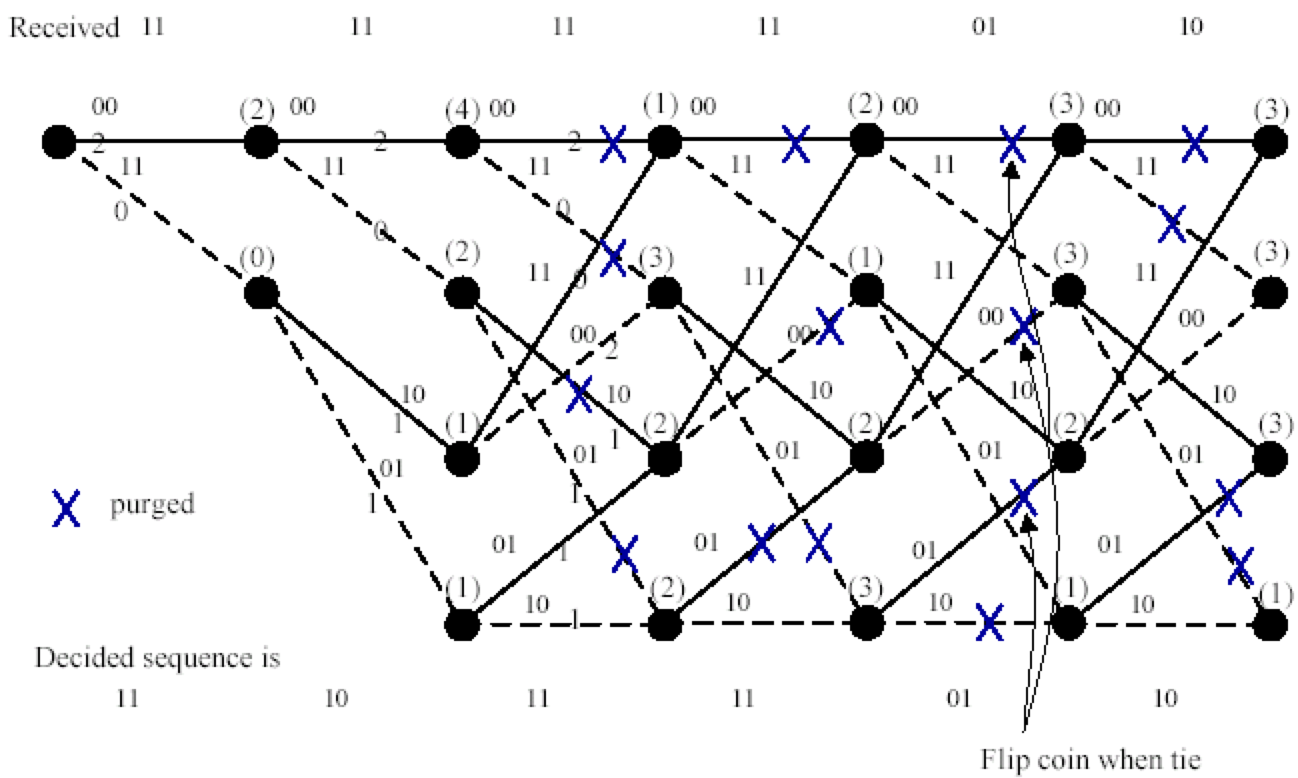
Trellis diagrams
Viterbi algorithm



Trellis diagram



Trellis decoding



Viterbi Algorithm

- Note that the number of states is 2^memory
- The number of branches is 2^(memory+1)
- Let $\sigma(j,k)$ be the partial cumulative metric at state j at k-th trellis section

1. Set $\sigma(j=a,0)=0$ (Starting at the state a)
2. At time k, compute the partial cumulative metrics for all paths merging to each state
3. Set $\sigma(j, k)$ equal to the best partial path metric entering the node corresponding to state-j at time t. Break tie with coin flipping. Mark the the best metric path
4. At the end of sequence, trace back the marked path for decoding

Bit rate calculations

Useful data bit rate = *(B x C x M x N) / T*

B efficiency of the RS block (188/204 = 0.92)
C convolution code rate (1/2, 2/3, ¾ etc)
M number of bits per carrier (2, 4, 6)
N number of carriers used (1512 for 2k, 6048 for 8k)
T total smbol duration time (894 us+28 us for 8k and 1/32 guard, 224 us + 7us for 2k mode and 1/32 guard)