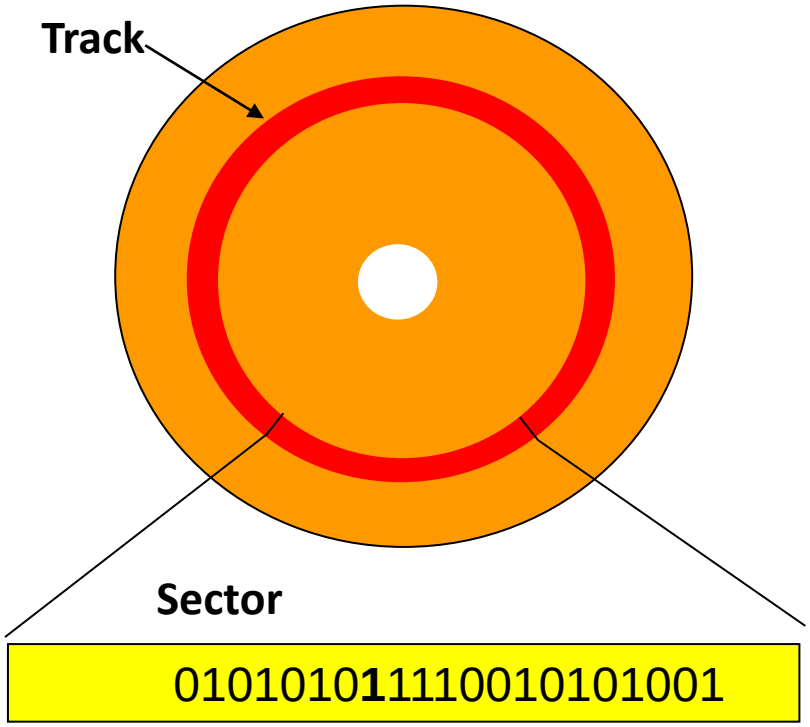
ELEC 519A
Selected Topics in Digital Communications: Information Theory

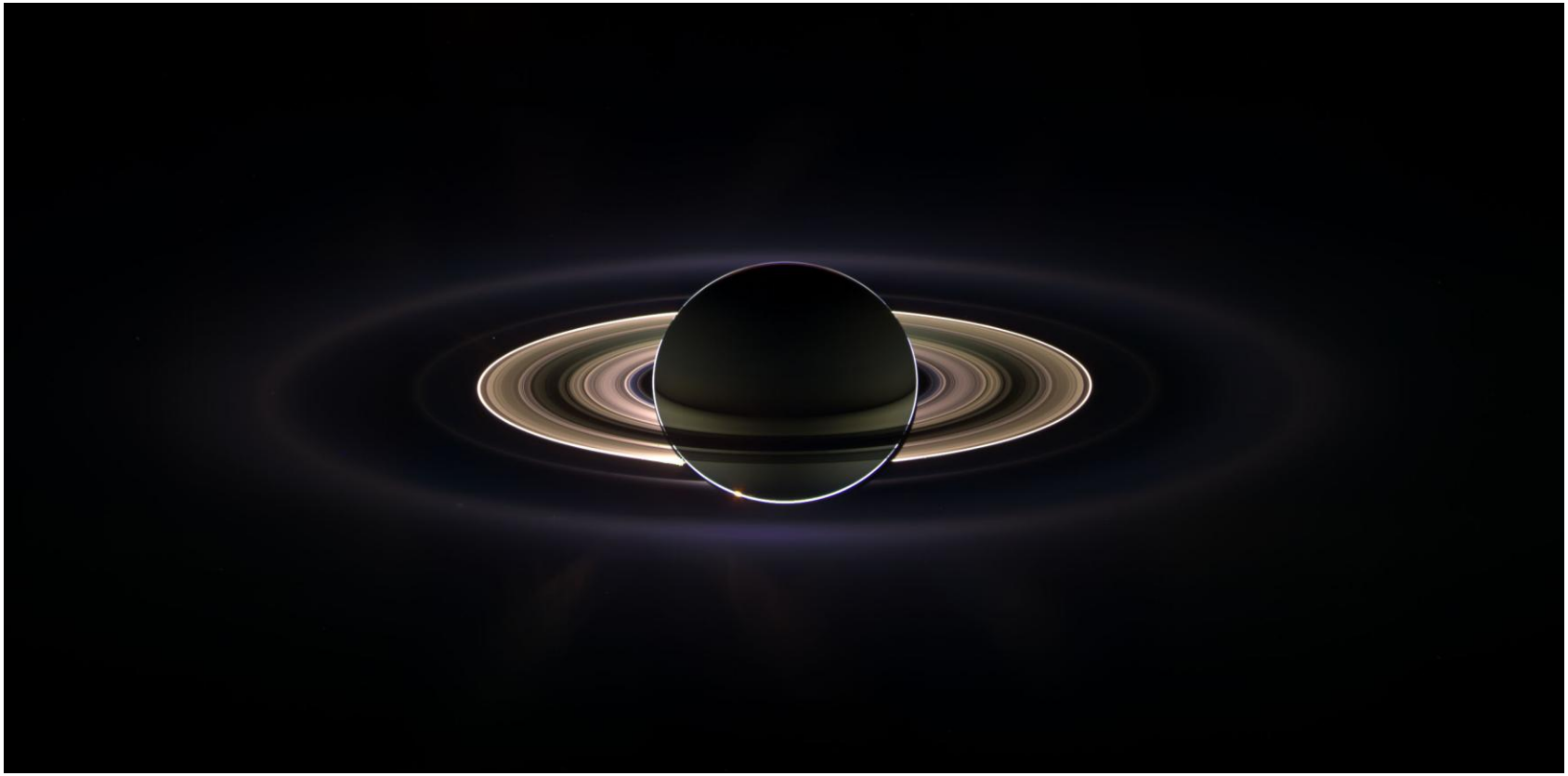
Error Correcting Codes

Magnetic Recording

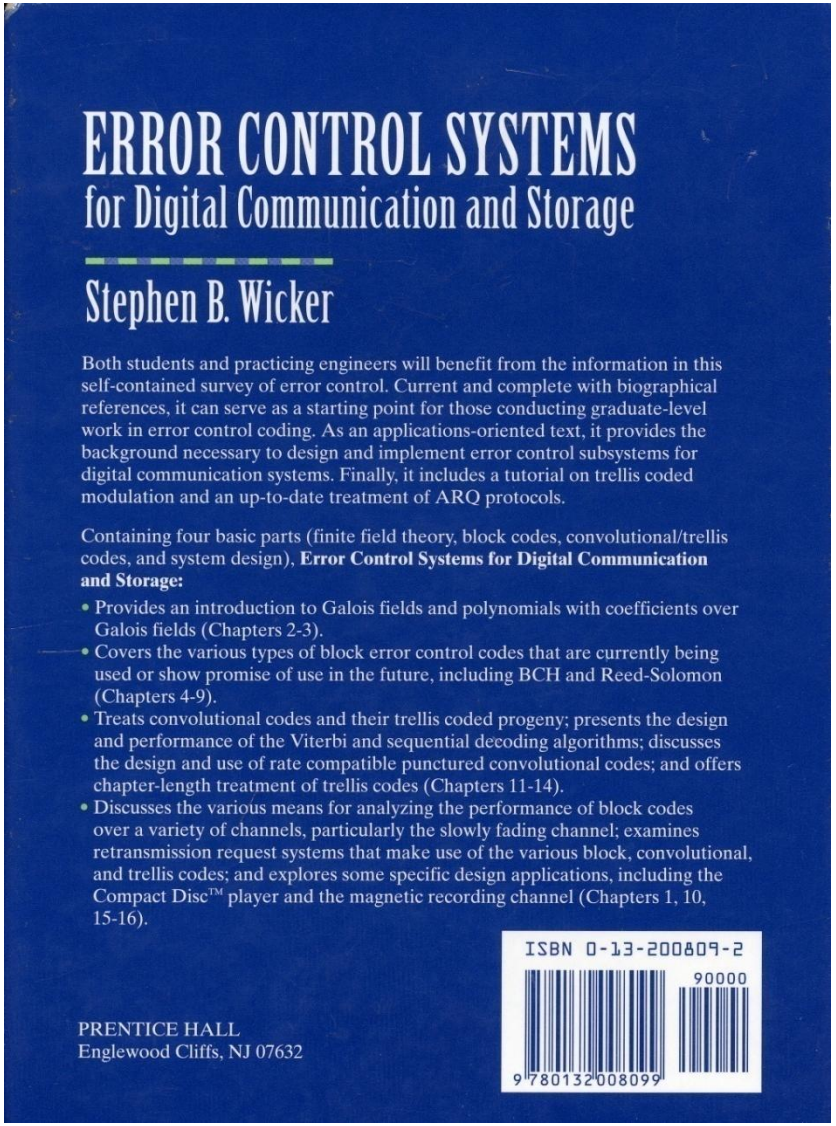
Some of the bits may change during reading/writing from the disc



Deep Space Communications



ISBN Codes



ERROR CONTROL SYSTEMS
for Digital Communication and Storage

Stephen B. Wicker

Both students and practicing engineers will benefit from the information in this self-contained survey of error control. Current and complete with biographical references, it can serve as a starting point for those conducting graduate-level work in error control coding. As an applications-oriented text, it provides the background necessary to design and implement error control subsystems for digital communication systems. Finally, it includes a tutorial on trellis coded modulation and an up-to-date treatment of ARQ protocols.

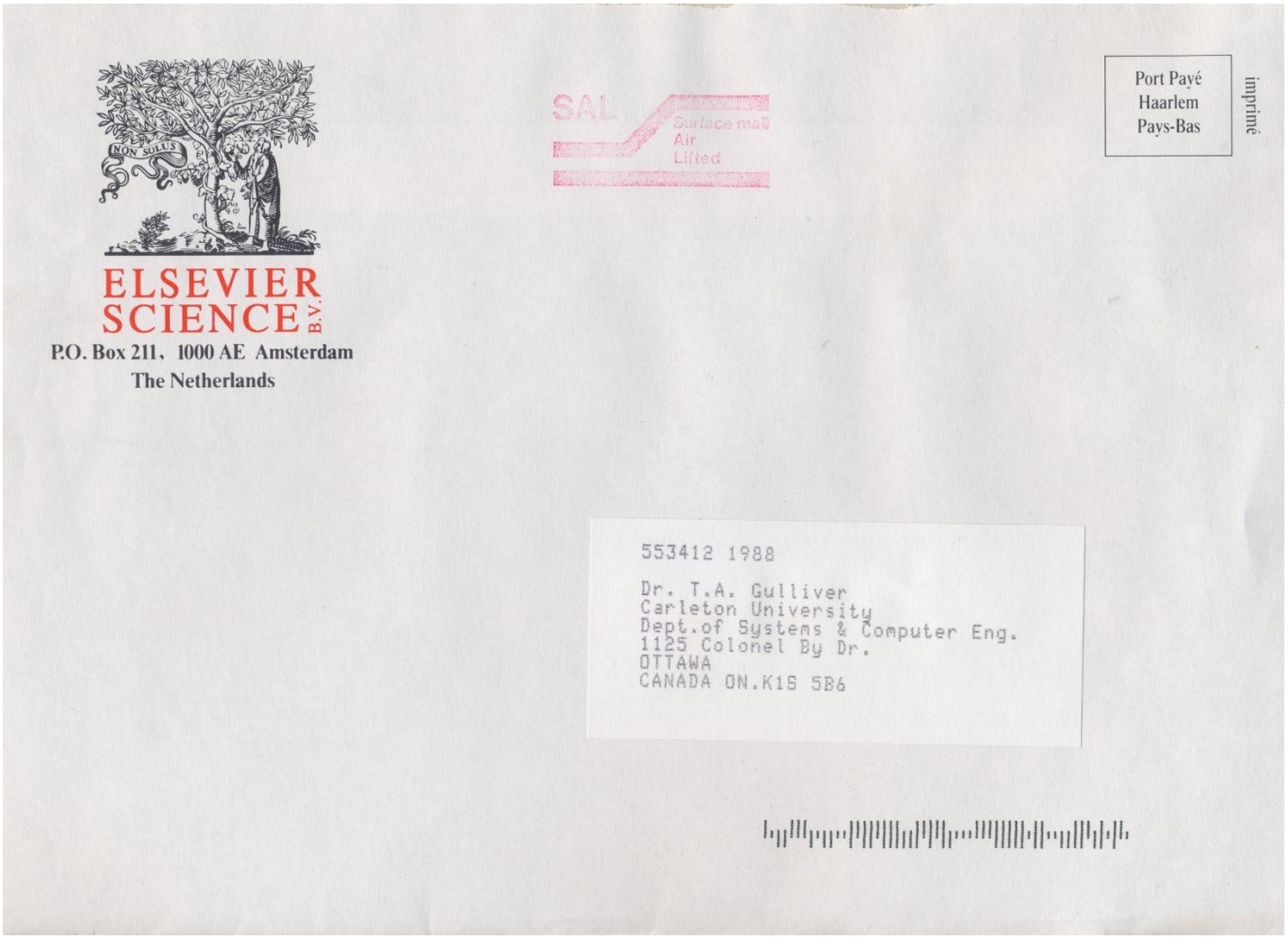
Containing four basic parts (finite field theory, block codes, convolutional/trellis codes, and system design), **Error Control Systems for Digital Communication and Storage:**

- Provides an introduction to Galois fields and polynomials with coefficients over Galois fields (Chapters 2-3).
- Covers the various types of block error control codes that are currently being used or show promise of use in the future, including BCH and Reed-Solomon (Chapters 4-9).
- Treats convolutional codes and their trellis coded progeny; presents the design and performance of the Viterbi and sequential decoding algorithms; discusses the design and use of rate compatible punctured convolutional codes; and offers chapter-length treatment of trellis codes (Chapters 11-14).
- Discusses the various means for analyzing the performance of block codes over a variety of channels, particularly the slowly fading channel; examines retransmission request systems that make use of the various block, convolutional, and trellis codes; and explores some specific design applications, including the Compact Disc™ player and the magnetic recording channel (Chapters 1, 10, 15-16).

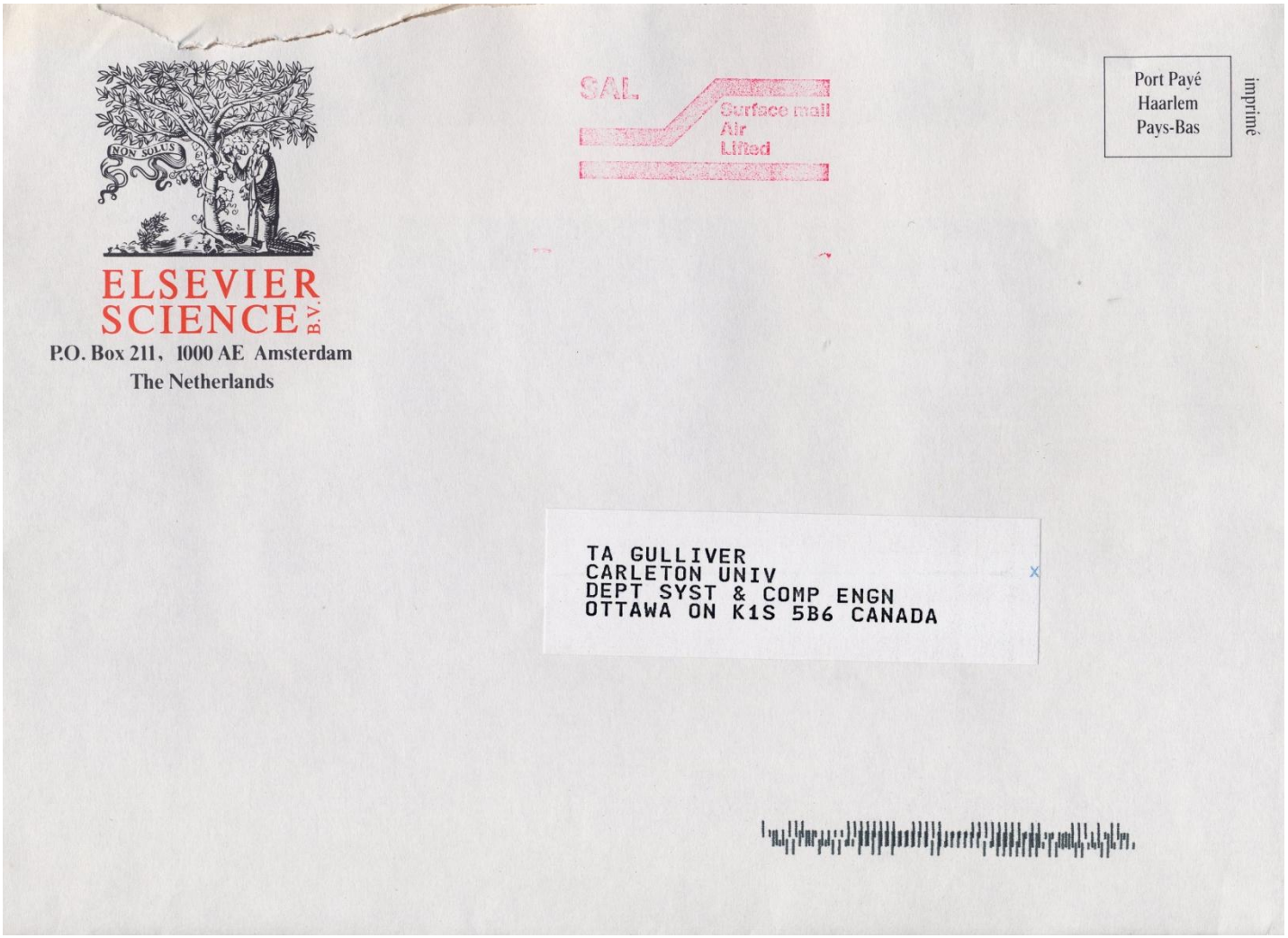
ISBN 0-13-200809-2
90000
9 780132 008099

PRENTICE HALL
Englewood Cliffs, NJ 07632

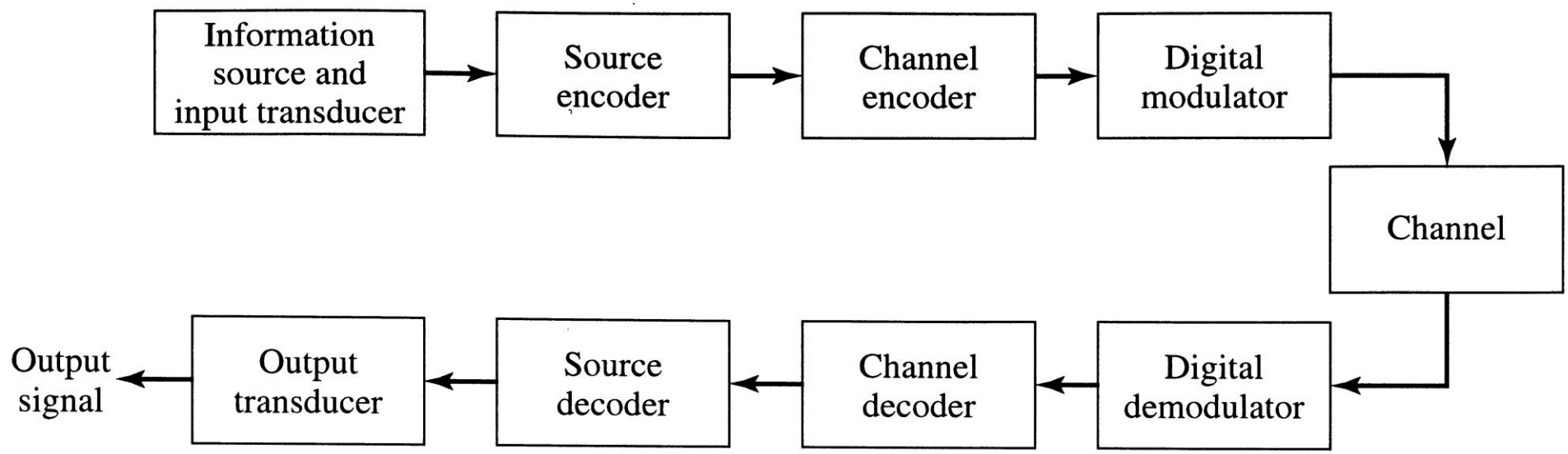
Bar Codes


NON SOLUS
ELSEVIER SCIENCE B.V.
P.O. Box 211, 1000 AE Amsterdam
The Netherlands
SAL
Surface mail
Air
Lifted
Port Payé
Haarlem
Pays-Bas
imprimé
553412 1988
Dr. T.A. Gulliver
Carleton University
Dept.of Systems & Computer Eng.
1125 Colonel By Dr.
OTTAWA
CANADA ON.K1S 5B6

Errors


NON SOLUS
ELSEVIER SCIENCE B.V.
P.O. Box 211, 1000 AE Amsterdam
The Netherlands
SAL
Surface mail
Air
Lifted
Port Payé
Haarlem
Pays-Bas
imprimé
TA GULLIVER
CARLETON UNIV
DEPT SYST & COMP ENGN
OTTAWA ON K1S 5B6 CANADA

Digital Communication System Model



Error Correcting Codes (ECCs)

- Redundancy is added to the data at the transmitter to permit error detection or correction at the receiver
- Encode k symbols into codewords of length n
- Code rate is $R = k/n$
- Used to correct errors due to noise, fading, interference and other channel impairments

Repetition Codes

- The most obvious means of adding redundancy is to repeat the data symbols

- Problem: $k = 1$, code rate $R = 1/n$ is very low for multiple error correction
- Example: Triple Repetition Code (TRC)

  0 is encoded as 000

  and

  1 is encoded as 111

TRC Decoding

- majority vote or nearest neighbor decoding

$$000, 001, 010, 100 \rightarrow 000$$

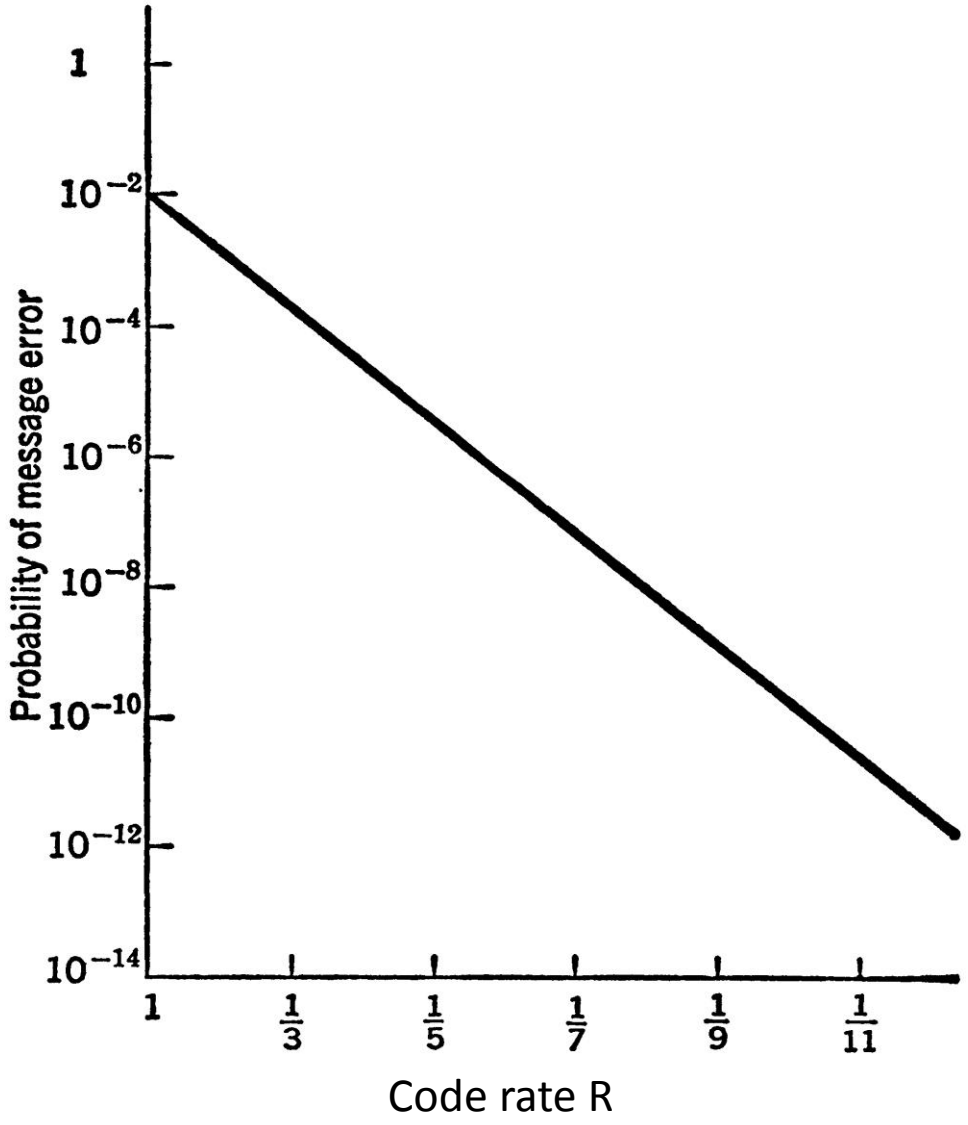
$$111, 110, 101, 011 \rightarrow 111$$

- the probability of a decoding error for a BSC is

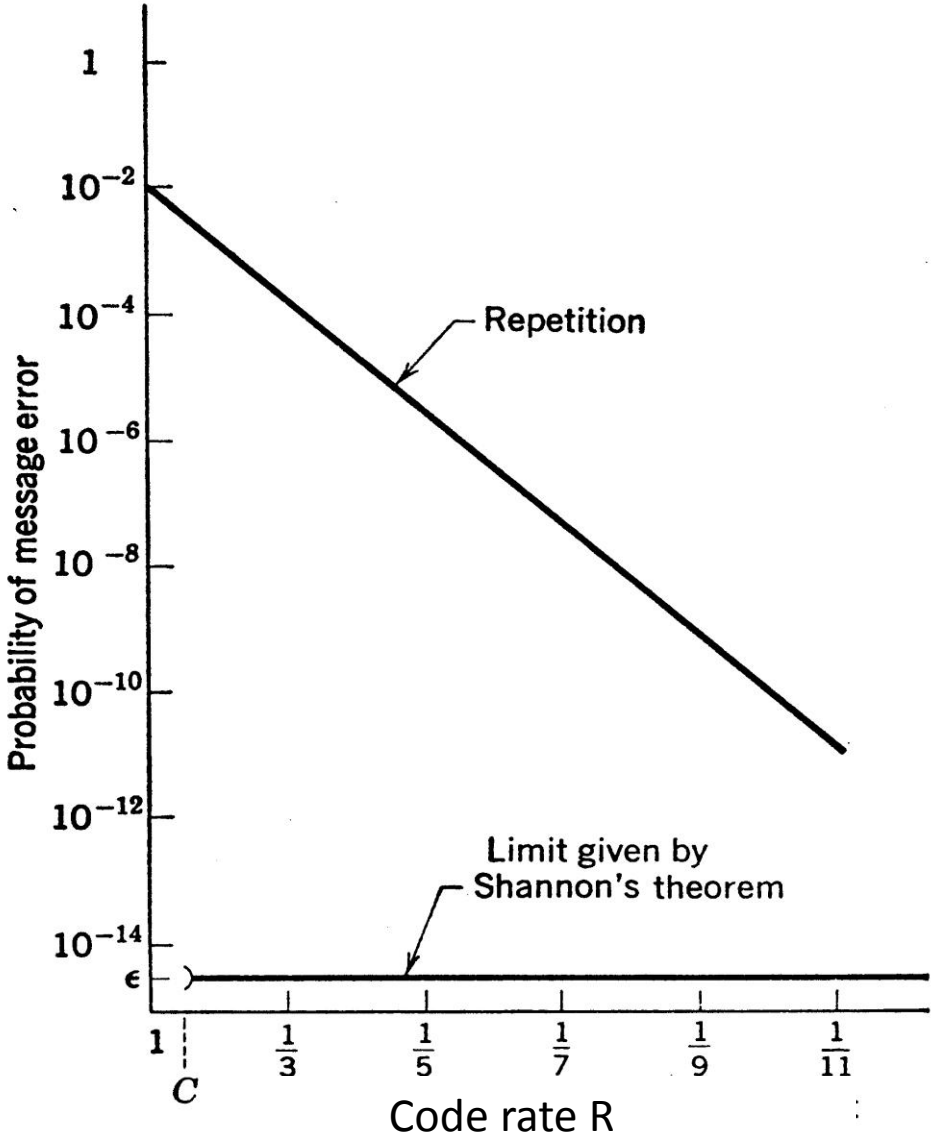
$$3p^2(1-p) + p^3 = 3p^2 - 2p^3 < p$$

- **Example:** If $p = 0.01$, then $P_e(C) = 2.98 \times 10^{-4}$
  - only one word in 3555 will be in error after decoding

Binary Repetition Codes



Binary Repetition Codes



ASCII Character Set

Least Significant Bits

Most Significant Bits	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 000	NUL (0) 00	SOH (1) 01	STX (2) 02	ETX (3) 03	EOT (4) 04	ENQ (5) 05	ACK (6) 06	BEL (7) 07	BS (8) 08	HT (9) 09	LF (10) 0A	VT (11) 0B	FF (12) 0C	CR (13) 0D	SO (14) 0E	SI (15) 0F
1 001	DLE (16) 10	DC1 (17) 11	DC2 (18) 12	DC3 (19) 13	DC4 (20) 14	NAK (21) 15	SYN (22) 16	ETB (23) 17	CAN (24) 18	EM (25) 19	SUB (26) 1A	ESC (27) 1B	FS (28) 1C	GS (29) 1D	RS (30) 1E	US (31) 1F
2 010	SP (32) 20	! (33) 21	" (34) 22	# (35) 23	$ (36) 24	% (37) 25	& (38) 26	' (39) 27	((40) 28	) (41) 29	* (42) 2A	+ (43) 2B	, (44) 2C	- (45) 2D	. (46) 2E	/ (47) 2F
3 011	0 (48) 30	1 (49) 31	2 (50) 32	3 (51) 33	4 (52) 34	5 (53) 35	6 (54) 36	7 (55) 37	8 (56) 38	9 (57) 39	: (58) 3A	; (59) 3B	< (60) 3C	= (61) 3D	> (62) 3E	? (63) 3F
4 100	@ (64) 40	A (65) 41	B (66) 42	C (67) 43	D (68) 44	E (69) 45	F (70) 46	G (71) 47	H (72) 48	I (73) 49	J (74) 4A	K (75) 4B	L (76) 4C	M (77) 4D	N (78) 4E	O (79) 4F
5 101	P (80) 50	Q (81) 51	R (82) 52	S (83) 53	T (84) 54	U (85) 55	V (86) 56	W (87) 57	X (88) 58	Y (89) 59	Z (90) 5A	[(91) 5B	\ (92) 5C	] (93) 5D	^ (94) 5E	_ (95) 5F
6 110	` (96) 60	a (97) 61	b (98) 62	c (99) 63	d (100) 64	e (101) 65	f (102) 66	g (103) 67	h (104) 68	i (105) 69	j (106) 6A	k (107) 6B	l (108) 6C	m (109) 6D	n (110) 6E	o (111) 6F
7 111	p (112) 70	q (113) 71	r (114) 72	s (115) 73	t (116) 74	u (117) 75	v (118) 76	w (119) 77	x (120) 78	y (121) 79	z (122) 7A	{ (123) 7B	| (124) 7C	} (125) 7D	~ (126) 7E	DEL (127) 7F

SPC Codes – Example 1

- ASCII symbols

  E = 1000101 c = 10001011

  G = 1000111 c = 10001110

- Received word

  r = 10001010

Binary Single Parity Check Codes

- Another simple class of codes
  - In this case, n = k+1
    - the codeword is the dataword with one additional bit
  - For even parity the additional bit is

$$q = \sum_{i=1}^{k} d_i \pmod 2$$

  - The additional bit ensures that there are an even number of 1s in the codeword

SPC Codes – Example 2

Coding table for the (4,3) SPC code

Dataword			Codeword			
0	0	0	0	0	0	0
0	0	1	0	0	1	1
0	1	0	0	1	0	1
0	1	1	0	1	1	0
1	0	0	1	0	0	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	1	1	1	1

SPC Codes

- To decode
  - Calculate the sum of the received bits (mod 2)
  - If the sum is 0 then the dataword is the first *k* bits of the received word
  - Otherwise declare an error
- Code can detect single errors
- But cannot correct an error since the error could be in any bit
- Low overhead but not very powerful
- Decoder can be implemented efficiently

Single Parity Check Code Example

- Consider the 2^{11} binary words of length 11 as datawords
- Let the probability of bit error be $p = 10^{-8}$
- Bits are transmitted at a rate of 10^7 bits per second
- The probability that a word is received incorrectly is approximately

$$11p(1-p)^{10} \approx \frac{11}{10^8}$$

- Therefore $\frac{11}{10^8} \cdot \frac{10^7}{11} = 0.1$ words per second are received incorrectly.
- One wrong word is received every 10 seconds, 360 erroneous words every hour and 8640 words every day without being detected!

SPC Code Example (Cont.)

- Now add one parity bit
- Any single error can be detected
- The probability of at least 2 errors is

$$1-(1-p)^{12}-12(1-p)^{11}p \approx \binom{12}{2}(1-p)^{10}p^2 \approx \frac{66}{10^{16}}$$

- Therefore approximately $\frac{66}{10^{16}} \cdot \frac{10^7}{12} \approx 5.5 \cdot 10^{-9}$ words per second are received with an undetectable error
- An undetected error occurs only every 2000 days ($2000 \approx 10^9/(5.5 \times 86400)$)

Modular Arithmetic

In mod (modulo) 2 arithmetic, 2 is the base (modulus) and there are no numbers other than 0 and 1. Any higher number mod 2 is obtained by dividing it by 2 and taking the remainder.

For example, $3 \equiv 1 \bmod 2$ and $4 \equiv 0 \bmod 2$.

mod 2 addition

+	0	1
0	0	1
1	1	0

same as logical XOR

mod 2 multiplication

•	0	1
0	0	0
1	0	1

same as logical AND

Vector Spaces

- Set of n-tuples over an alphabet *A*
  - n-dimensional vector space
- Example: binary n-tuples of length 5 – V_5
  - 5-dimensional vector space

$$\left.\begin{array}{l} 00000 \\ 00001 \\ 00010 \\ 00011 \\ 00100 \\ \vdots \\ 11111 \end{array}\right\} 32 \quad \text{5-tuples}$$

Vector Space Operations

vector addition

$$\begin{array}{r} 11001 \\ +10011 \\ \hline 01010 \end{array}$$

scalar multiplication

$$a \bullet \bar{v}$$

$$0 \bullet (11001) = 00000$$

$$1 \bullet (11001) = 11001$$

$$a \in A$$

The space is closed under vector addition and scalar multiplication

Inner Product

$$\bar{u} \bullet \bar{v} = \sum_{i=0}^{n-1} u_i \bullet v_i$$

$$\begin{aligned} (11001) \bullet (10011) &= 1 \bullet 1 + 1 \bullet 0 + 0 \bullet 0 + 0 \bullet 1 + 1 \bullet 1 \\ &= 2 \\ &= 0 \bmod 2 \end{aligned}$$

(11001) and (10011) are orthogonal

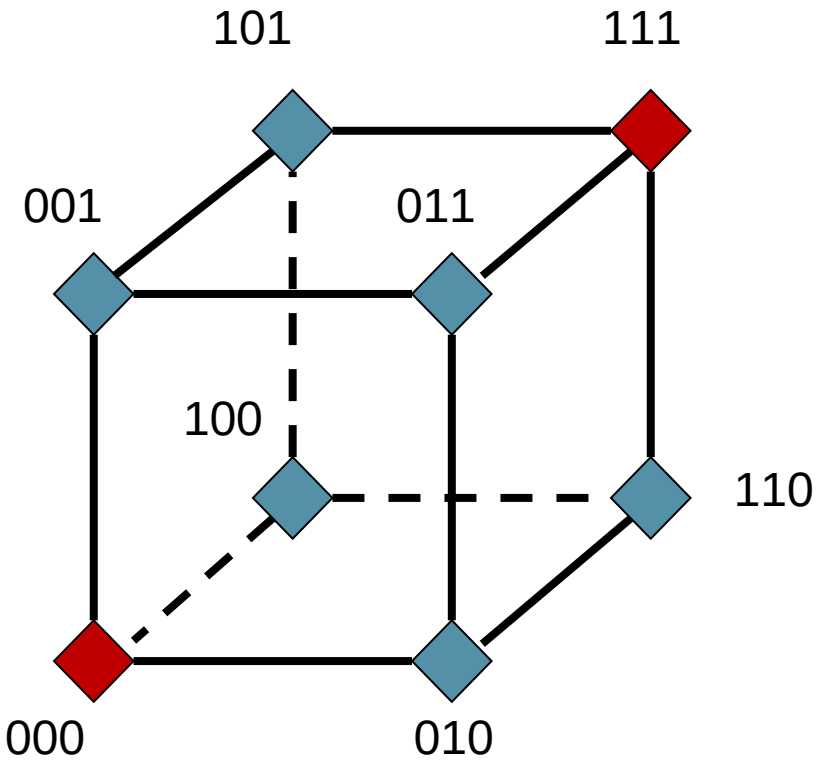
Vector Subspaces

- A smaller vector space which is closed under vector addition and scalar multiplication
- Example: subspace of V_5

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{matrix}$$

$$\begin{array}{r} 00111 \\ +11011 \\ \hline 11100 \end{array}$$

Triple Repetition Code



V_3

Basis

- A minimal number of linearly independent vectors from the vector space that span the space

$$\begin{bmatrix} 00111 \\ 11100 \end{bmatrix}$$

$$\begin{aligned} 0 \bullet (00111) + 0 \bullet (11100) &= 00000 \\ 0 \bullet (00111) + 1 \bullet (11100) &= 11100 \\ 1 \bullet (00111) + 0 \bullet (11100) &= 00111 \\ 1 \bullet (00111) + 1 \bullet (11100) &= 11011 \end{aligned}$$

Dual Spaces

- Set of vectors orthogonal to a vector space

S	$S^{\perp}$
0000	0000
0101	1010
0001	1000
0100	0010

Vector Space Dimensions

- If a basis has *k* elements then the vector space is said to have dimension *k*

$$S \qquad\qquad S^{\perp}$$

$$\begin{bmatrix} 0001 \\ 0100 \end{bmatrix} \qquad \begin{bmatrix} 1000 \\ 0010 \end{bmatrix}$$

$$\dim(S) + \dim(S^{\perp}) = \dim(V)$$

Example

- For the space generated by the basis

$$\begin{bmatrix} 00111 \\ 11100 \end{bmatrix}$$

what is the dual space?

Self-Dual Spaces

$$S = S^{\perp}$$

- Example

S	$S^{\perp}$
0000	0000
1010	1010
0101	0101
1111	1111

Binary Codes in Vector Spaces

Sets of codewords can be considered as vectors in the vector space V_n of binary vectors of length n.

Definition A subset $C \subseteq V_n$ is a binary linear block code if $u + v \in C$ for all $u, v \in C$.

C is a k dimensional subspace of V_n.

Binary Linear Block Codes

- Linear codes are the most important class of error correcting codes
  - simple description
  - nice structure and properties
  - easy encoding
  - conceptually easy decoding
- Linear code: mod 2 sum of any two codewords is a codeword
- Block code: codewords have a finite length *n*
- The number of codewords is

$$|C| = M = 2^k$$

- Each codeword represents *k* data bits
- The code rate is

$$R = \frac{\log_2 M}{n} = \frac{k}{n}$$

Basis (Generator) Matrices

- Triple repetition code $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

- Single parity check code

$$\begin{bmatrix} & & 1 \\ & & 1 \\ & & 1 \\ & I_7 & 1 \\ & & 1 \\ & & 1 \\ & & 1 \end{bmatrix}$$

E = 1000101 c = 10001011

G = 1000111 c = 10001110

Length 5 Code

- *k* x *n* Generator matrix

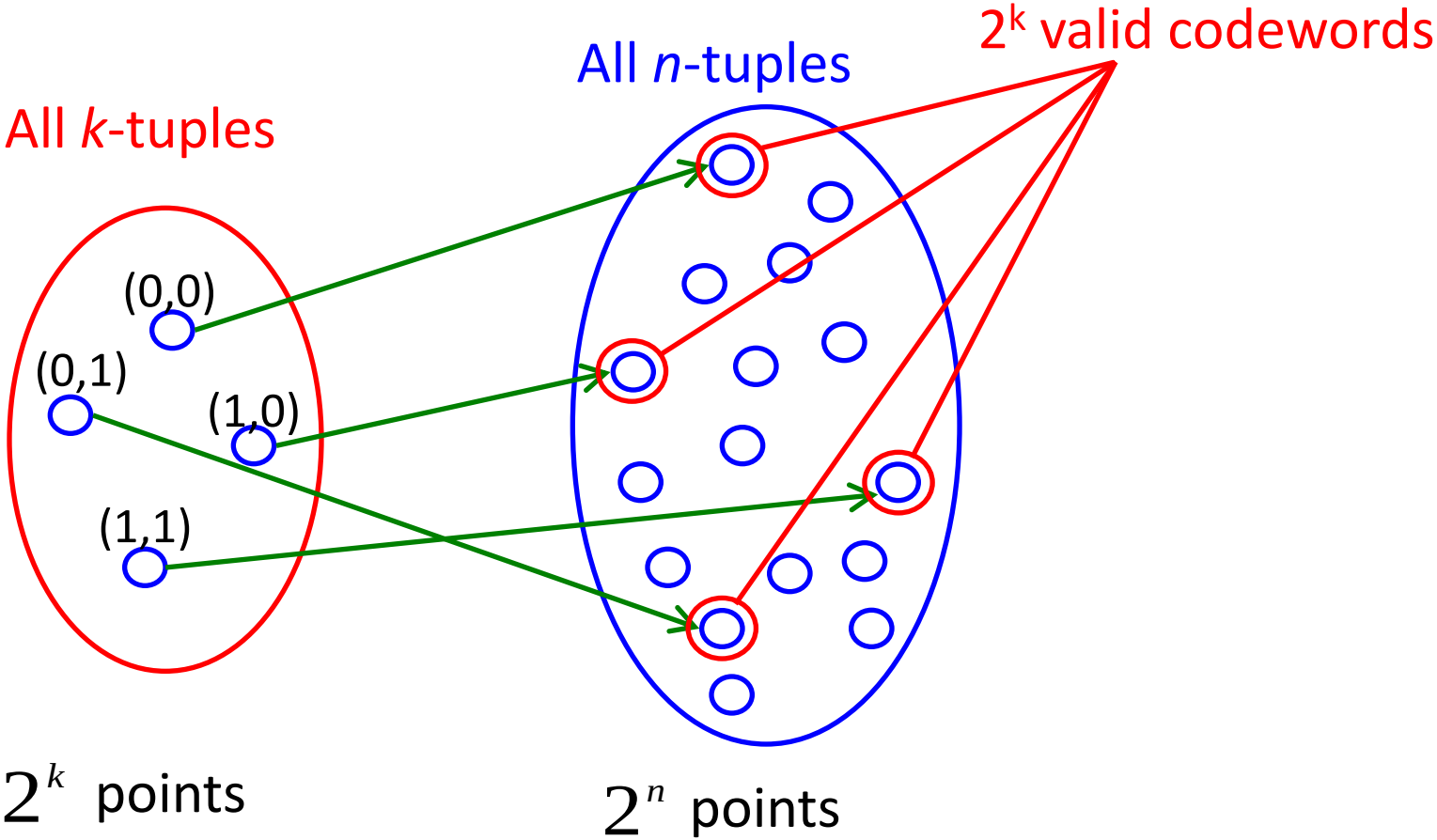
$$G = \begin{matrix} 5 \\ \left[\begin{array}{c} 00111 \\ 11100 \end{array}\right] \end{matrix} 2$$

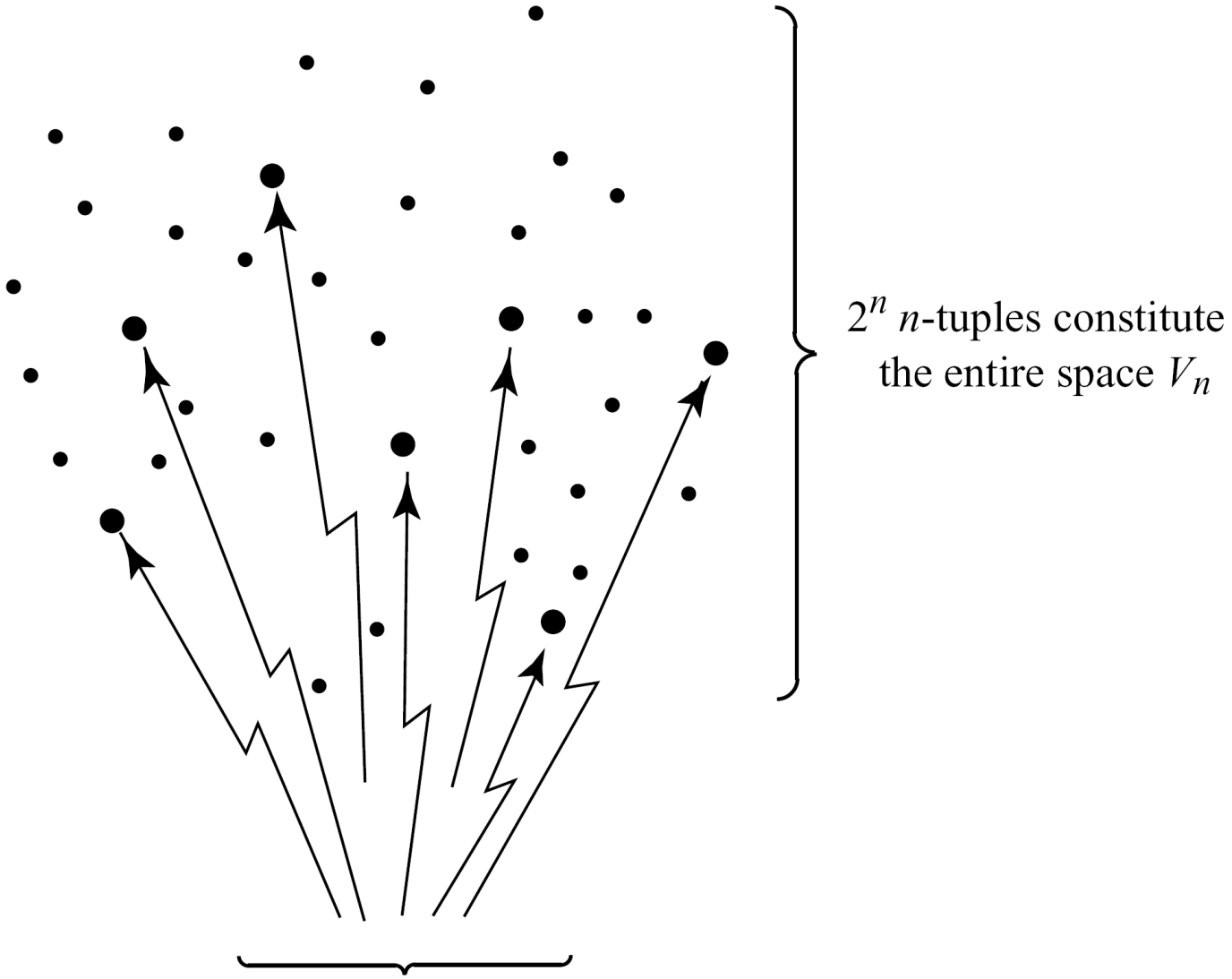
-

m	c
00	00000
01	11100
10	00111
11	11011

Linear Codes as Vector Spaces (Cont.)

$$(x_1, x_2, \ldots, x_k) \mapsto (c_1, c_2, \ldots, c_n)$$




2^n n-tuples constitute
the entire space V_n
2^k n-tuples constitute
the subspace of codewords

Which of the following binary codes are linear?

$C_1 = \{00, 01, 10, 11\}$

$C_2 = \{000, 011, 101, 110\}$

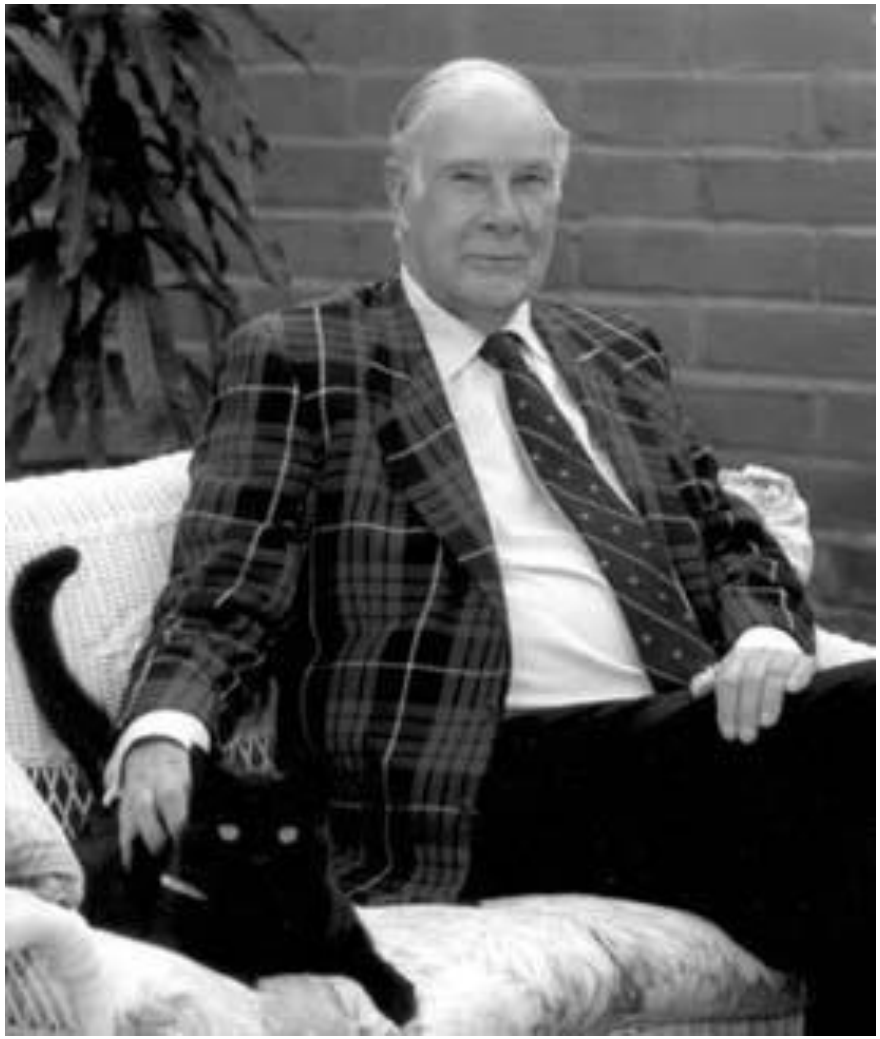
$C_3 = \{00000, 11000, 00111, 11111\}$

$C_4 = \{101, 111, 011\}$

$C_5 = \{000, 001, 010, 011\}$

$C_6 = \{0000, 1001, 0110, 1110\}$

Richard W. Hamming (1915-1998)



Hamming at Bell Labs

- The development of error correcting codes began in 1947 at Bell Laboratories
- Hamming had access to a mechanical relay computer on some weekends
- The computer employed an error detecting code, but with no operator on duty during weekends, the computer simply stopped or went on to the next problem when an error occurred

“Two weekends in a row I came in and found that all my stuff had been dumped and nothing was done.” And so I said, “Damn it, if the machine can detect an error, why can't it locate the position of the error and correct it?”

Hamming Weight and Distance

- The concept of closeness of two codewords is formalized through the Hamming distance.
- Let x and y be any two codewords in C

  $x = 00111 \quad y = 11100$
- The Hamming weight of a codeword is defined as the number of nonzero elements

  $w(x) = w(00111) = 3 \quad w(y) = w(11100) = 3$
- The Hamming distance between two codewords is defined as the number of places in which they differ

  $d(x,y) = d(00111,11100) = 4$

Hamming distance properties

(1) $d(x,y) = 0 \Leftrightarrow x = y$
(2) $d(x,y) = d(y,x)$
(3) $d(x,z) \leq d(x,y) + d(y,z)$ triangle inequality

Hamming Distances for Linear Codes

- For a binary linear code, the mod 2 sum of any two codewords is another codeword

  $x + y = z$ 00111 + 11100 = 11011

- Thus

  $d(x,y) = w(x+y) = w(z) = w(11011) = 4$

- Since we are concerned with the error correcting capability of a code *C*:

  What is the most important criteria for a linear (*n*,*k*) code?

- The minimum Hamming distance d(*C*)

(5,2,3) Code

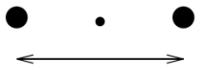
- *k* x *n* Generator matrix

$$G = \begin{matrix} \\ \left[\begin{array}{c} 00111 \\ 11100 \end{array}\right] \end{matrix} \begin{matrix} 5 \\ 2 \end{matrix}$$

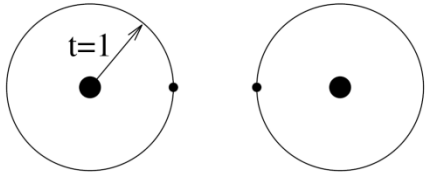
-

m	c	w
00	00000	0
01	11100	3
10	00111	3
11	11011	4

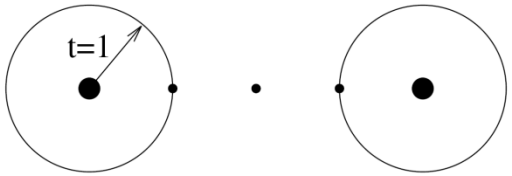
- d(*C*) = 3 (also called d_{min})



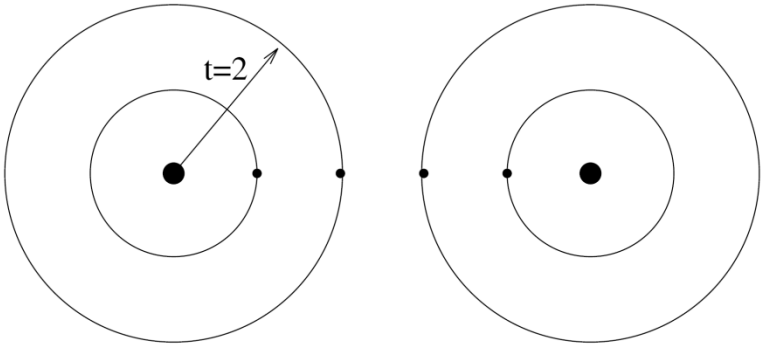
d=2



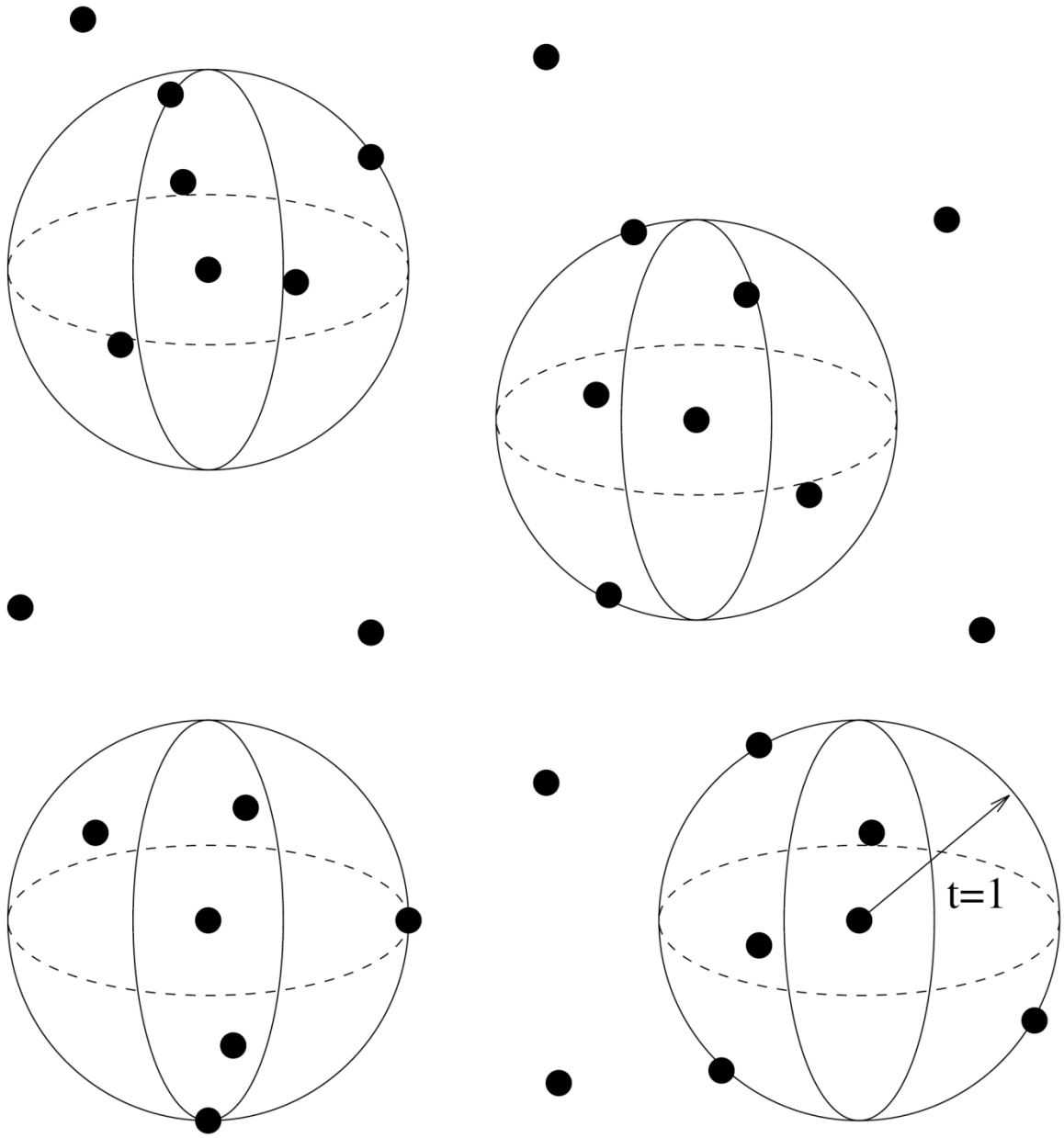
d=3



d=4



d=5


V_5
t=1

(5,2,3) Code

- *k* x *n* Generator matrix

$$G = \begin{matrix} \\ \left[\begin{array}{c} 00111 \\ 11100 \end{array}\right] \end{matrix} \begin{matrix} 5 \\ 2 \end{matrix}$$

-

m	c	w
00	00000	0
01	11100	3
10	00111	3
11	11011	4

- d(*C*) = 3, $t = \left\lfloor \frac{3-1}{2} \right\rfloor = 1$

Minimum Hamming Distance

- An important parameter of a code C is its minimum distance

  $$d(C) = \min \{d(x,y) \mid x,y \in C, x \neq y\}$$

- (1) A code C can detect up to v errors if $d(C) \geq v + 1$
- (2) A code C can correct up to t errors if $d(C) \geq 2t + 1$

Proof

- (1) Follows from the codewords being at least distance $v + 1$ apart, so that v errors cannot transform one codeword into another.
- (2) Suppose $d(C) \geq 2t + 1$. Let a codeword x be transmitted and a word y received such that $d(x,y) \leq t$. If $x' \neq x$ is a codeword, then $d(x,y) \geq t + 1$ because otherwise $d(x',y) < t + 1$ and therefore $d(x,x') \leq d(x,y) + d(y,x') < 2t + 1$ which contradicts the assumption that $d(C) \geq 2t + 1$.

Important Linear Block Codes

There are many classes of practical linear block codes:

- Hamming codes
- Cyclic codes (CRC codes)
- Reed-Solomon codes
- BCH codes
- LDPC codes
- Turbo codes
- …

Notation and Examples

An (n,k,d) code C is a linear code such that

- n - is the length of the codewords
- k - is the number of data symbols in a codeword
- d - is the minimum distance of C

Examples:

C_1 = {00, 01, 10, 11} is a (2,2,1) code.

C_2 = {000, 011, 101, 110} is a (3,2,2) code.

C_3 = {00000, 11100, 00111, 11011} is a (5,2,3) code.

A good code has small n, large k, and large d.

Bases for Linear Codes

If C is a linear (n,k) code, then it has a basis consisting of k linearly independent codewords

Example:

C_7 = {0000000, 1111111, 1000101, 1100010,
0110001, 1011000, 0101100, 0010110,
0001011, 0111010, 0011101, 1001110,
0100111, 1010011, 1101001, 1110100}

has basis

{1111111, 1000101, 1100010, 0110001}

How many different bases for a binary linear code?

$$\frac{1}{k!}\prod_{i=0}^{k-1}\left(2^k - 2^i\right)$$

Bases for the (5,2,3) Code

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\mathbf{G}_3 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$