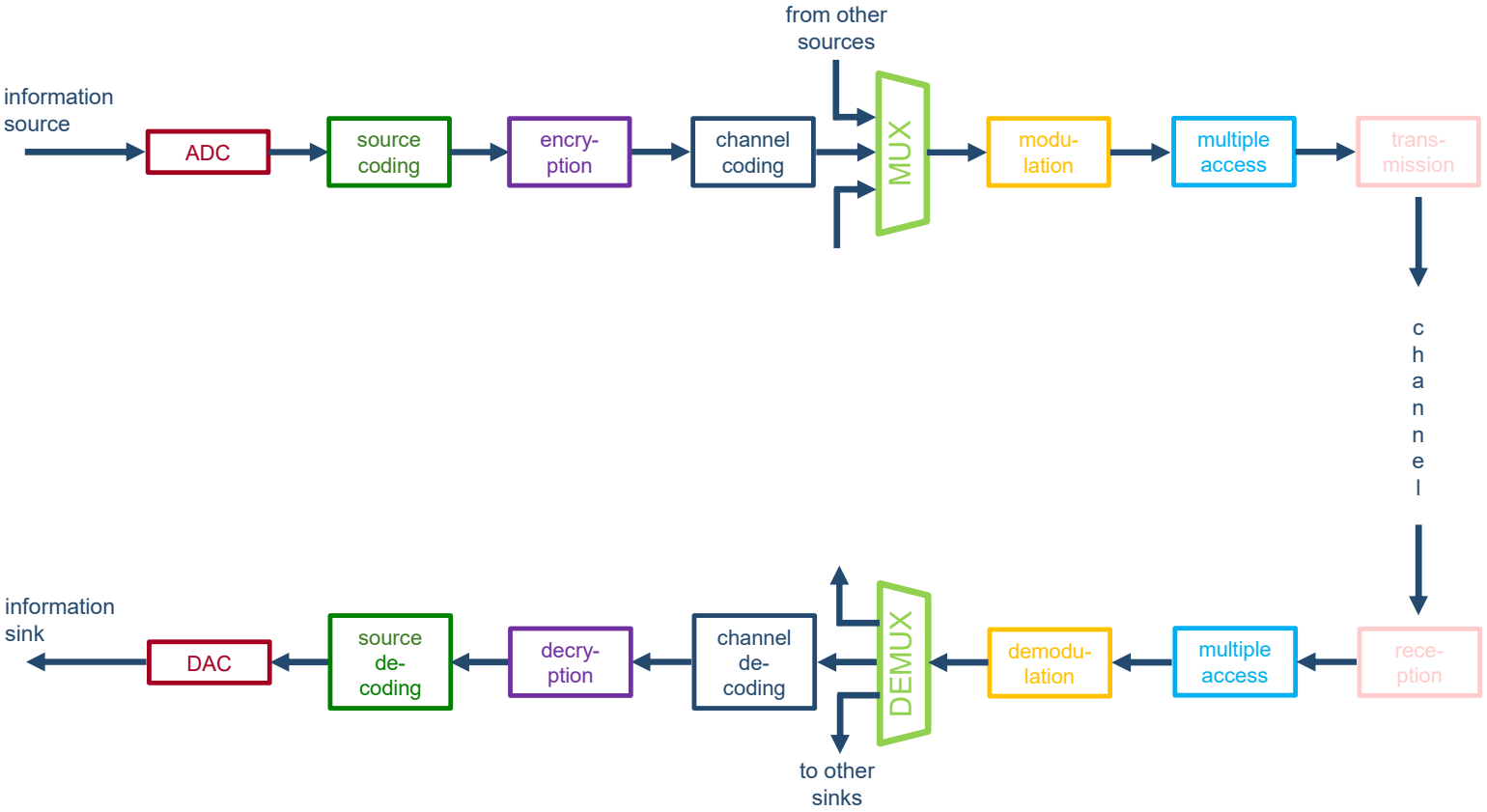


938II – Electronics and communication technologies (2026/27)

Communication technologies: Digital communications

Giacomo Bacci
giacomo.bacci@unipi.it

Elements of a digital communication system



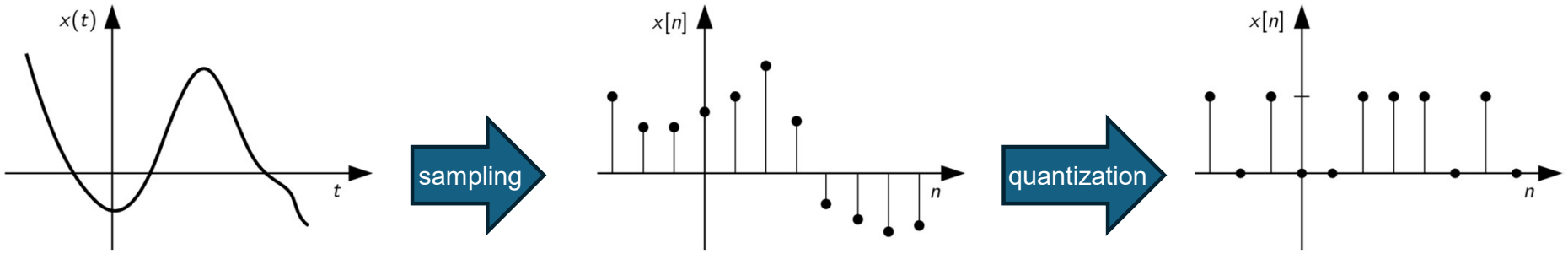


Analog-to-digital conversion (ADC)

Analog-to-digital conversion (ADC)

The ADC operation consists into two steps:

- Sampling, which makes the signal a sequence of values: time becomes discrete, values of the sequence are still in the real domain
- Quantization, which represents the values of the sequence with a finite number of bits: values now belong to a finite set



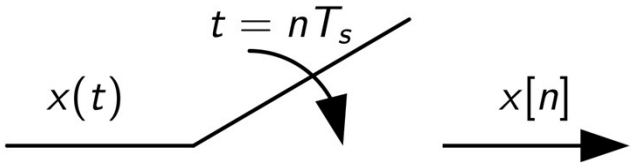
analog signal sequence digital signal



Sampling and interpolation

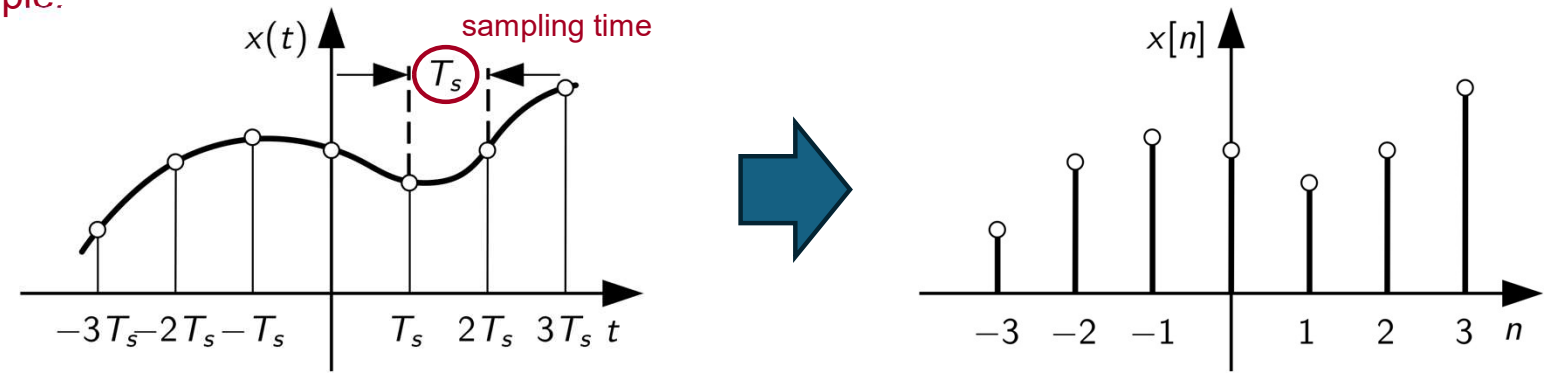
Sampling an analog signal

Sampling an analog signal $x(t)$ means collecting, one after the other, the sequence $x[n]$ of values taken by that signal at time instants nT_s:

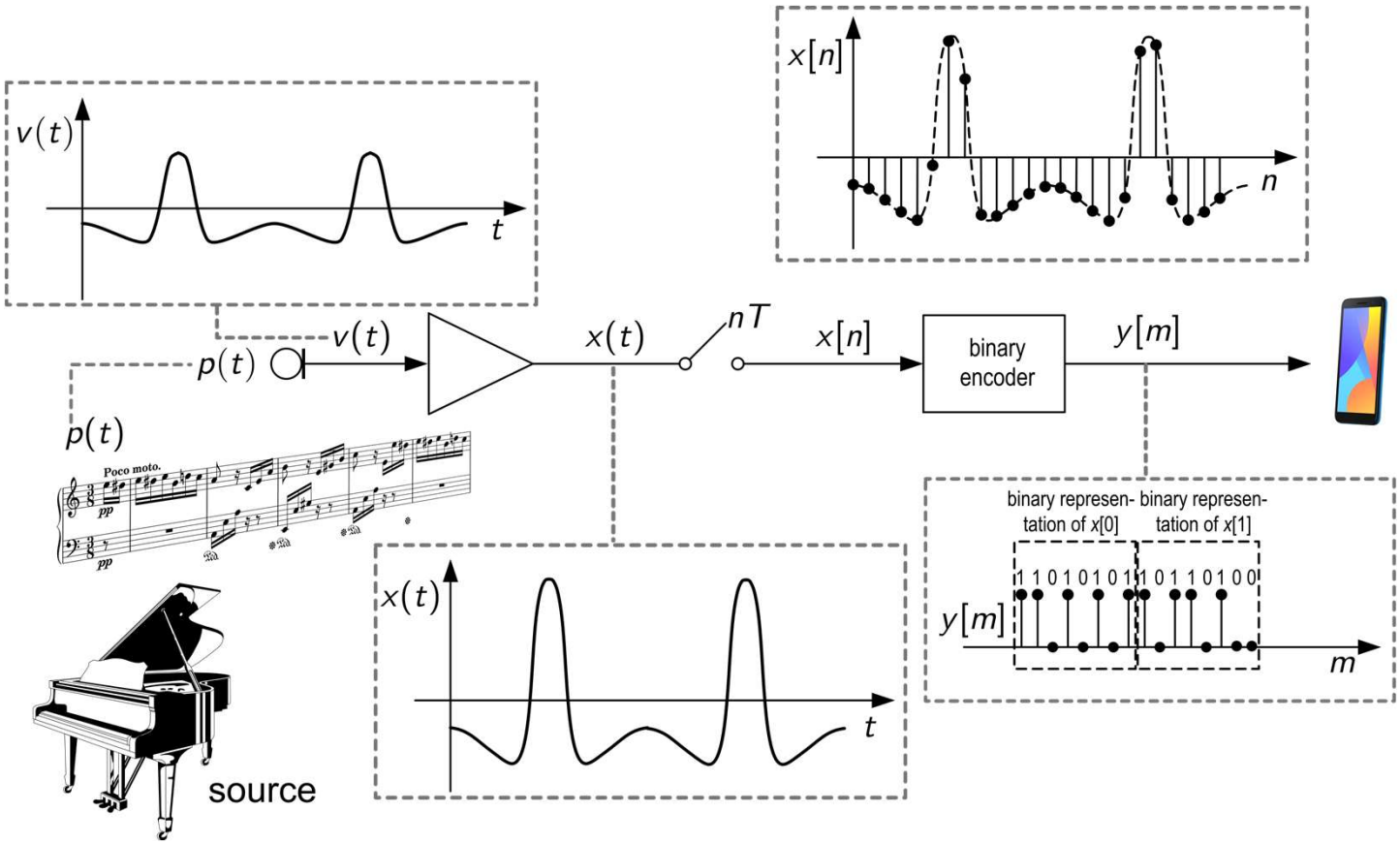


sampling frequency $f_s = 1/T_s$

Example:

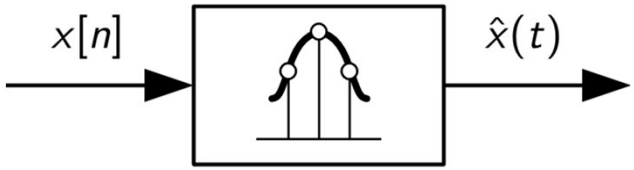


Example: digital recording of an audio signal

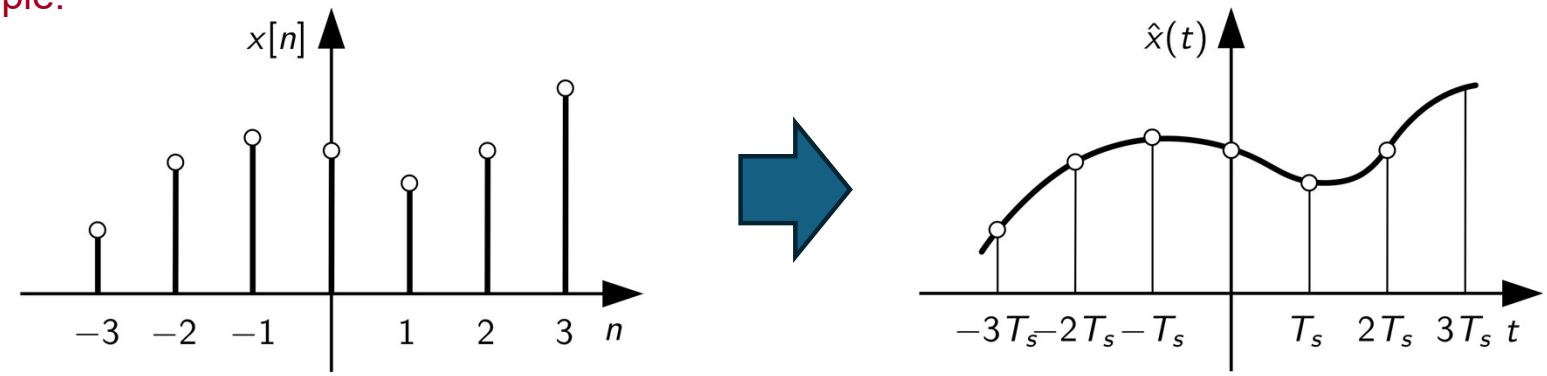


Back to the analog domain: interpolation

To reconstruct, with sufficiently high fidelity, the original signal starting from the sequence of samples, we can perform the operation of interpolation:

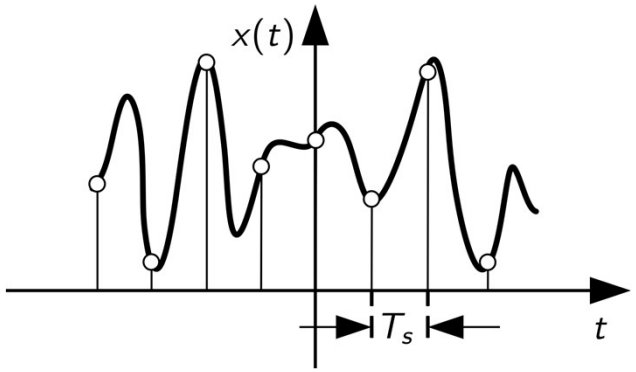


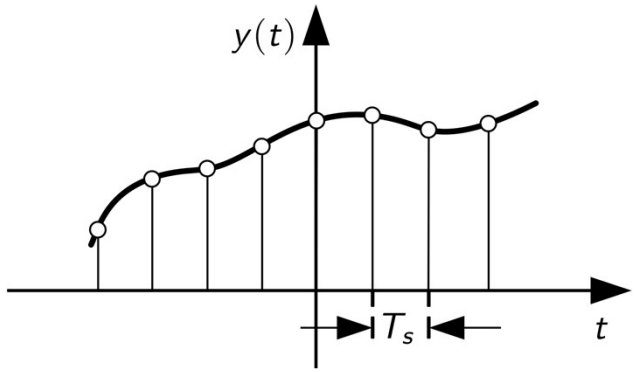
Example:



Sampling frequency (1/2)

The natural question is: how can we properly set the sampling frequency f_s?



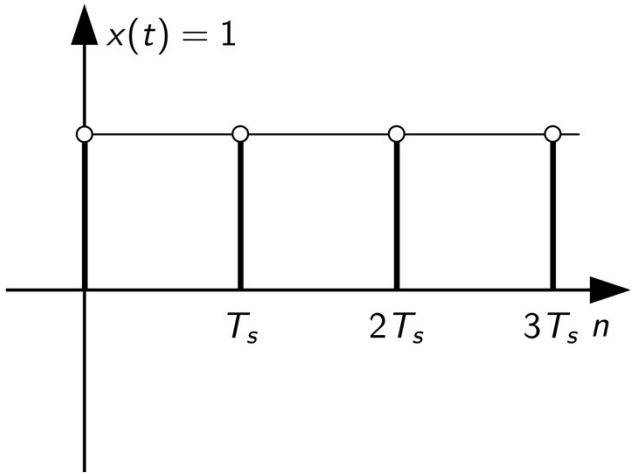


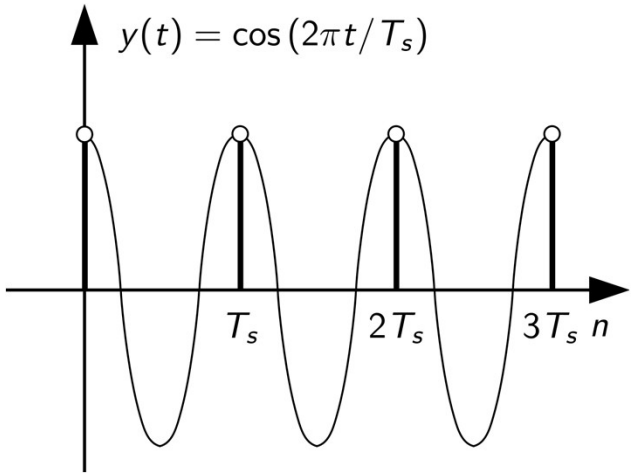
The sampling period is adequate for $y(t)$, but it is clearly too large for $x(t)$!

Note: increasing f_s (or equivalently, reducing T_s) means increasing the number of samples per unit of time, with drawbacks in terms of hardware and storage requirements

Sampling frequency (2/2)

How critical is the choice of the sampling frequency?



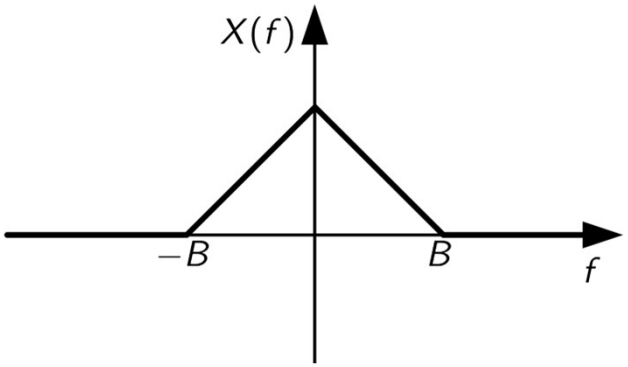


To select the proper sampling frequency, it is worth moving to the frequency domain

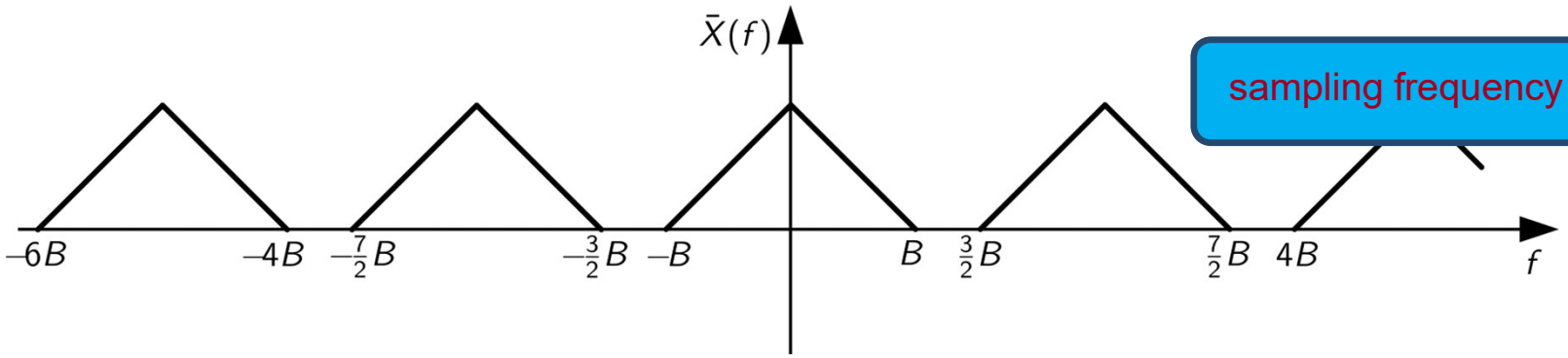
Fourier transform of a sequence

Sequence in the frequency domain (1/2)

Fourier transform of the analog signal $x(t)$:



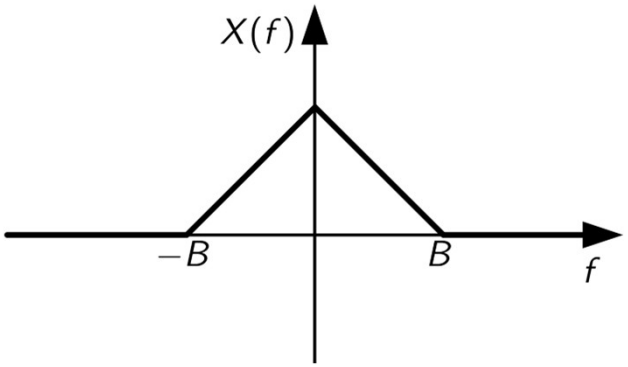
What happens to the Fourier transform of the sampled signal $x[n]$, when using $f_s = 5B/2$?



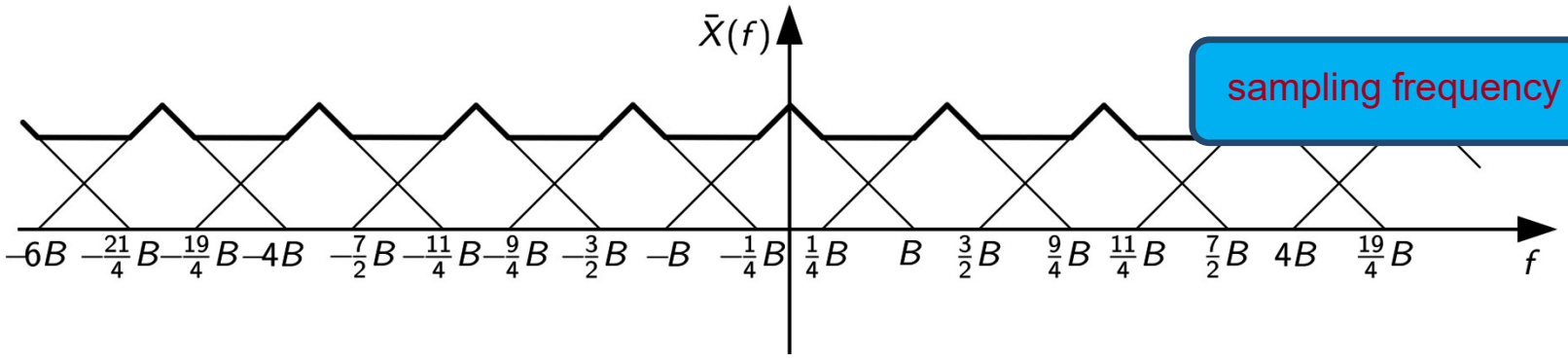
sampling frequency $f_s = \frac{5}{2}B$

Sequence in the frequency domain (2/2)

Fourier transform of the analog signal $x(t)$:



How the Fourier transform of $x[n]$ changes when, instead, we use $f_s = 5B/4$?



sampling frequency $f_s = \frac{5}{4}B$

Aliasing (1/2)

Sampling in the time domain implies replicating the spectrum of the analog signal in the frequency domain:

$$\overline{X}(f) = \frac{1}{T_s} \sum_{k=-\infty}^{+\infty} X\left(f - \frac{k}{T_s}\right)$$

This also implies that the Fourier transform of a sequence is periodic:

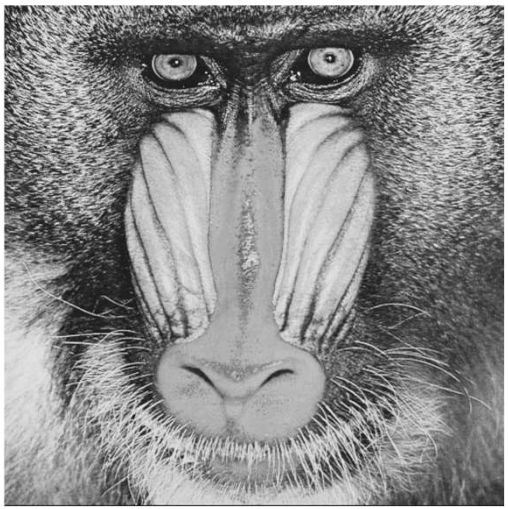
$$\overline{X}(f) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j2\pi n f T_s} = \overline{X}\left(f - \frac{\ell}{T_s}\right), \quad \ell \in \mathbb{Z}$$



Aliasing (2/2)

The superposition of replicas of the Fourier transform of the analog signal is called aliasing

When the sampling frequency is too small compared to the bandwidth of the analog signal, this generates ambiguity in the sampled version of the signal (the aliasing error):



original image



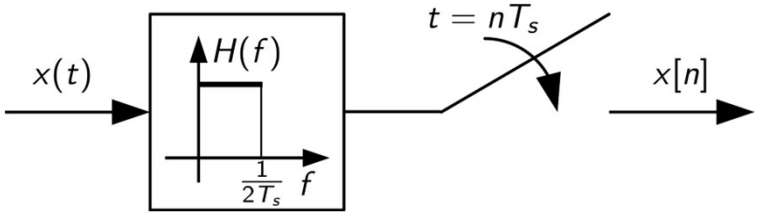
image sampled with insufficient spatial resolution

Nyquist frequency

To avoid aliasing errors when sampling a strictly bandlimited signal with bandwidth *B*, we need to satisfy the Nyquist condition:

$$f_s = \frac{1}{T_s} \geq 2B$$

In case of band-unlimited signals, we can place an anti-aliasing (low-pass) filter to limit the bandwidth of the analog signal, so as to avoid the aliasing error



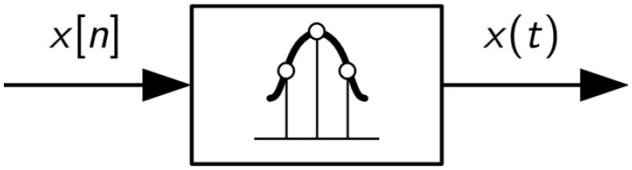


Interpolation

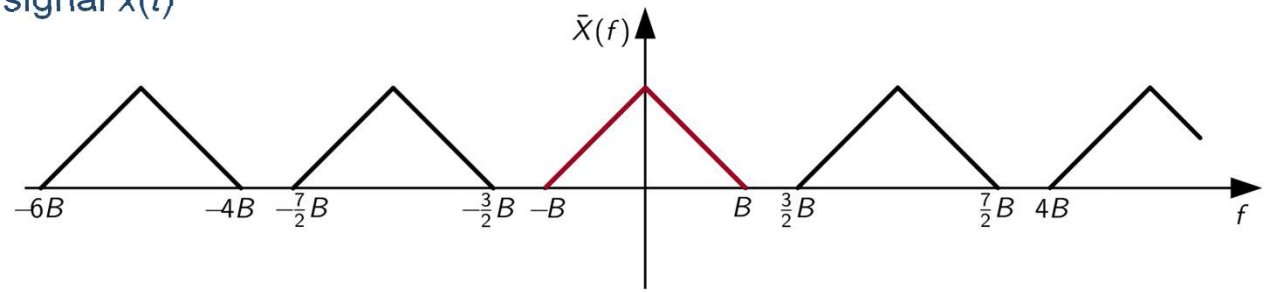
Interpolation (1/3)

How can we reconstruct the analog signal $x(t)$ starting from the sequence of samples $x[n]$?

This operation is called interpolation and is performed by the digital-to-analog converter (DAC)



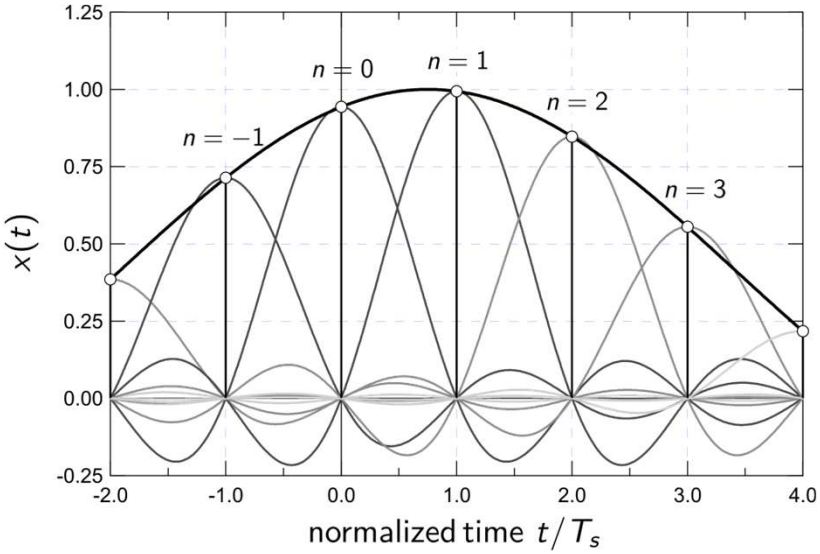
Ideally, we can apply a low-pass filtering on $x[n]$ to obtain back the original spectrum – and hence the original signal $x(t)$



Interpolation (2/3)

This goes under the name of cardinal interpolation, and is the result of the Shannon's sampling theorem:

$$x(t) = \sum_{n=-\infty}^{+\infty} x[n] \, \mathrm{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

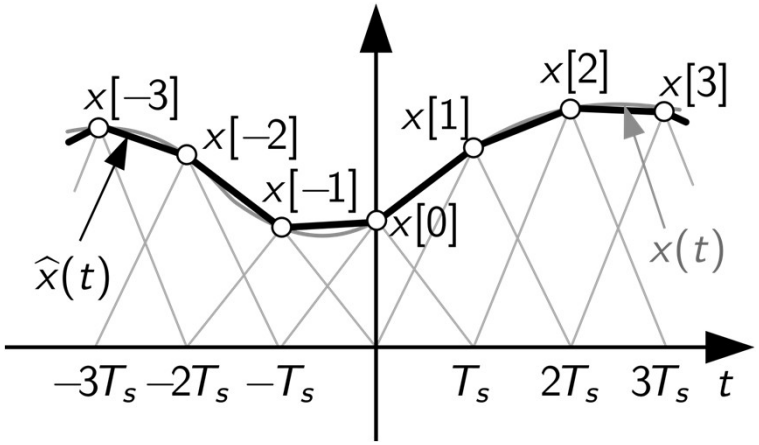


Interpolation (3/3)

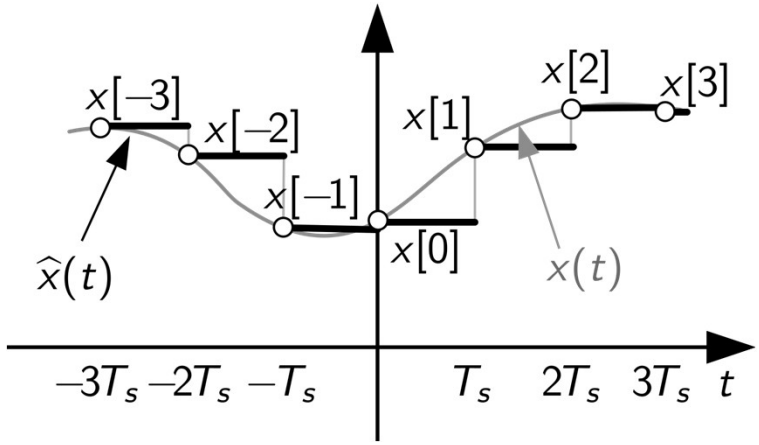
The cardinal interpolator calls for an infinite number of samples, including future ones.

Simpler methods, yet achieving good results that provide an approximated version $\hat{x}(t)$ of the original signal, include:

- linear interpolation

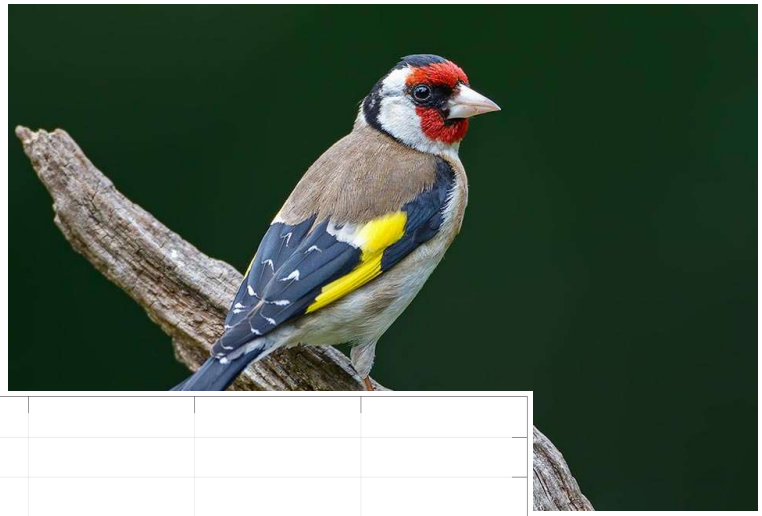


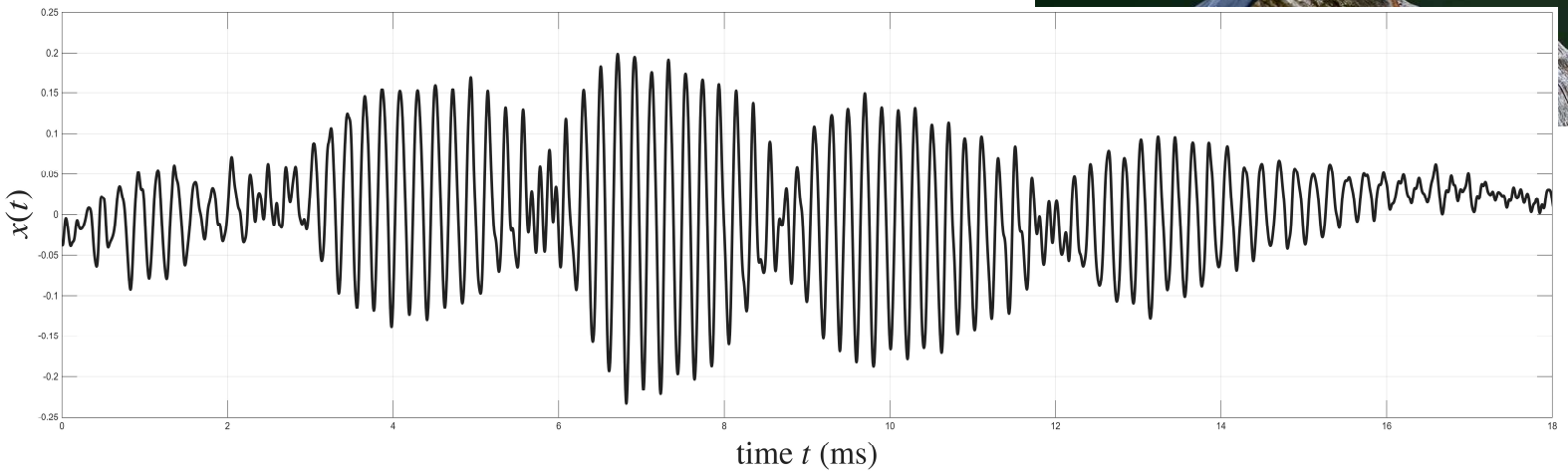
- sample & hold (S&H) interpolation



What happens when violating the Nyquist rule? (1/5)

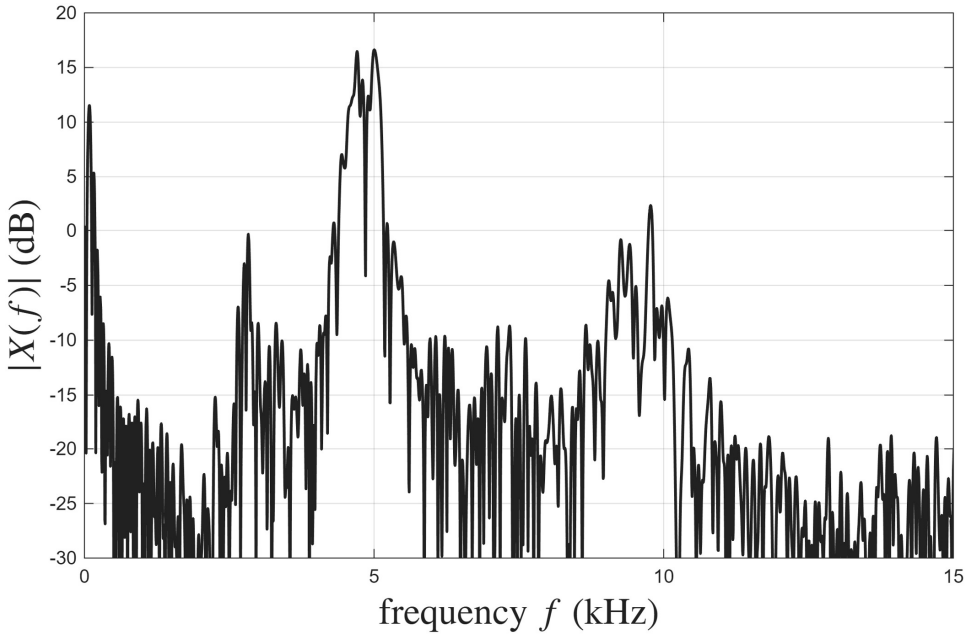
Whistle of a European goldfinch (*Carduelis carduelis*):





What happens when violating the Nyquist rule? (2/5)

Whistle of a European goldfinch (*Carduelis carduelis*):

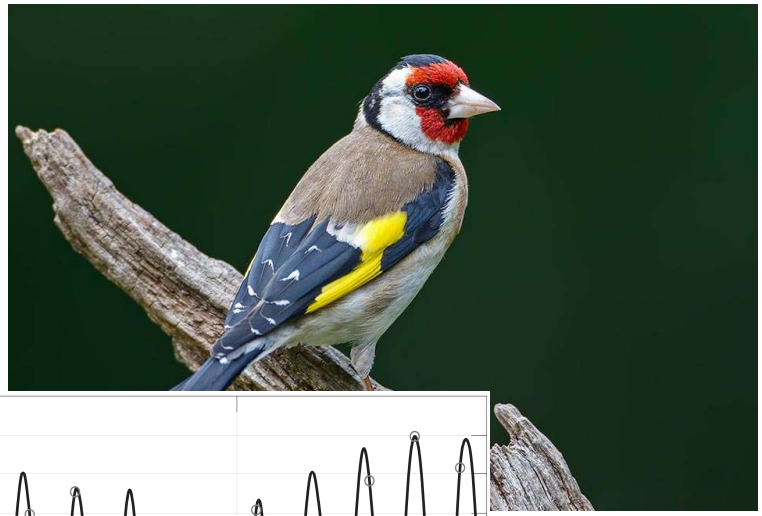


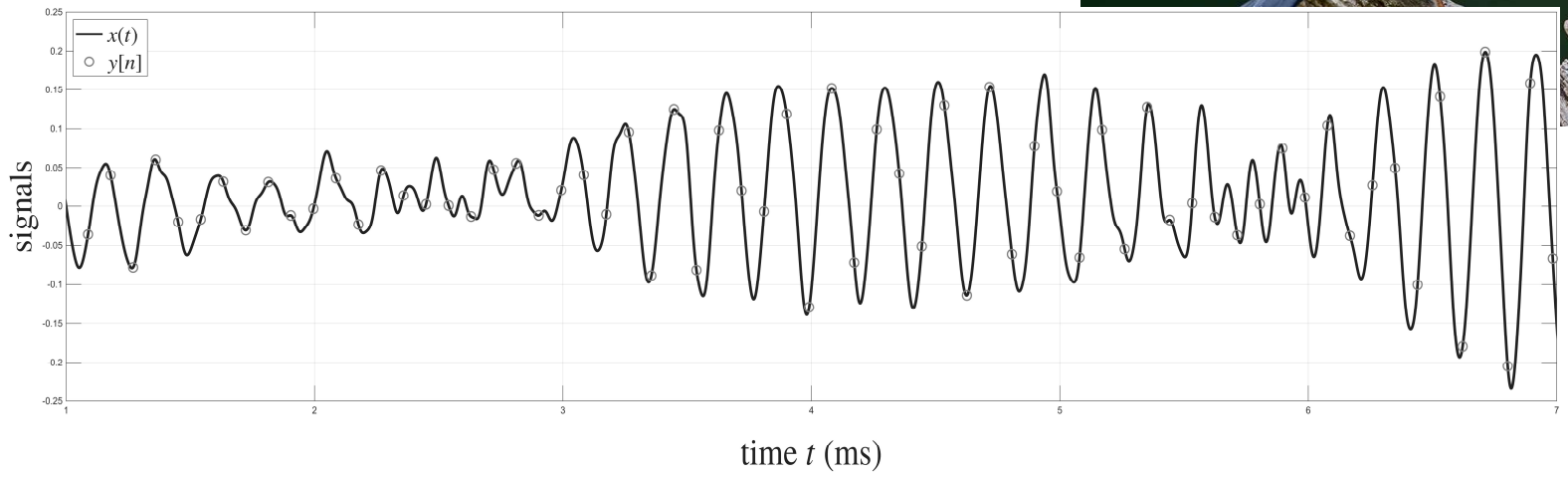


signal bandwidth $B \geq 10\,\text{kHz}$

What happens when violating the Nyquist rule? (3/5)

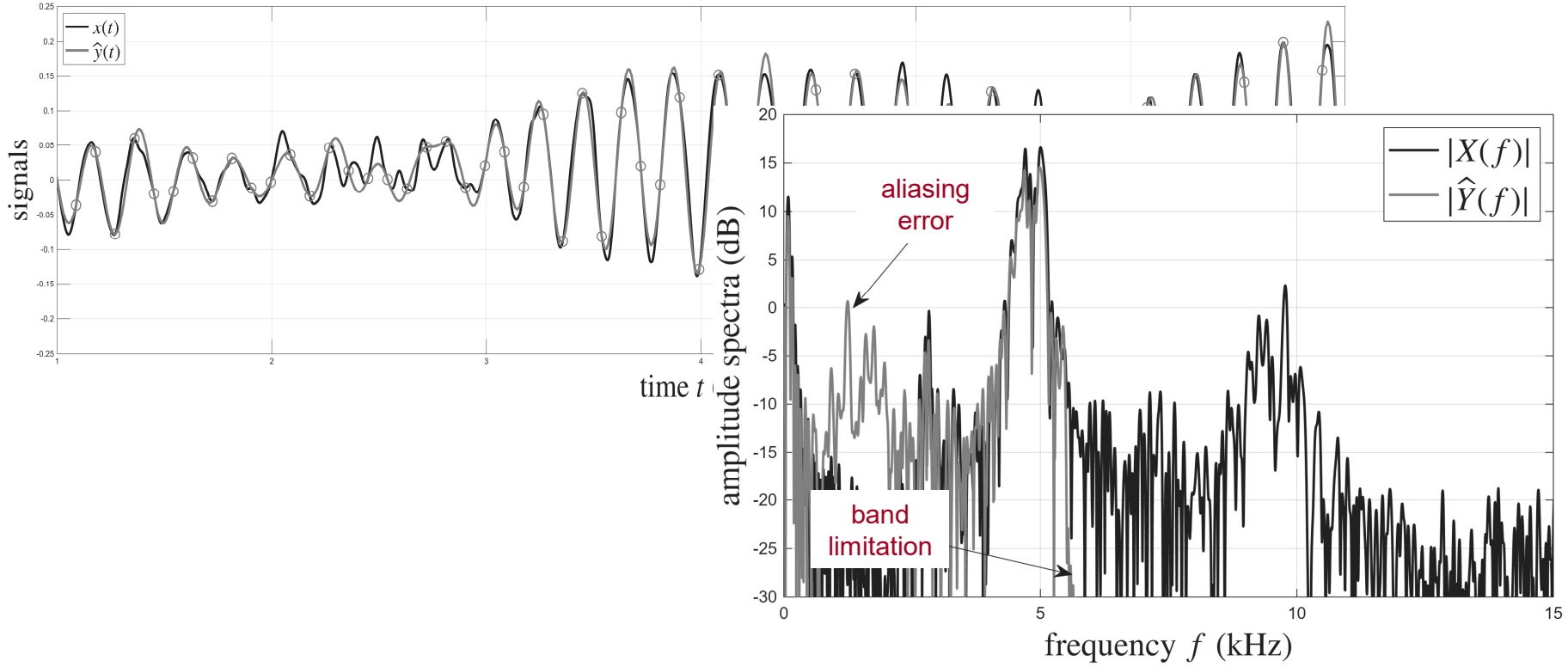
Sampling of the signal with f_s = 11.025 kHz < 2B:





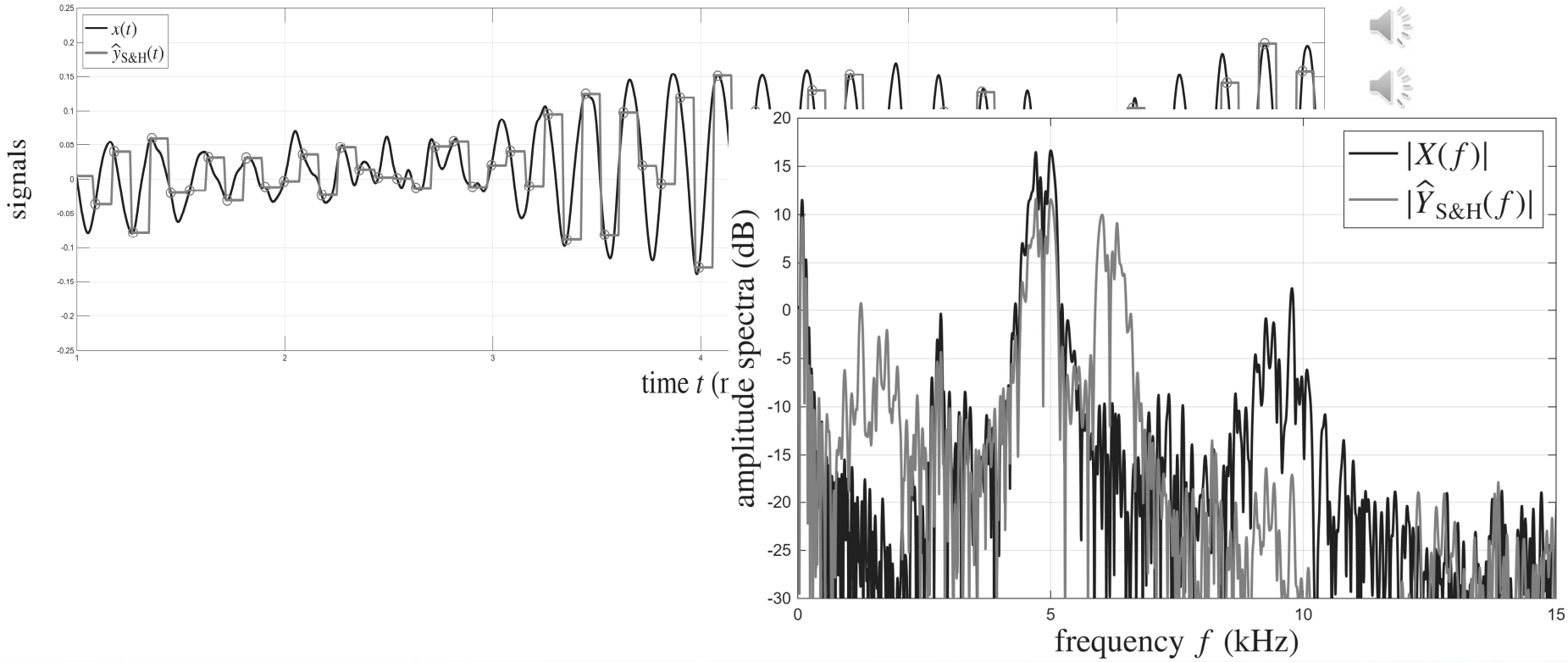
What happens when violating the Nyquist rule? (4/5)

Signal reconstruction using cardinal interpolator:



What happens when violating the Nyquist rule? (5/5)

Signal reconstruction using S&H interpolator:





Quantization

Quantization (1/3)

To transform an analog signal into a digital one, the job is not done just after sampling: we need to express the sample over a finite number of bits

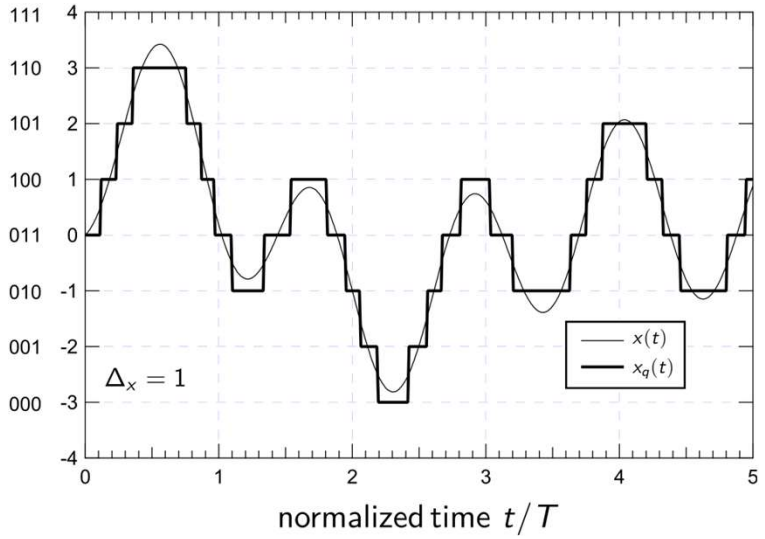
This task is called quantization, and works as follows:

$$x_q[n] = \Delta_x \cdot \left\lfloor \frac{x[n]}{\Delta_x} \right\rfloor$$

rounding

quantization step: $\Delta_x = \frac{D_x}{2^b - 1}$

b: number of quantization bits





Quantization (2/3)

The impact of a different number of bits (i.e., number of intensity levels) can be easily visualized in a 2D image:



8 bits / pixel



4 bits / pixel



2 bits / pixel

Quantization (3/3)

We can also experience it in a musical song (*Help!* by The Beatles):



- 16 bits/sample:
- 4 bits/sample:
- 2 bits/sample:



Source coding



Source coding

Source coding aims at data compression, involving encoding information to use fewer bits than the original representation:

- Lossless compression identifies and eliminates statistical redundancy
- Lossy compression identifies and removes unnecessary information

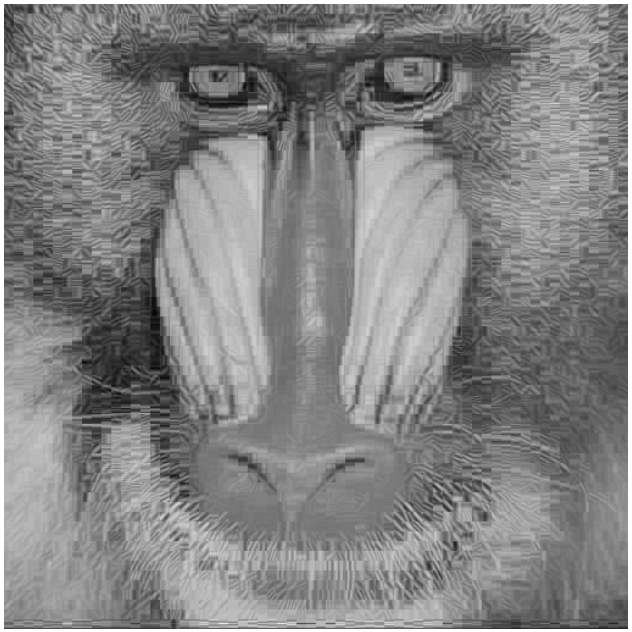




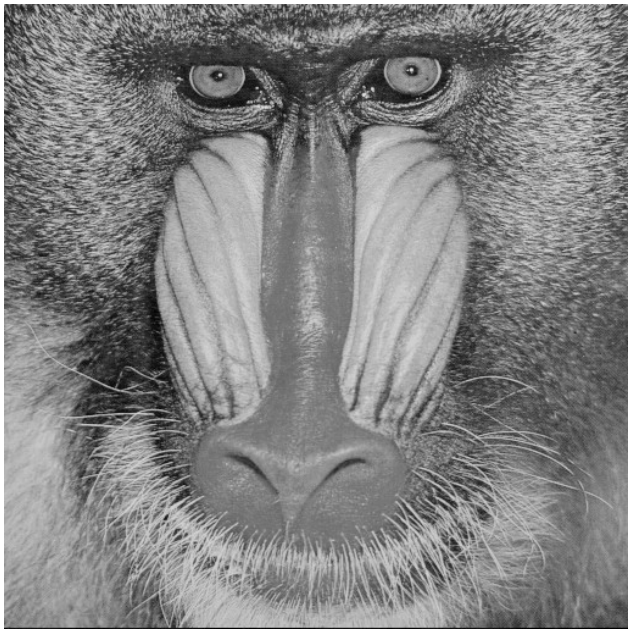
Image coding

An example of lossy compression: image coding

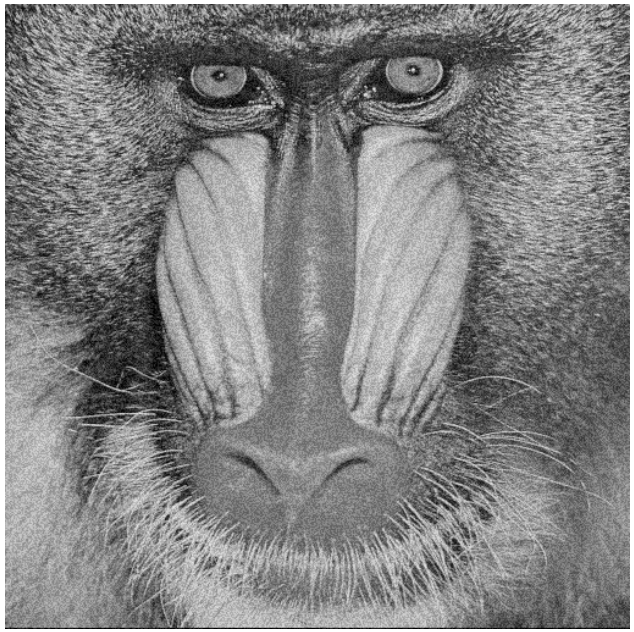
Image coding adopts a lossy compression, which is based on the specific properties of the human vision



altered #1



original

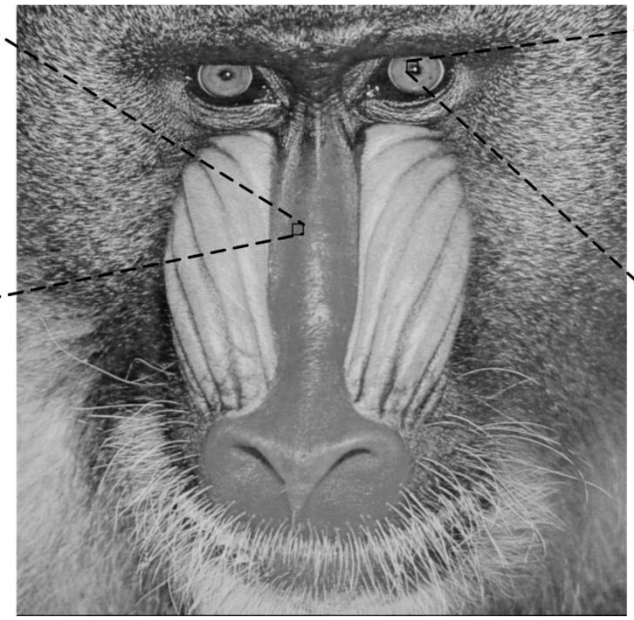


altered #2

Image representation

nose (intensity)

114	119	108	115	105	110	116	117
119	113	119	115	108	109	115	109
112	119	116	114	113	111	113	117
122	118	126	113	108	111	121	120
122	119	119	114	110	113	113	116
117	114	113	111	111	110	119	125
114	113	109	107	109	115	114	120
116	118	113	117	118	117	122	125



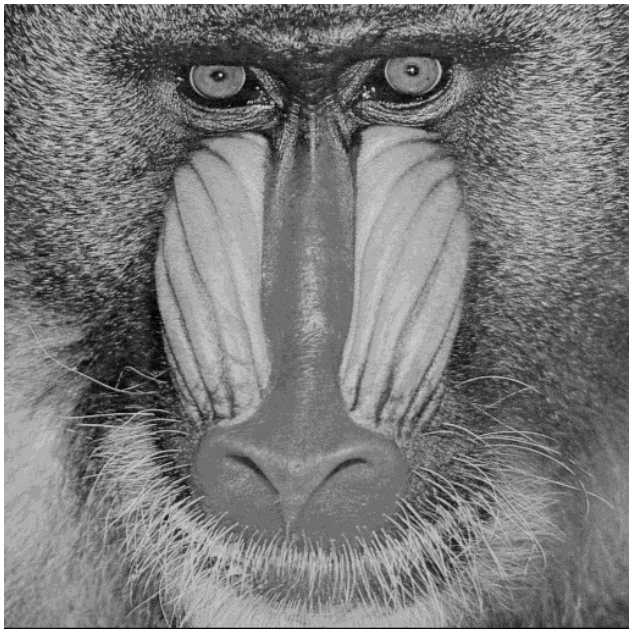
eye (intensity)

139	151	130	114	117	115	115	114
142	117	124	105	129	146	128	131
122	103	83	63	67	70	47	75
108	96	88	57	38	41	37	37
117	98	53	40	34	117	144	78
99	104	67	123	96	212	220	188
114	101	133	203	96	209	218	177
110	78	41	55	36	104	150	104

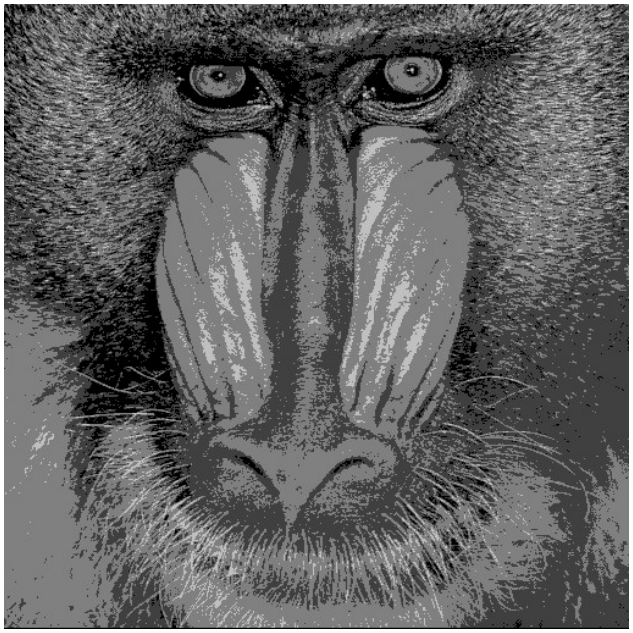
8 bits / pixel

Quantization in the space domain

Raw quantization:



4 bits / pixel



2 bits / pixel

Discrete cosine transform (DCT)

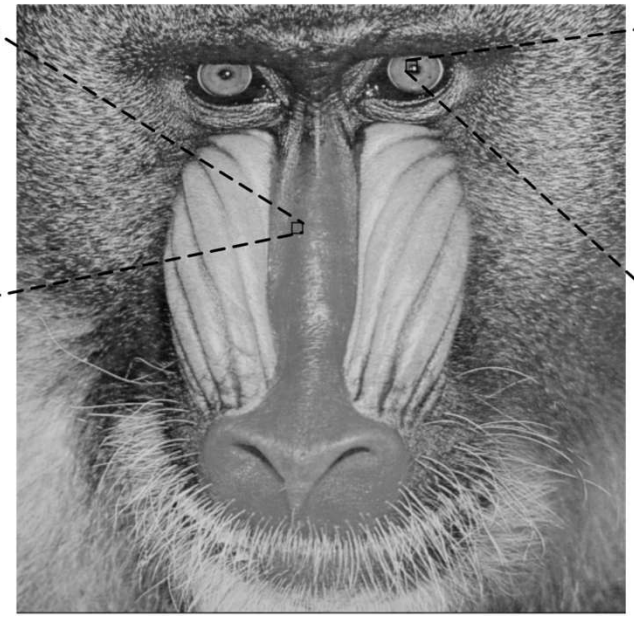
Image analysis using DCT:

nose (intensity)

114	119	108	115	105	110	116	117
119	113	119	115	108	109	115	109
112	119	116	114	113	111	113	117
122	118	126	113	108	111	121	120
122	119	119	114	110	113	113	116
117	114	113	111	111	110	119	125
114	113	109	107	109	115	114	120
116	118	113	117	118	117	122	125

nose (DCT spectrum)

920	1	19	−9	−2	3	−1	−1
−6	11	−3	−3	−3	5	−2	−1
−4	−8	−2	4	2	−1	−6	−6
−7	−1	6	1	2	1	0	−3
9	2	2	−1	−1	2	−1	−2
−3	−5	5	−1	−2	−3	0	−2
3	−6	0	−3	3	−5	−6	−2
−5	1	−3	3	0	−6	−1	−5



eye (intensity)

139	151	130	114	117	115	115	114
142	117	124	105	129	146	128	131
122	103	83	63	67	70	47	75
108	96	88	57	38	41	37	37
117	98	53	40	34	117	144	78
99	104	67	123	96	212	220	188
114	101	133	203	96	209	218	177
110	78	41	55	36	104	150	104

eye (DCT spectrum)

858	−47	90	51	−52	53	4	−47
−32	129	−27	−18	32	−72	17	48
116	−35	−16	−11	11	9	−4	−1
155	−91	−9	12	−29	16	−13	−27
−145	79	37	7	−14	−9	−20	27
28	35	−47	−50	15	11	7	−17
−54	−5	29	12	6	−28	−13	−2
−6	14	7	3	−8	12	−1	−4

Quantization: space vs. frequency (1/3)

Quantization tables:

1	1	1	2	4	8	16	32
1	1	2	2	4	8	16	32
1	2	4	4	8	16	32	64
2	2	4	8	16	32	32	64
4	4	8	16	16	32	64	128
8	8	16	32	32	128	128	256
16	16	32	32	64	128	256	256
32	32	64	64	128	256	256	256

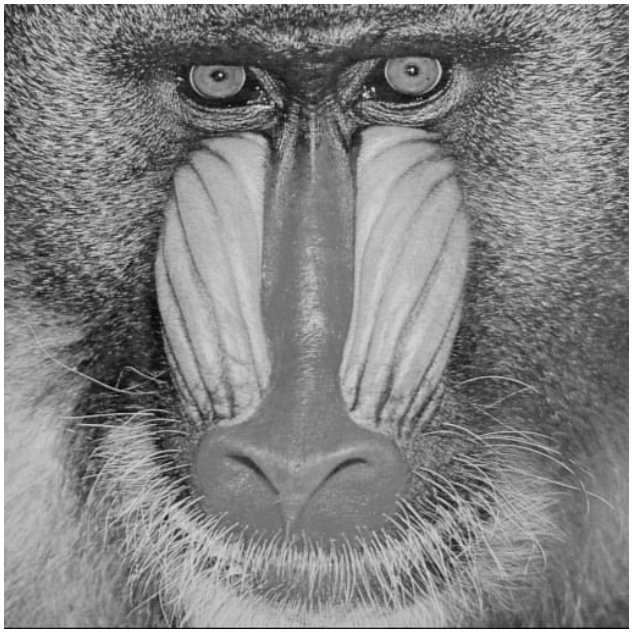
low compression

1	2	4	8	16	16	32	64
2	2	4	8	16	128	256	256
4	4	8	16	64	256	256	256
8	8	16	64	256	256	256	256
16	16	64	256	256	256	256	256
16	128	256	256	256	256	256	256
32	256	256	256	256	256	256	256
64	256	256	256	256	256	256	256

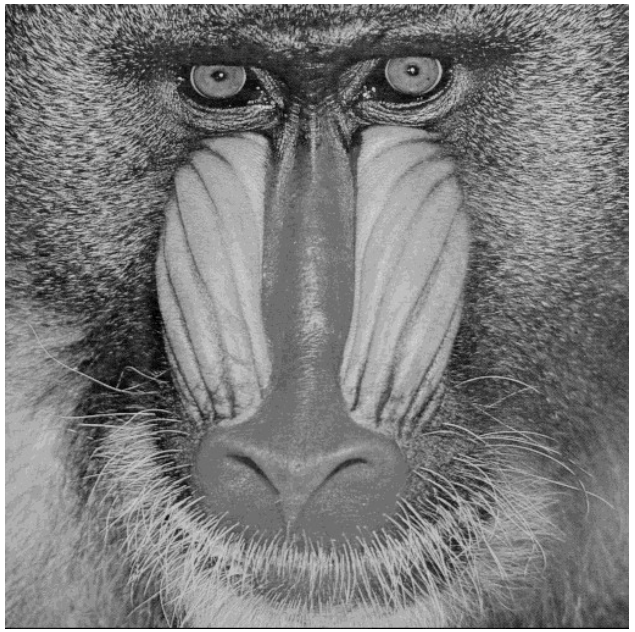
high compression

Quantization: space vs. frequency (2/3)

50% original file size



DCT-based quantization

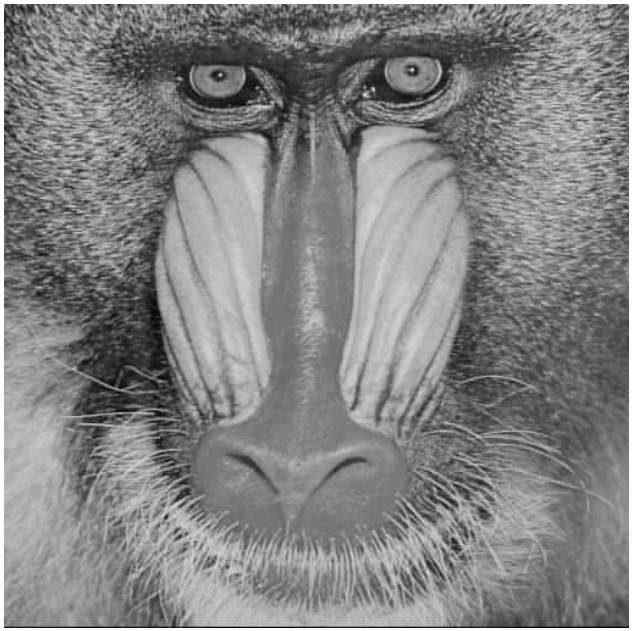


raw quantization

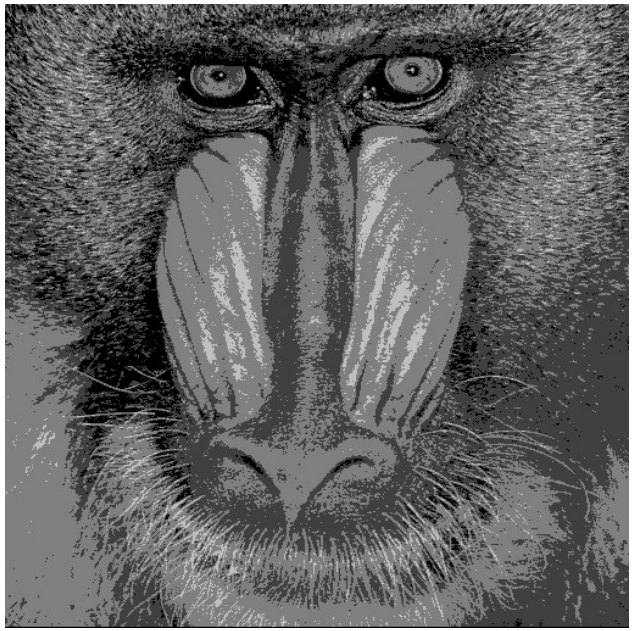


Quantization: space vs. frequency (3/3)

25% original file size



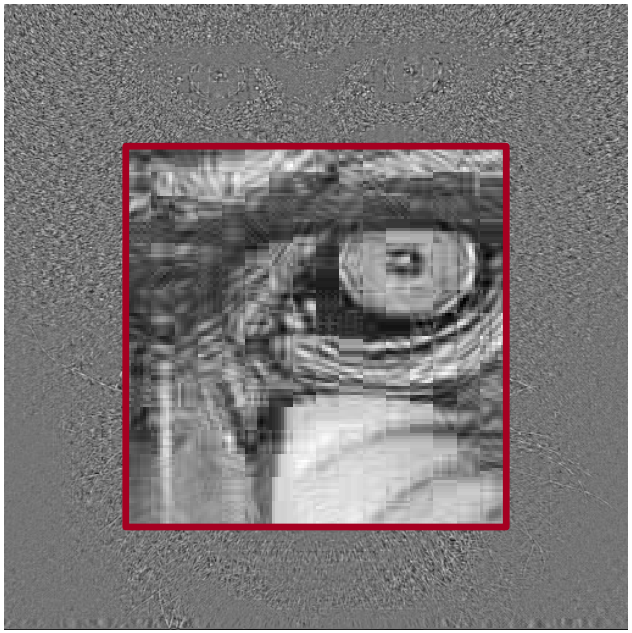
DCT-based quantization



raw quantization

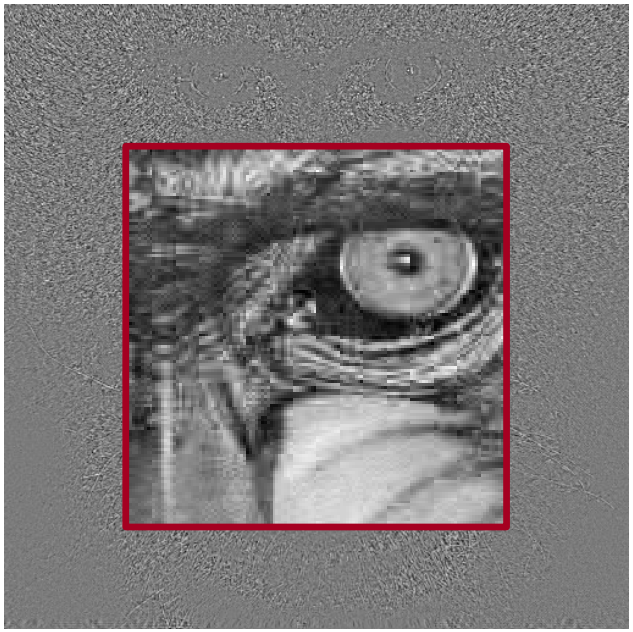
DFT vs. DCT

discrete Fourier transform (DFT)



vs.

discrete cosine transform (DCT)





The joint picture expert group (JPEG) algorithm

JPEG encoding scheme:



JPEG decoding scheme:

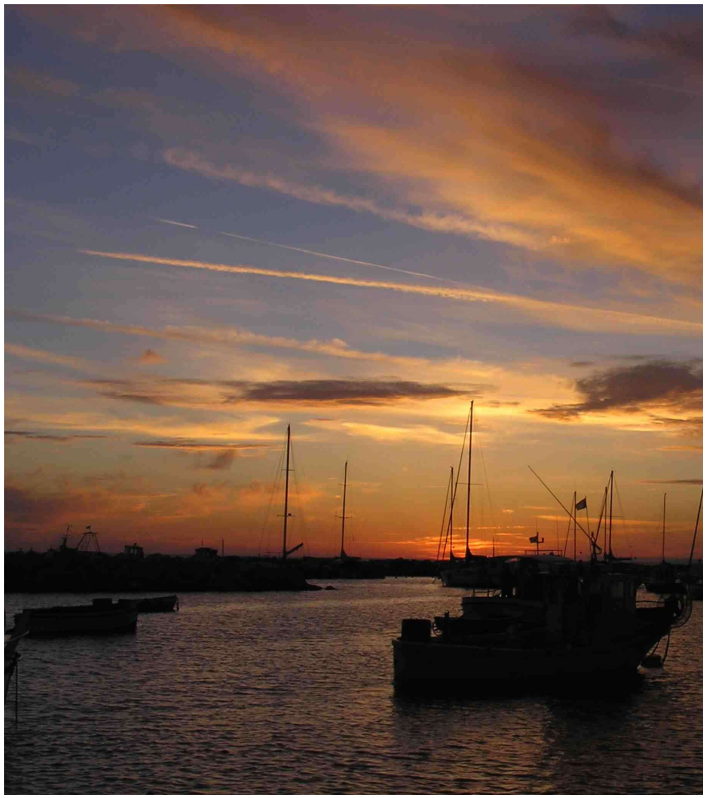




Examples of compressed images (1/4)

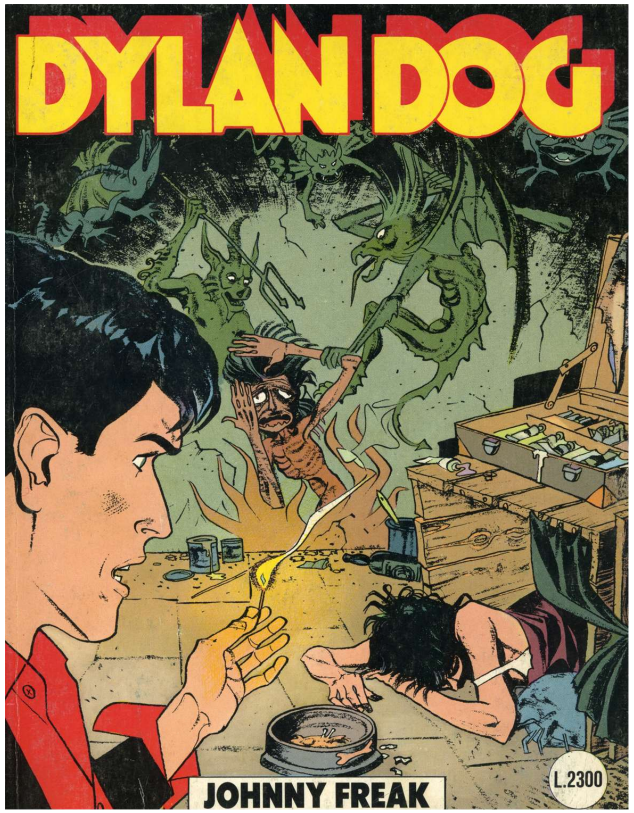


original image

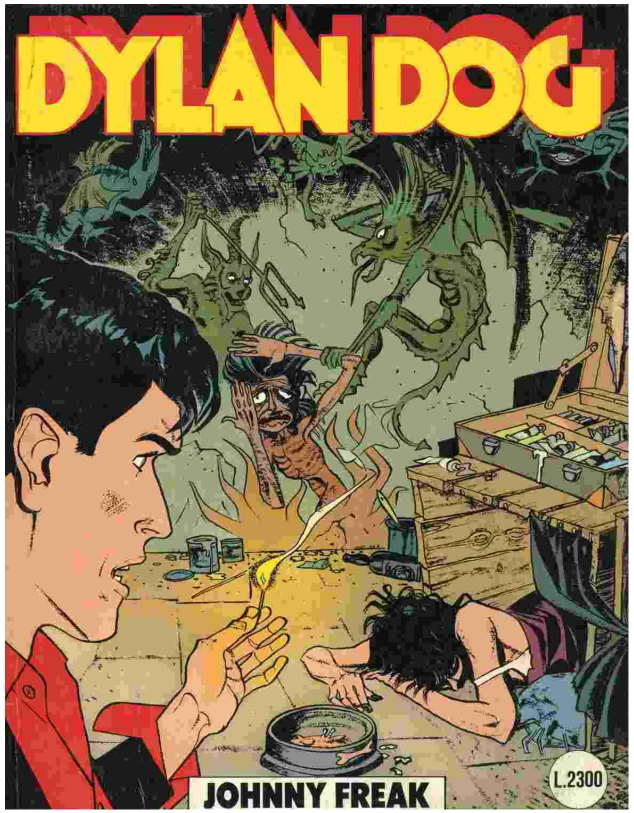


compressed image (ratio 1:10)

Examples of compressed images (2/4)

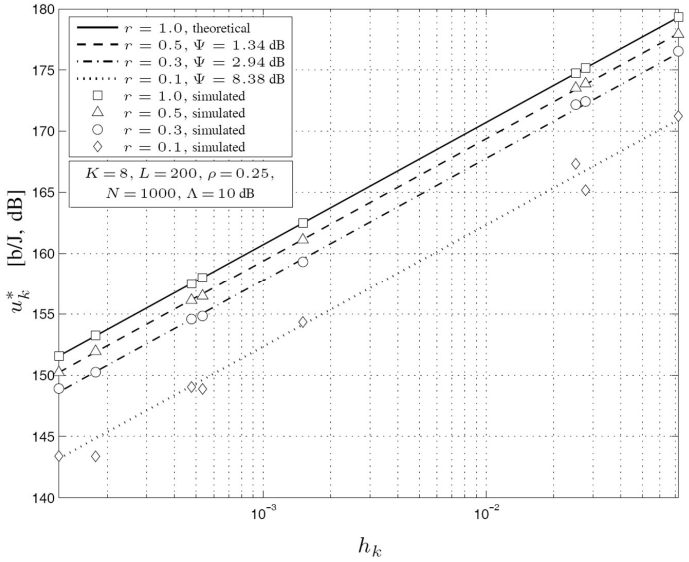


original image

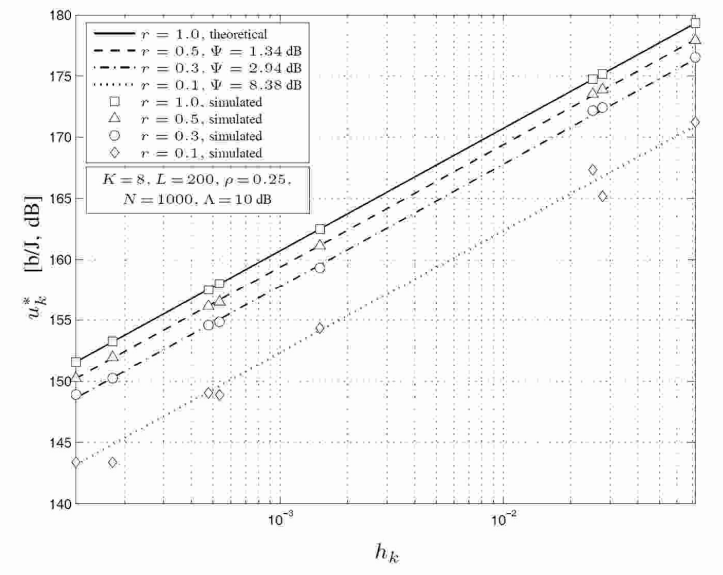


compressed image (ratio 1:10)

Examples of compressed images (3/4)



original image



compressed image (ratio 1:10)



Examples of compressed images (4/4)



2.66 MB



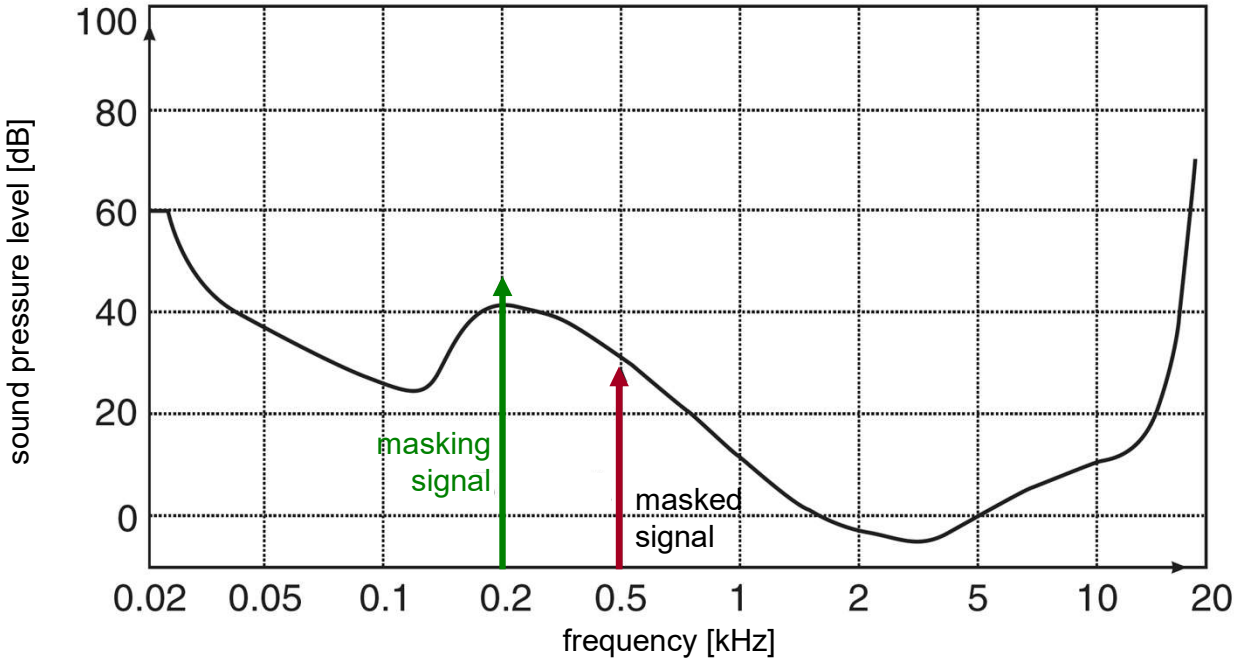
7.02 MB



Audio coding

Basics of audio coding (1/2)

Similarly, audio coding adopts a lossy compression, again based on the psycho-acoustic model





Basics of audio coding (2/2)

Audio coding, like image coding, is based on the semantics of the source:

- **Uncompressed, CD-quality audio (1.41 Mb/s):**
- **Compressed, MPEG-1/2 audio layer III (MP3) audio (32 kb/s):**
- **Compressed using JPEG algorithm (32 kb/s):**



Video coding



Why do we need compressed videos?

high-definition video (HD):

- component resolution (R, G, B): 8 bits / component
- color image composition: 3 components / pixel
- resolution: 1920 × 1080 pixels / frame
- image refresh rate: 60 frames / s

$$R_b = (8 \cdot 3 \cdot 1920 \cdot 1080 \cdot 60)\,\mathrm{b/s} \approxeq 2.99\,\mathrm{Gb/s}$$

ultra-high-definition video (UHD, aka 4k):

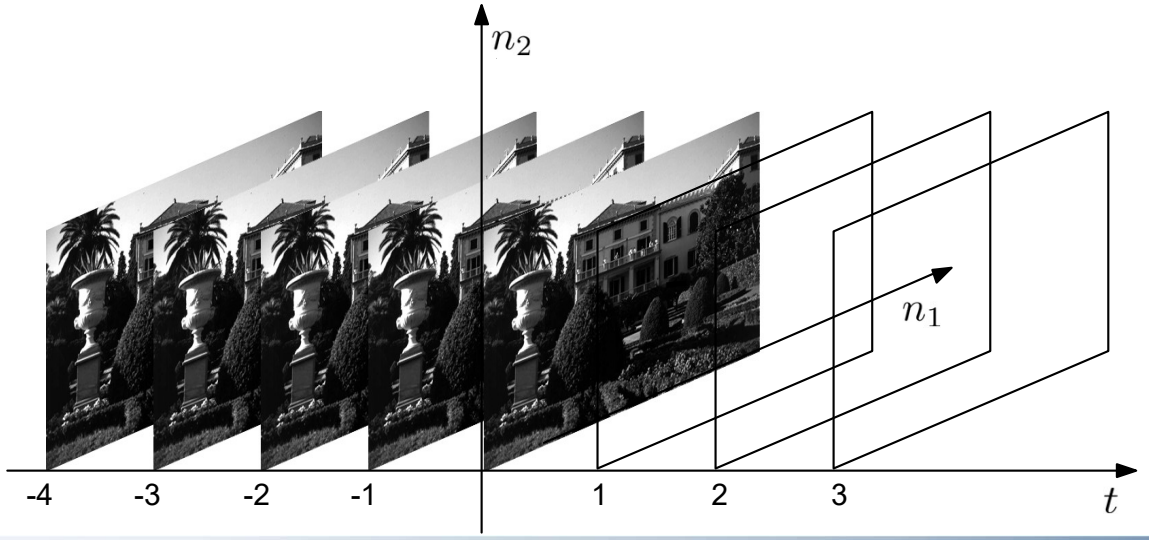
- 3840 × 2160 pixels / frame, using 120 f/s refresh rate and 48 bits per pixel

$$R_b = (16 \cdot 3 \cdot 3840 \cdot 2160 \cdot 120)\,\mathrm{b/s} \approxeq 47.78\,\mathrm{Gb/s}$$

Principles of video coding

We can compress a video stream, based on:

- **spatial correlation**, using the same principles exploited in image compressing
- **temporal correlation**, using time memory across successive frames





Examples of videos

poorly time-correlated video



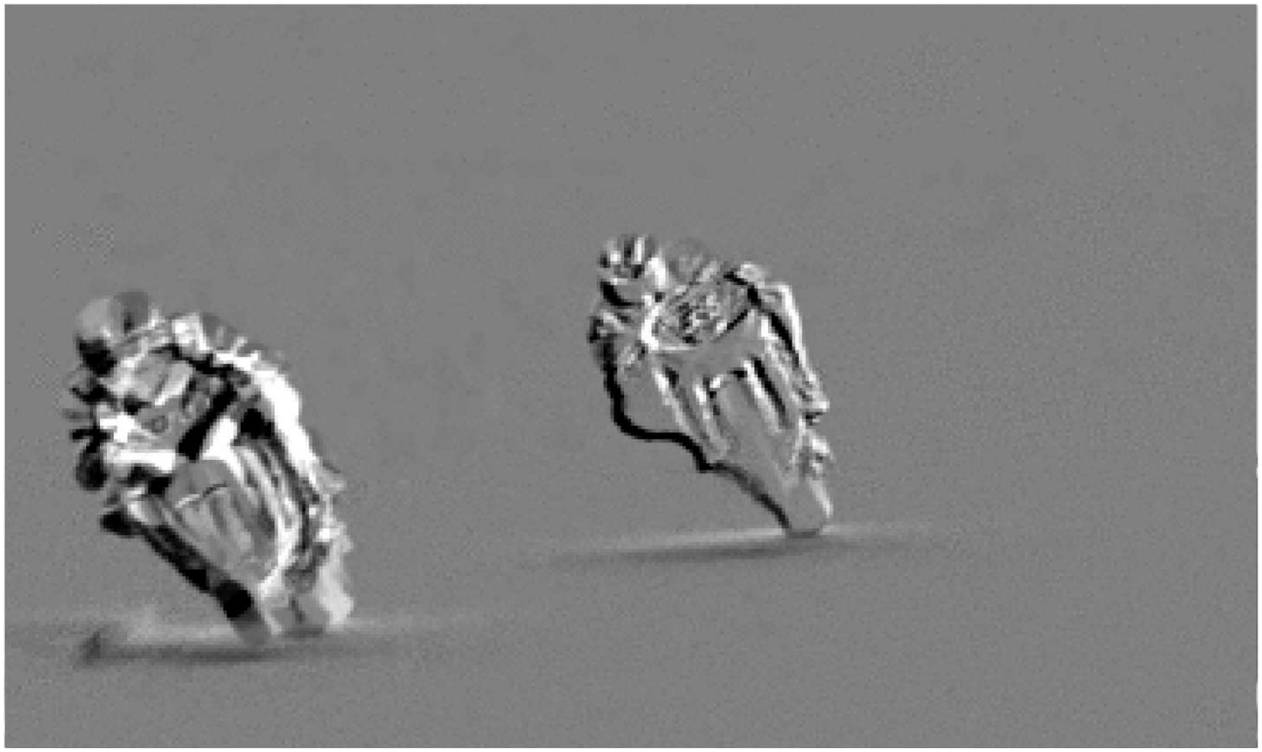
highly time-correlated video





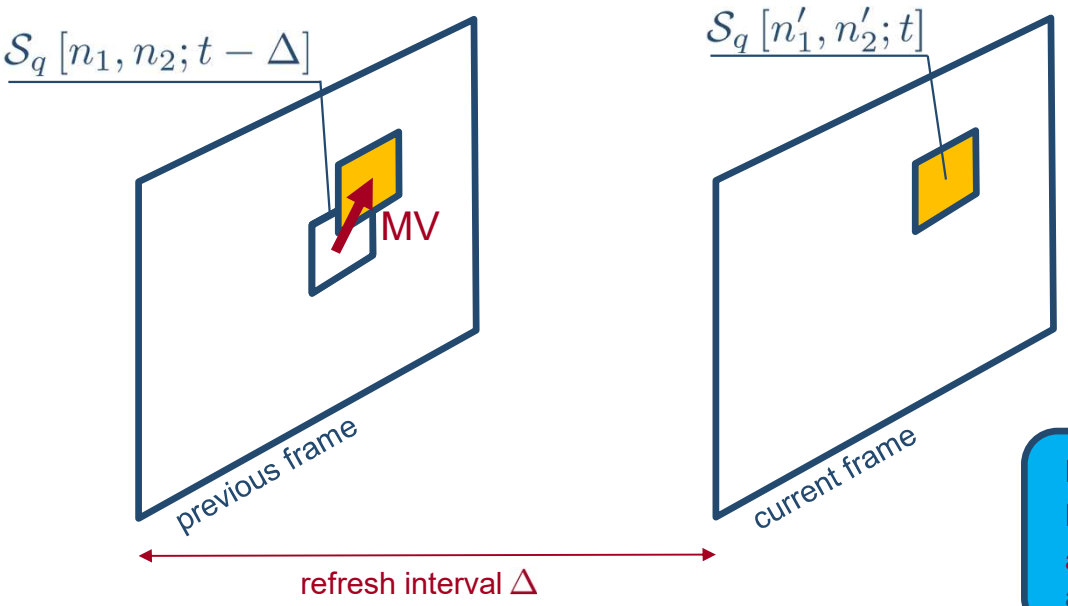
Time correlation

$$\mathcal{S}[n_1, n_2; t] - \mathcal{S}[n_1, n_2; t - \Delta]$$



Motion prediction (1/3)

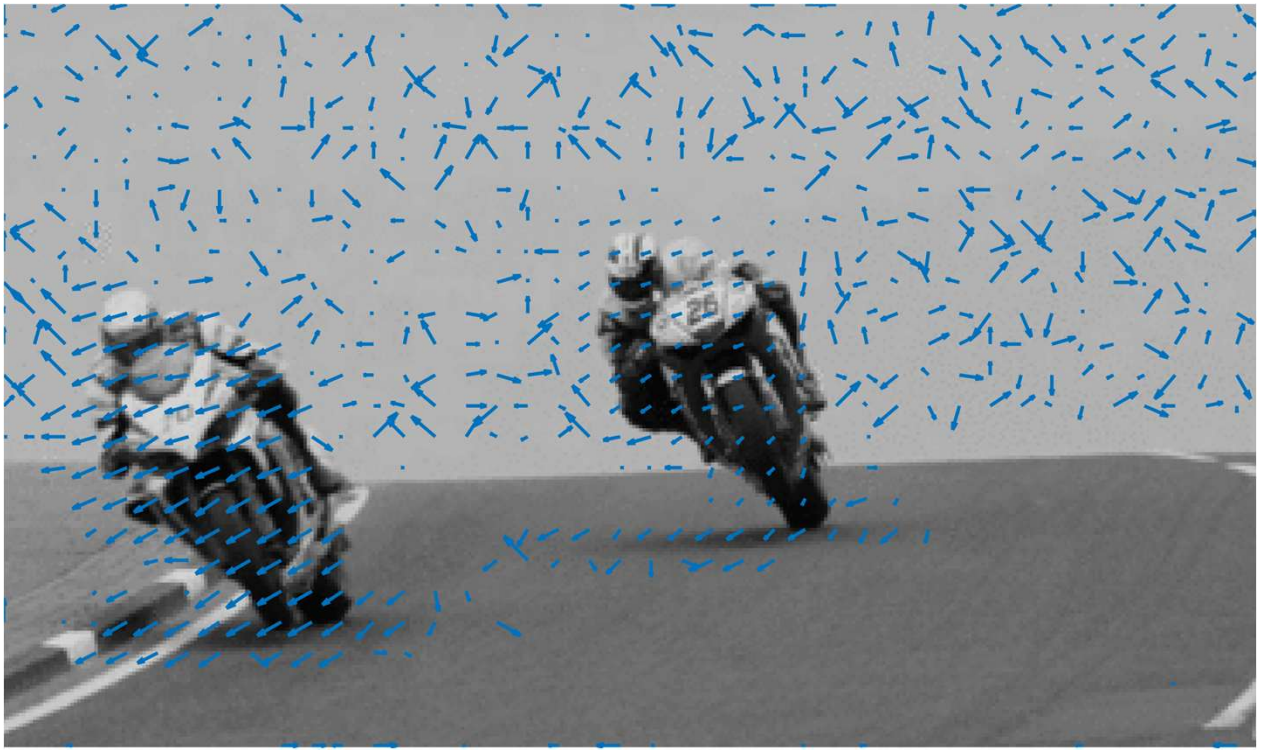
Effective video compression techniques rely on motion prediction based on motion vectors (MVs):



MPEG-1: 16×16 blocks
MPEG-2: 16×8 blocks
advanced standards: adaptive blocks



Motion prediction (2/3)





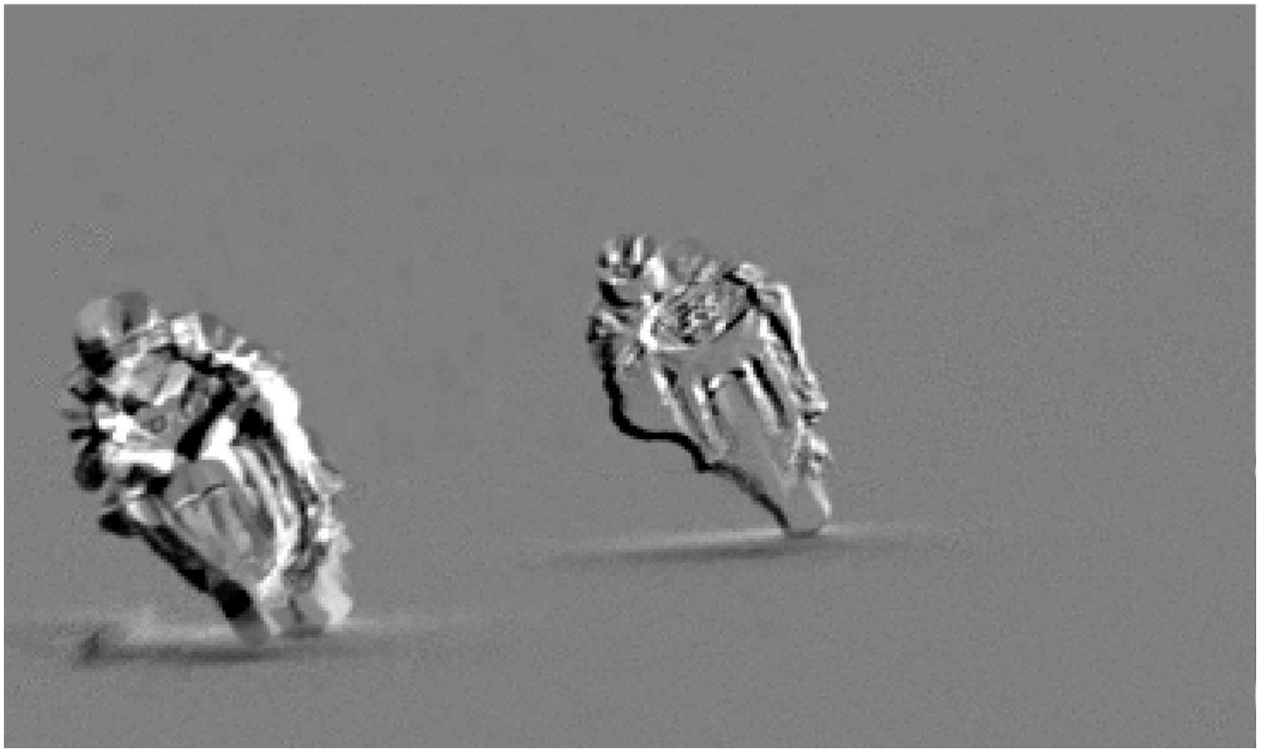
Motion prediction (3/3)





Motion compensation (1/2)

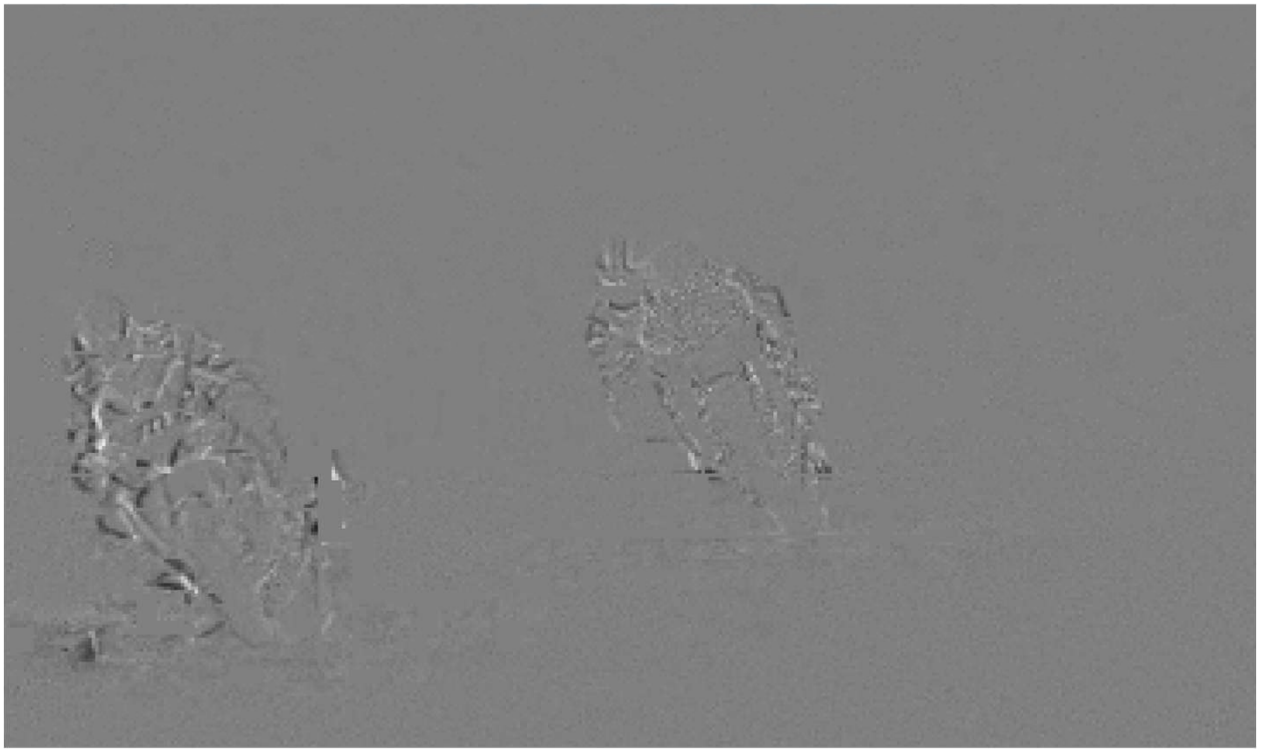
$$\mathcal{S}[n_1, n_2; t] - \mathcal{S}[n_1, n_2; t - \Delta]$$ **(without MC)**





Motion compensation (2/2)

$$\mathcal{S}[n_1, n_2; t] - \hat{\mathcal{S}}[n_1, n_2; t]$$ **(with MC)**





Examples of MPEG-2 compressed videos (1/3)



6 MB/s



4 MB/s



1.5 MB/s

Examples of MPEG-2 compressed videos (2/3)



6 MB/s



4 MB/s



1.5 MB/s

Examples of MPEG-2 compressed videos (3/3)



6 MB/s



4 MB/s



1.5 MB/s



Evolution of video compression standards (1/4)

The MPEG-2 standard (a.k.a. H.222/H.262) was designed to target standard definition (SD) used for DVD videos (first release: 1996)

With the increasing demand for additional quality, other standards have been deployed:

- advanced video coding (AVC), also referred to as H.264 or MPEG-4 Part 10 (first release: 2004), targets high definition (HD), thanks to adaptive macro-blocks, more effective entropic coding and lossy compression techniques
- high efficiency video coding (HEVC), also known as H.265 and MPEG-H Part 2 (first release: 2013), targets ultra-high-definition (UHD), thanks to improved MV prediction and motion compensation



Evolution of video compression standards (2/4)

MPEG-2: 3.54 MB



Evolution of video compression standards (3/4)

AVC: 1.19 MB





Evolution of video compression standards (4/4)

HEVC: 1.04 MB





Lossless source coding



Examples of lossless compression

- fixed-to-fixed code: ASCII code
- fixed-to-variable code: Morse code
- variable-to-fixed code: Lempel-Ziv-Welch algorithm

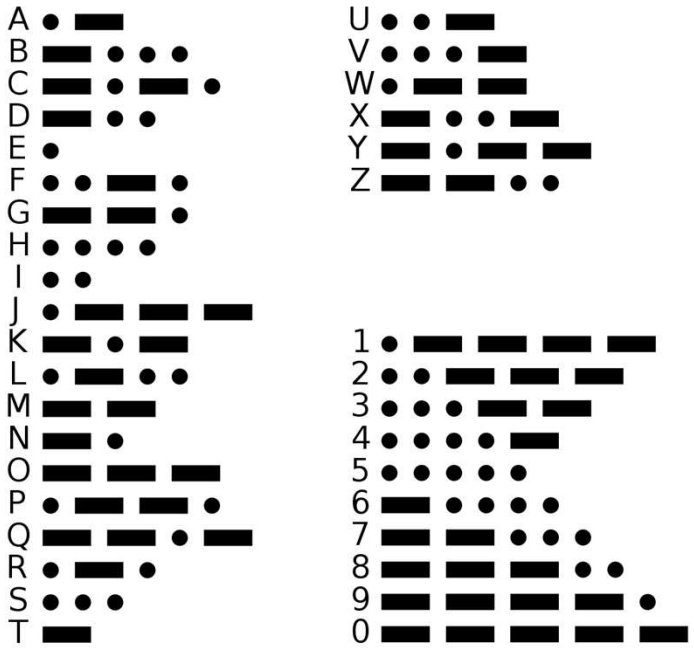
An example of lossless compression: The ASCII code

dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char
0	0	000	NULL	32	20	040	space	64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051	)	73	49	111	I	105	69	151	i
10	a	012	LF	42	2a	052	*	74	4a	112	J	106	6a	152	j
11	b	013	VT	43	2b	053	+	75	4b	113	K	107	6b	153	k
12	c	014	FF	44	2c	054	,	76	4c	114	L	108	6c	154	l
13	d	015	CR	45	2d	055	-	77	4d	115	M	109	6d	155	m
14	e	016	SO	46	2e	056	.	78	4e	116	N	110	6e	156	n
15	f	017	SI	47	2f	057	/	79	4f	117	O	111	6f	157	o
16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1a	032	SUB	58	3a	072	:	90	5a	132	Z	122	7a	172	z
27	1b	033	ESC	59	3b	073	;	91	5b	133	[	123	7b	173	{
28	1c	034	FS	60	3c	074	<	92	5c	134	\	124	7c	174	|
29	1d	035	GS	61	3d	075	=	93	5d	135	]	125	7d	175	}
30	1e	036	RS	62	3e	076	>	94	5e	136	^	126	7e	176	~
31	1f	037	US	63	3f	077	?	95	5f	137	_	127	7f	177	DEL

An example of lossless compression: The Morse code

International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.





The Lempel-Ziv-Welch (LZW) algorithm

- Lempel and Ziv patented the algorithm in 1978 as LZ78
- Welch further improved the algorithm in 1984
- easy and universal implementation, widely used for zip (and its variants) and GIF formats
- the original LZW algorithm encodes sequences as fixed-length 12-bit sequences



Encryption



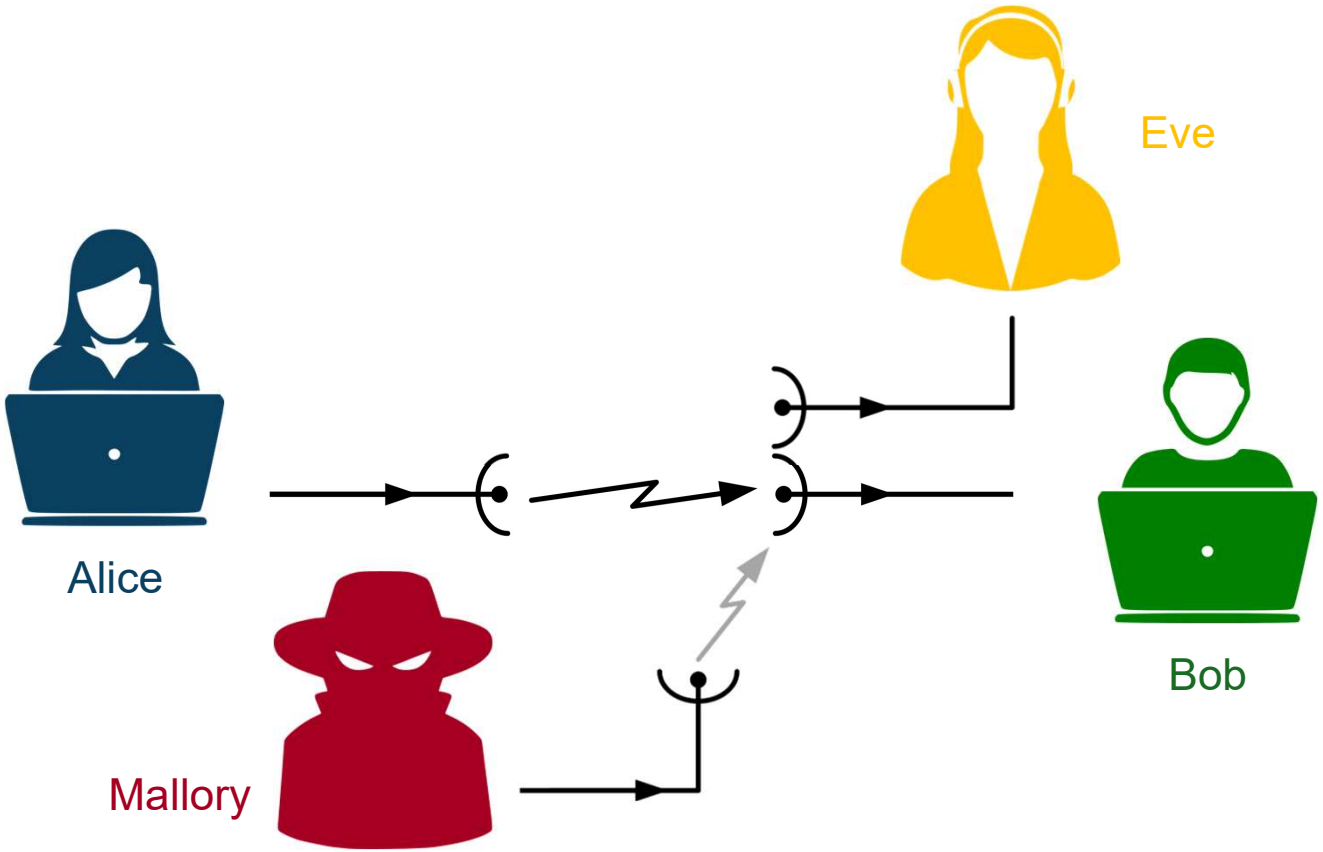
Features of encrypted communications

- Confidentiality: third-party cannot read exchanged data
  attacks: eavesdropping, sniffing
- Integrity: third-party cannot change exchanged data
  attacks: man-in-the-middle
- Authentication: each party is sure who is really communicating with
  attacks: masquerading, spoofing, traffic generation

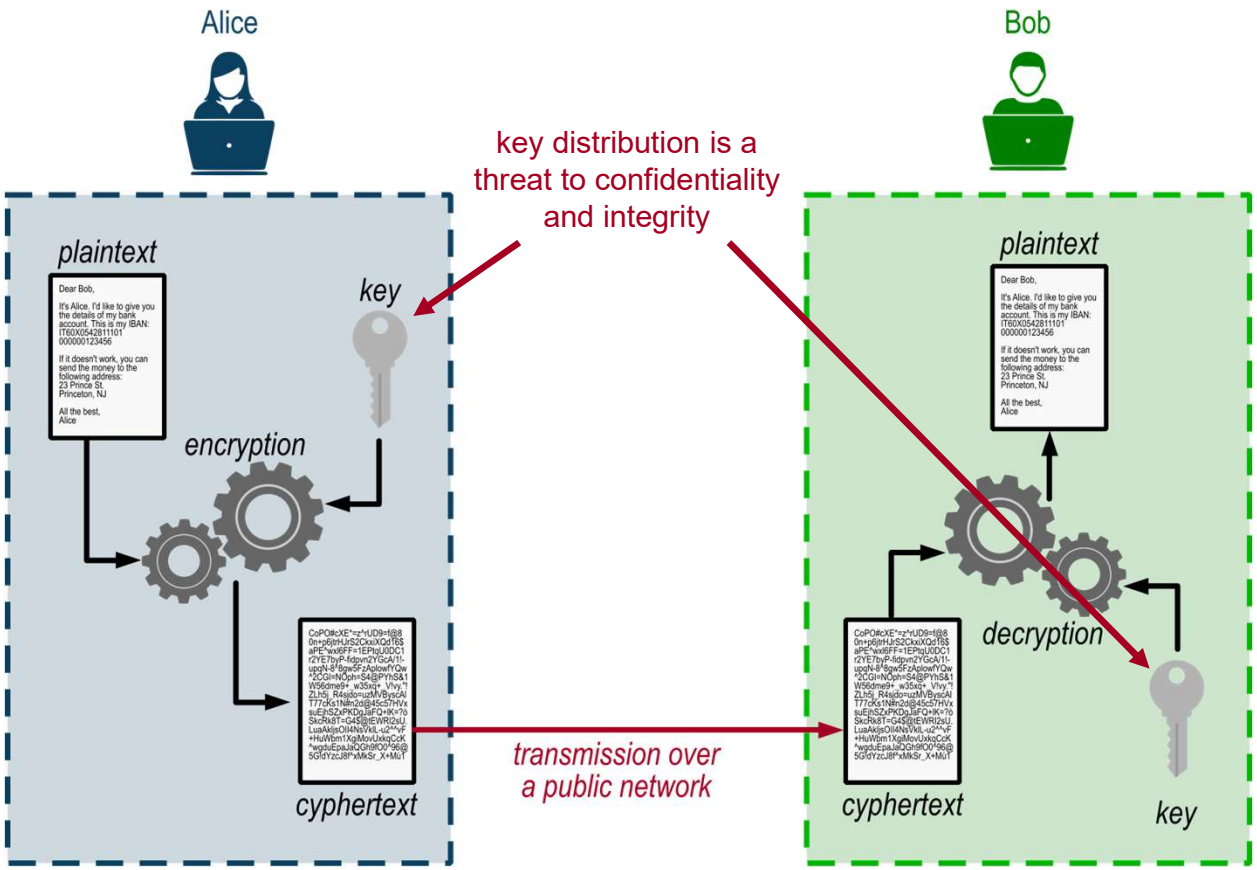


- Availability: time the system is in a functioning condition
  attacks: denial of service (DoS), distributed DoS (DDoS)

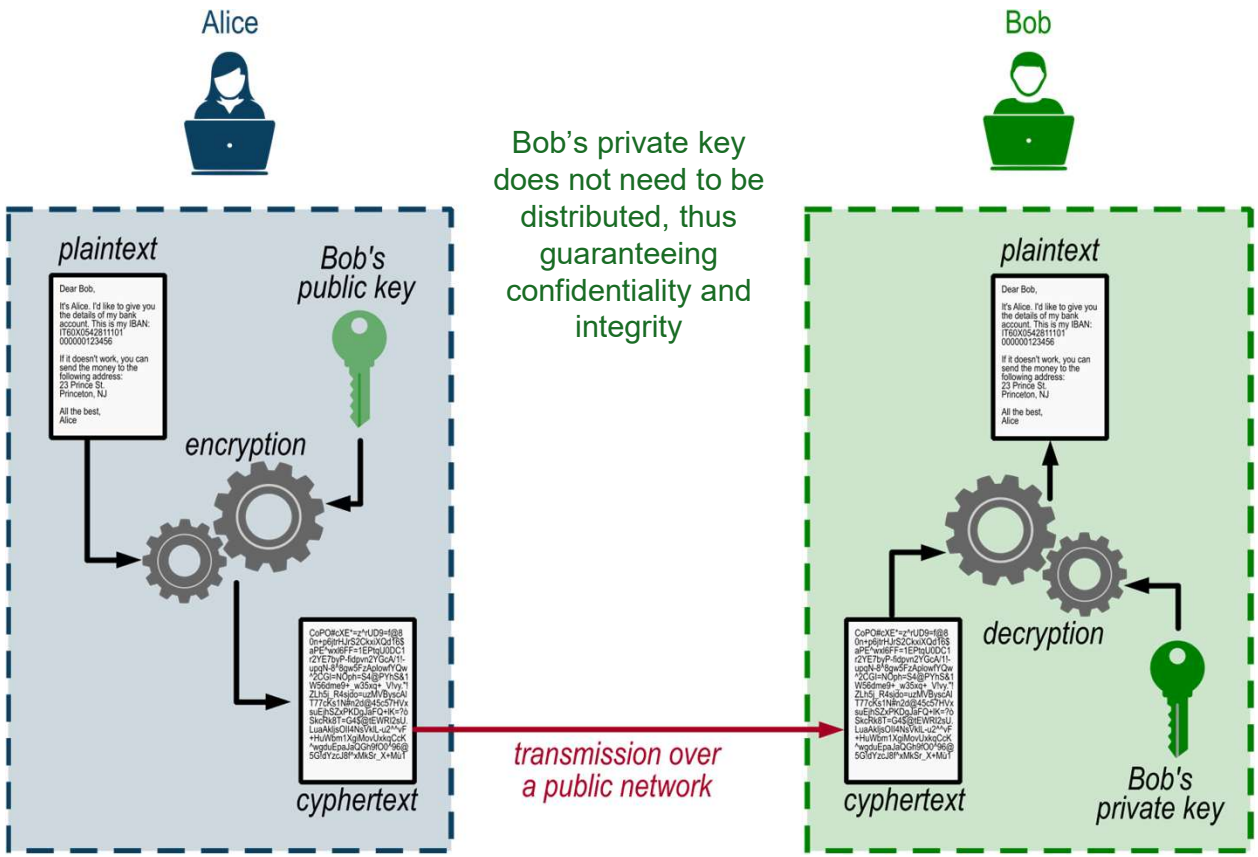
Players in a cybersecurity scenario



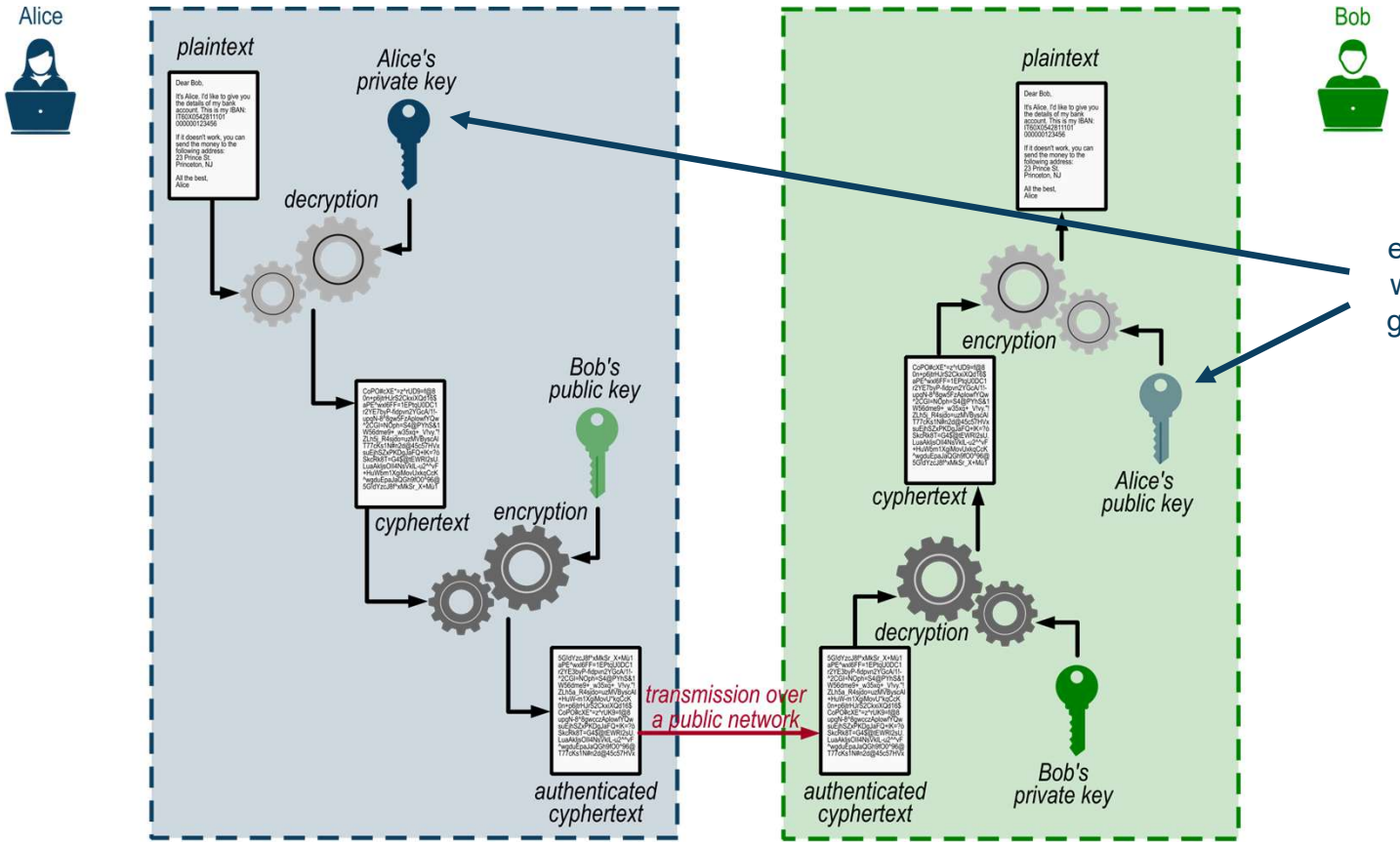
Encryption using a secret (symmetric) key



Encryption using a public (asymmetric) key



Adding authentication using a private key



encrypting the plaintext with Alice's private key guarantees authenticity to the message



Channel coding



Motivating example

Compare your error-correcting skills between this message:

> Tihs messgae cnotains smoe typso, btu I guses yuo cna egt ist contnet whit osme effrot...

and this other message:

> Wno m'I fdaira uyo tnacon sgues thwa s'ti hreet, sa erhte rea ywa oto namy rrores...

Any language (like English, Italian, etc.) shows a structure, which helps the listeners to reconstruct the meaning of a sentence, even in presence of mispelled or unknown words!

General principles (1/5)

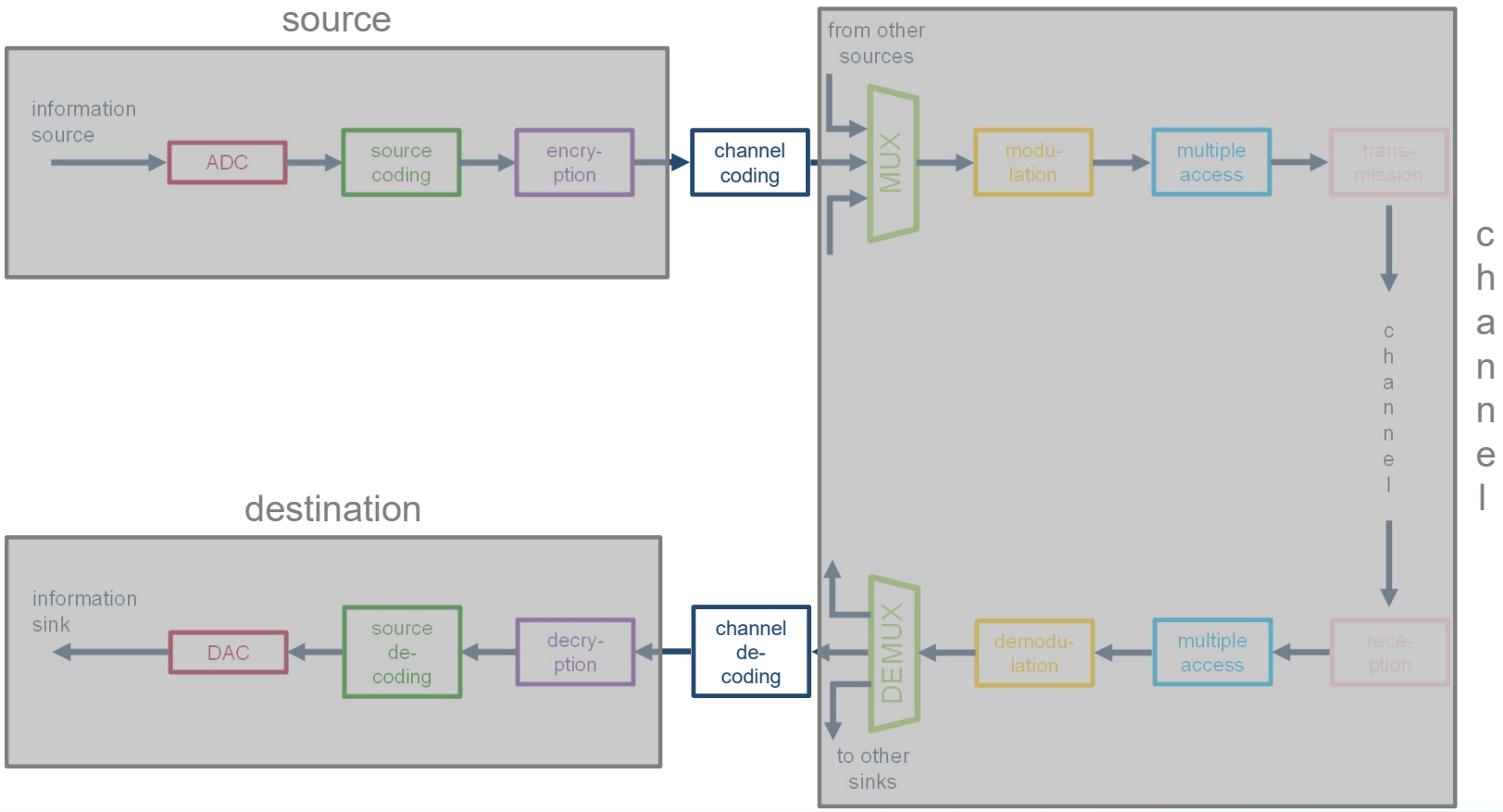
Channel coding attempts to do the same: it gives the channel-encoded message a particular structure, such that bit errors introduced by signal propagation through the error-prone channel can be detected (and potentially corrected) at the receive side



Two ways of recovering the packets:

- forward error correction (FEC)
- automatic repeat request (ARQ)

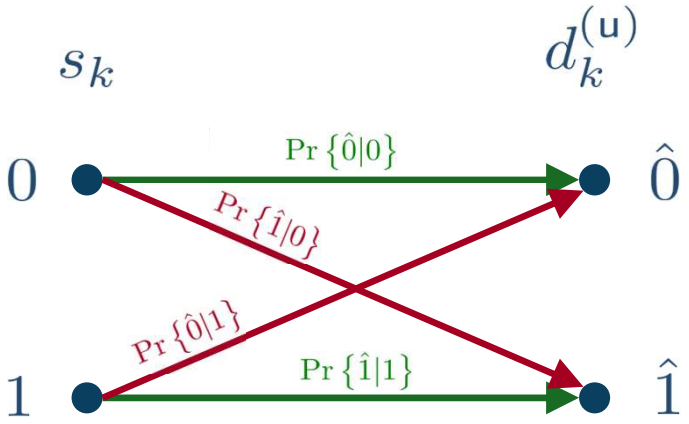
General principles (2/5)



General principles (3/5)

Let us suppose we do not use channel coding:





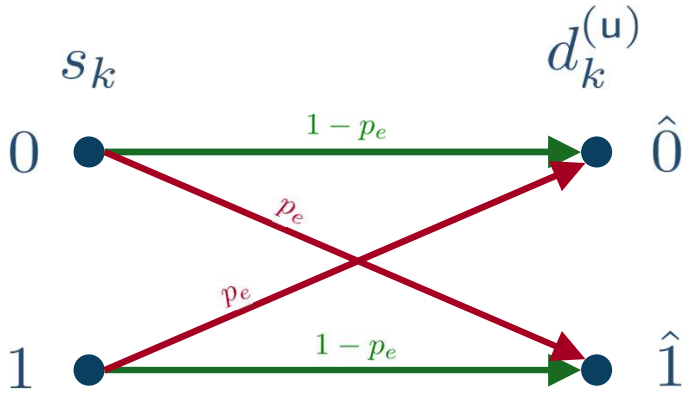
The quantities $\Pr\{\hat{1}|0\}$ and $\Pr\{\hat{0}|1\}$ are the probabilities that a source bit is flipped by the channel, i.e., the probabilities of errors

In practical systems, the probability of error can be measured in terms of bit error rate (BER)

General principles (4/5)

A customary assumption is to consider $\Pr\{\hat{1}|0\} = \Pr\{\hat{0}|1\} = p_e$, in the so-called binary symmetric channel (BSC):





Other channel models exist (e.g., the Z-channel that mimics an optical-fiber connection)

General principles (5/5)

Our goal is to introduce a channel coding stage such that the end-to-end error probability P_e is (much) lower than the bit-error probability p_e introduced by the BSC:



$$P_e = \Pr\left\{d_k^{(c)} \neq s_k\right\} < \Pr\left\{d_k^{(u)} \neq s_k\right\} = p_e$$

Channel coding for dummies: Repetition coding (1/4)

The simplest way to make the communication more “robust”, against bit flipping introduced by the channel, is to use the repetition code scheme: for each source bit, we send *N* copies

Example: *N*=5

$$s_k = 0 \Rightarrow \mathbf{c} = [0,0,0,0,0]$$
$$s_k = 1 \Rightarrow \mathbf{c} = [1,1,1,1,1]$$

At the receiver, we can apply a decision strategy based on a majority rule:

- If at least (*N*+1)/2 received coded bits are 0, we set $d_k^{(c)} = \hat{0}$, and $d_k^{(c)} = \hat{1}$ otherwise

Channel coding for dummies: Repetition coding (2/4)

What is the detection and correction performance of such approach?

$$P_e = \Pr\left\{\frac{N+1}{2}\text{ errors}\right\} + \Pr\left\{\frac{N+3}{2}\text{ errors}\right\} + \cdots + \Pr\left\{N\text{ errors}\right\}$$

Assuming that bit-flipping errors introduced by the channel are independent events,

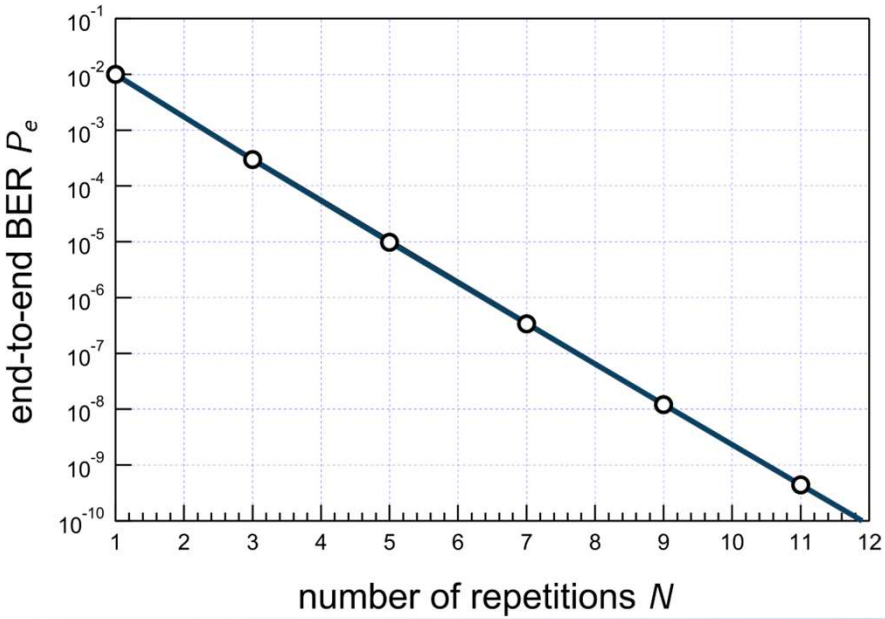
$$\Pr\left\{n\text{ errors}\right\} = \binom{N}{n} p_e^n \cdot (1-p_e)^{N-n} = \frac{N!}{n!(N-n)!} p_e^n \cdot (1-p_e)^{N-n}$$

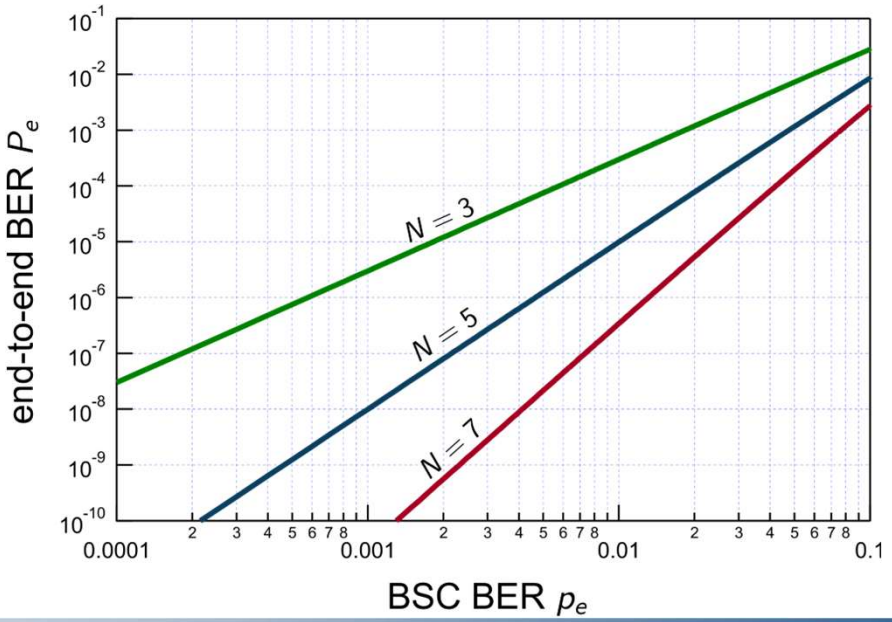
$$P_e = \sum_{n=\frac{N+1}{2}}^{N} \binom{N}{n} p_e^n \cdot (1-p_e)^{N-n} \approxeq \binom{N}{(N+1)/2} p_e^{(N+1)/2} \cdot (1-p_e)^{(N-1)/2}$$

Channel coding for dummies: Repetition coding (3/4)

What is the detection and correction performance of such approach?

$$P_e \approxeq \binom{N}{(N+1)/2} p_e^{(N+1)/2} \cdot (1-p_e)^{(N-1)/2}$$





Channel coding for dummies: Repetition coding (4/4)

What is the drawback? For each source bit, we are sending *N* coded bits!

We are increasing the amount of resources required for the transmission by a factor *N*

In channel coding, an important figure of merit is the coding rate *r*:

number of source bits at the input of the channel encoder

$$r = \frac{K}{N} \leq 1$$

number of coded bits at the output of the channel encoder

- the higher *r* (the closer to 1), the lower the amount of resources requested (i.e., the redundancy introduced) ☑
- the higher *r*, the lower the protection and the error detection/correction capabilities of the encoder ☒

For the repetition code, $r = 1/N$ (that decreases as *N* increases)

Parity check codes (1/3)

A way to improve the efficiency of the channel encoder (i.e., increasing r for a given performance P_e, or, equivalently, decreasing P_e for a given r) is through the use of 2D parity check codes, based on the XOR operation (row by row, and column by column):

A	B	$A \boxplus B$
0	0	0
0	1	1
1	0	1
1	1	0

coding rate

$$r = \frac{K}{N} = \frac{K}{(\sqrt{K}+1)^2} = \frac{16}{25}$$

Example: K=16

$$\mathbf{s} = [0\,1\,1\,1\,0\,0\,1\,0\,1\,1\,1\,0\,1\,0\,0\,1]$$

step 1: sort the bits into a $\sqrt{K} \times \sqrt{K}$ matrix

step 2: add parity bits

step 3: sort the matrix into a $(\sqrt{K}+1)^2$-element vector

0	1	1	1	1
0	0	1	0	1
1	1	1	0	1
1	0	0	1	0
0	0	1	0	1

$$\mathbf{c} = [0\,1\,1\,1\,0\,0\,1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,0\,0\,1\,0\,1\,0\,1\,1\,1]$$

Parity check codes (2/3)

How does error detection and correction work?

0	1	1	1	1
0	0	1	0	1
1	1	1	0	1
1	0	0	0	0
0	0	1	0	1

case 1: ☑

case 2: ☑

case 3: ☒

What is the detection and correction performance of the 2D parity check code?

$$P_e \approxeq \frac{K(\sqrt{K}-1)^2}{4} p_e^4$$

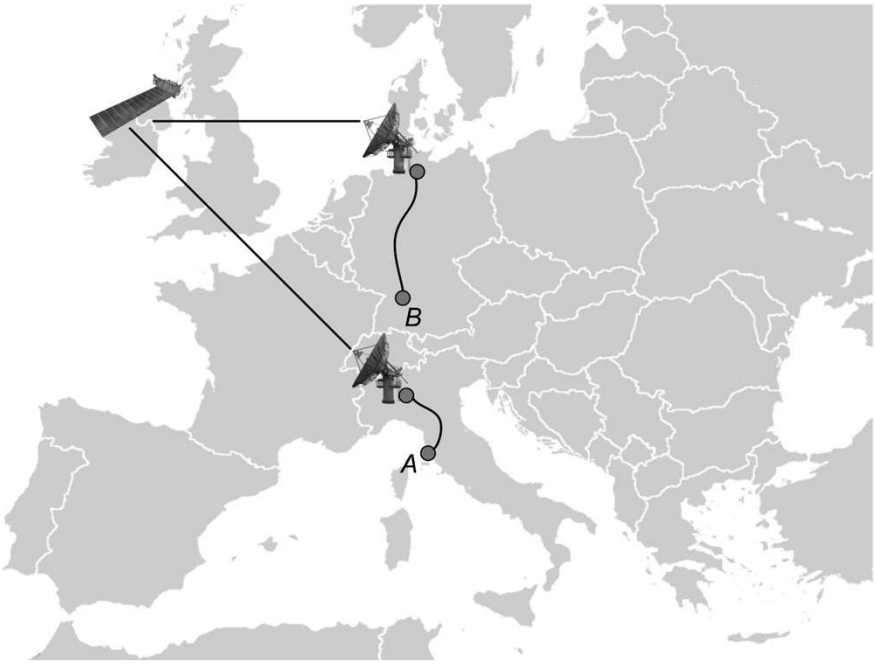
With K=16 and p_e=10^{-2}, P_e is comparable with a repetition code with N=7, but with a coding rate r = 16/25 ≈ 4.5·1/7

Parity check codes (3/3)

Parity check codes are used in both broadcast (e.g., satellite communications) and unicast (e.g., email) communications.

FEC is best suited for broadcast services, where a return channel is typically not available

ARQ mechanisms are better suited for unicast services





Linear block codes (1/2)

Linear block codes are a generalization of the parity check codes

Error detection and correction can be performed using a syndrome vector, which can be derived from the generation matrix, and requires linear operations

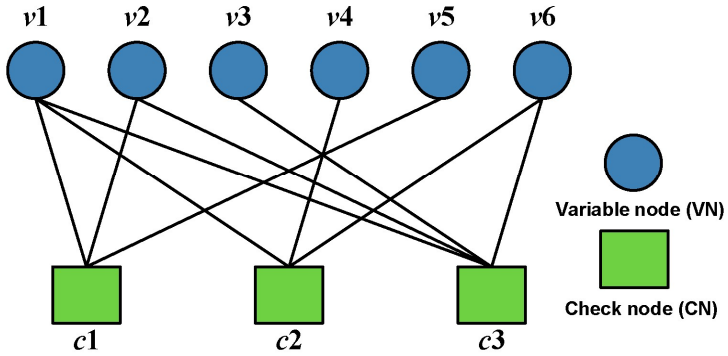
Notable examples are:

- Reed-Solomon (RS) codes, e.g. used in satellite digital video broadcasting (DVB-S) services
- low-density parity check (LDPC) codes



Linear block codes (2/2)

- LDPC codes are linear block codes characterized by sparse parity check matrices



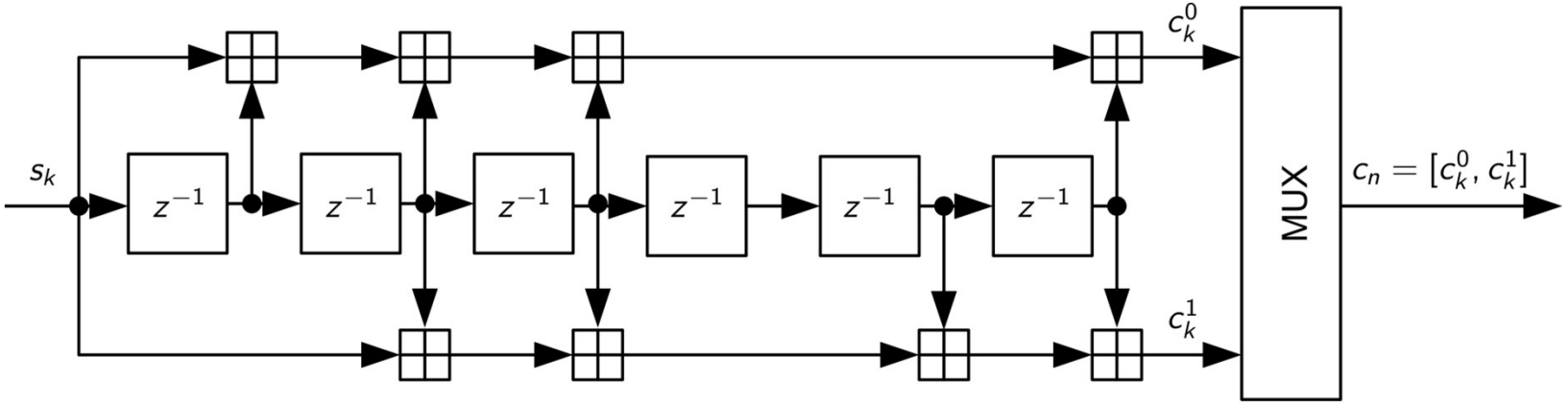
- Invented by Gallagher in 1960, but reproposed by MacKay and Neal in 1995, when technology of iterative decoding was mature enough
- Particularly performing with large block lengths (e.g., 64,800 bits)
- LDPC codes yield quasi-error-free (QEF) reception even in poor link conditions

Convolutional codes (1/4)

Convolutional codes introduce correlation between source bits, by using a feedforward architecture

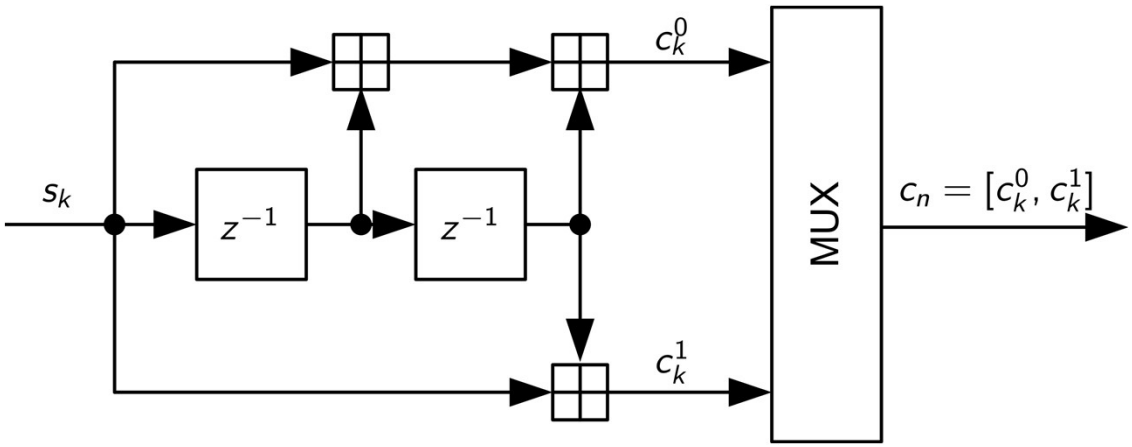
Notable examples:

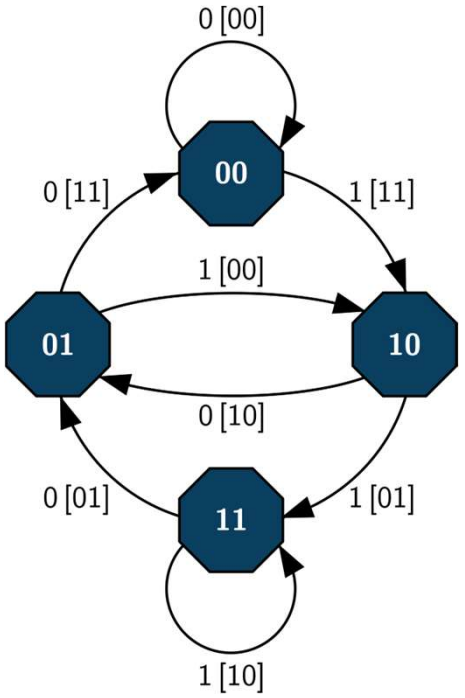
- convolutional code with rate r=1/2, used in WiFi, GPS/Galileo, and DVB-T
- Turbo codes



Convolutional codes (2/4)

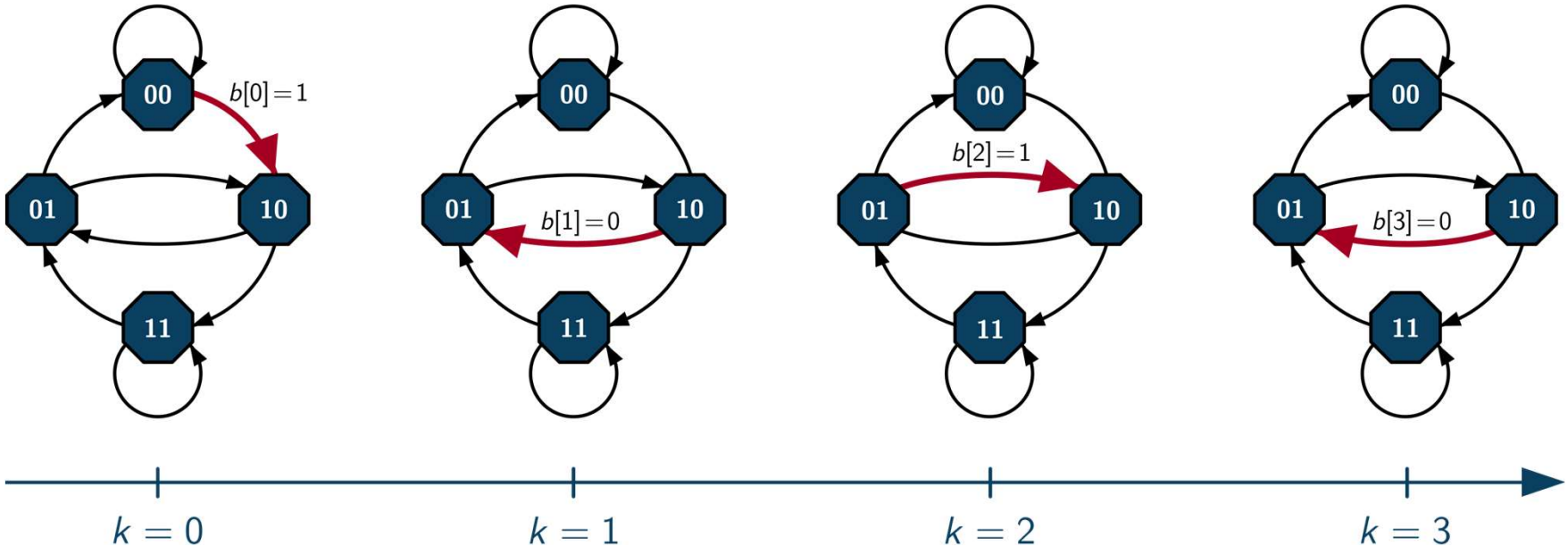
From the encoder structure, we can derive the corresponding finite-state machine:





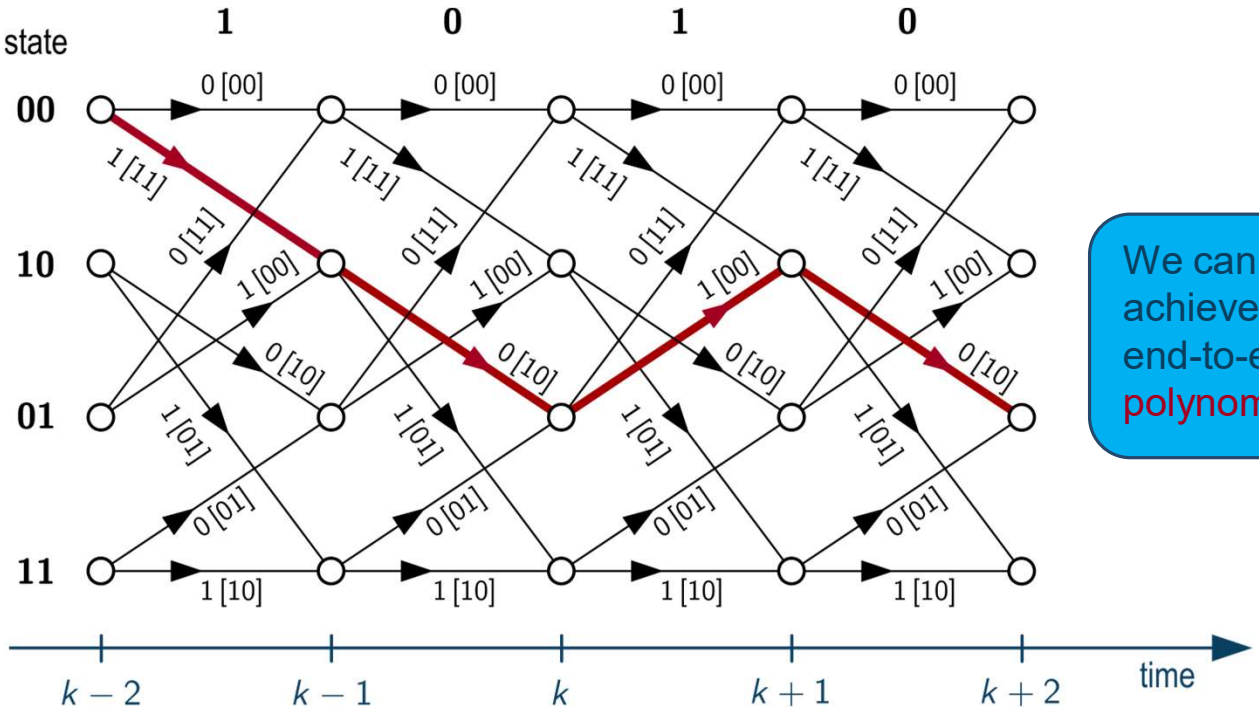
Convolutional codes (3/4)

Suppose the input bit sequence is **s** = [1 0 1 0]:



Convolutional codes (4/4)

A more convenient representation is the trellis diagram, which unfolds the state transitions through time:

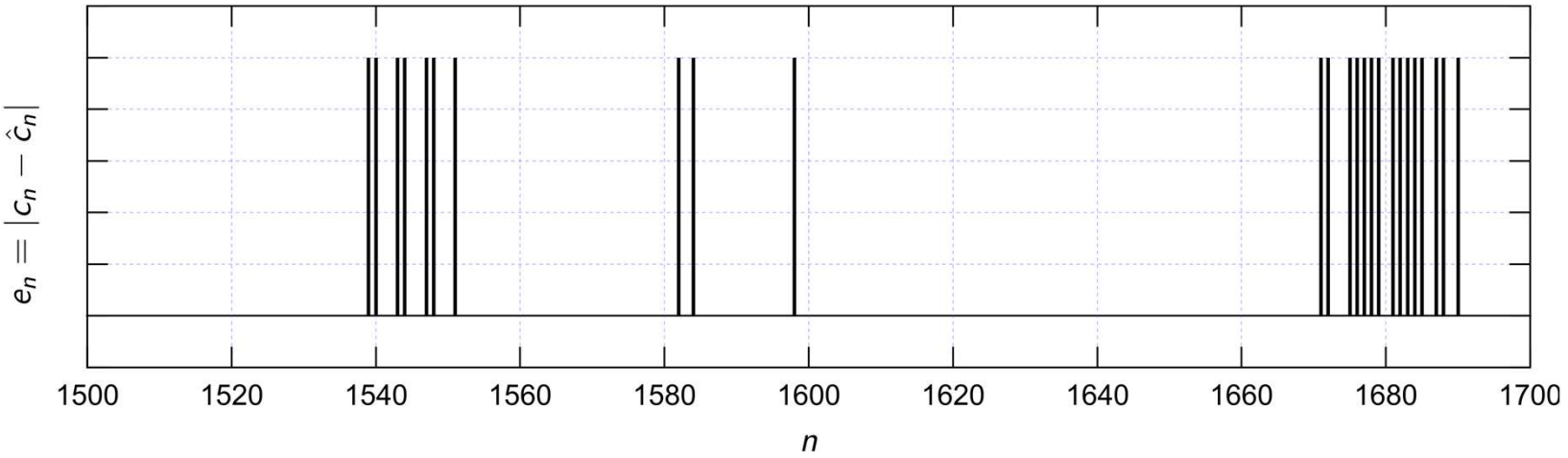


We can use the Viterbi algorithm to achieve exponentially decreasing end-to-end error probability P_e with polynomial-time decoding complexity



Interleaving (1/3)

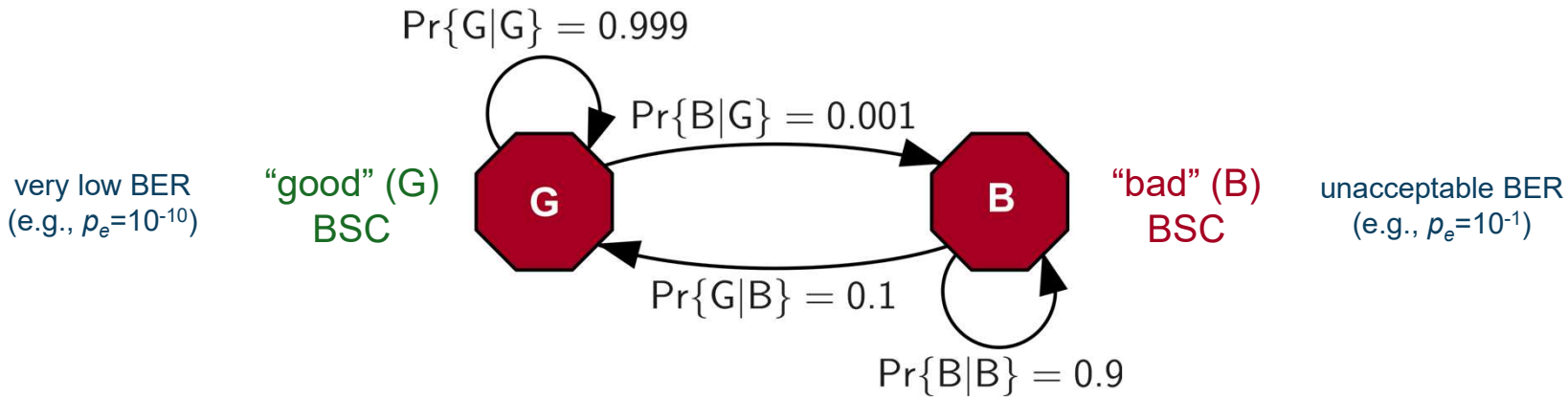
In many scenarios, the channel cannot be modeled as a BSC, due to the presence of error bursts (e.g., due to the sudden changes in the propagation environment)



Interleaving (2/3)

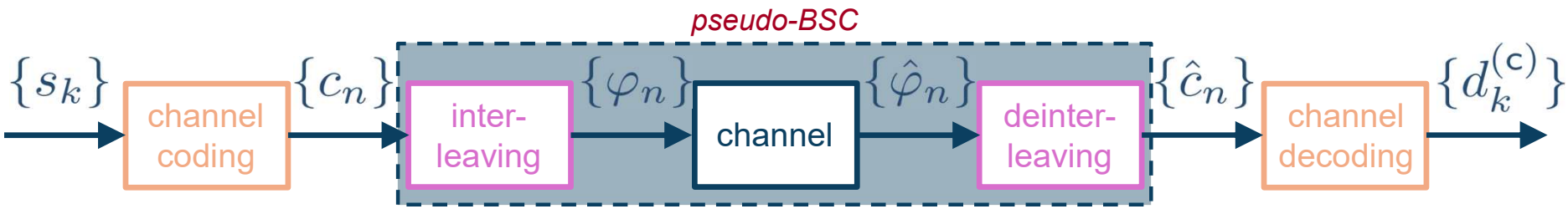
In many scenarios, the channel cannot be modeled as a BSC, due to the presence of error bursts (e.g., due to the sudden changes in the propagation environment)

This situation can be modeled as a two-state finite machine:



Interleaving (3/3)

To cope with error bursts, we can equip the channel encoder with an interleaver, which scrambles the outputs of the encoder so as to decorrelate the impact of the "bad" BSC:



Example: interleaving a block with length N=16

$$\mathbf{c} = [1011011100100110]$$

$$\boldsymbol{\varphi} = [1000010111111100]$$

step 1: sort **c** in a 4×4 matrix by rows

step 2: populate φ by reading the 4×4 matrix by columns

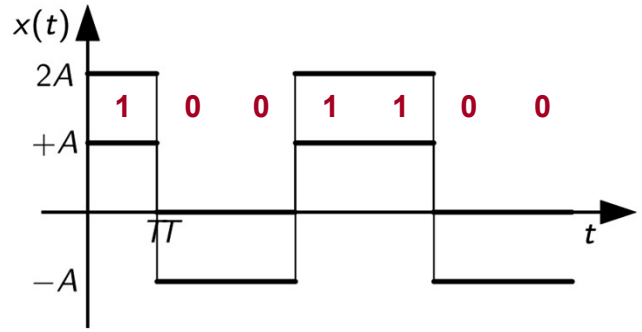
c_0=1	c_1=0	c_2=1	c_3=1
c_4=0	c_5=1	c_6=1	c_7=1
c_8=0	c_9=0	c_{10}=1	c_{11}=0
c_{12}=0	c_{13}=1	c_{14}=1	c_{15}=0

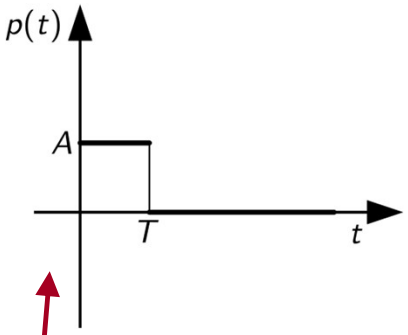


Modulation

Pulse shaping (1/3)

We need to adapt our sequence of bits into a signal which is suitable to be transmitted over a physical medium (air in wireless systems, a cable in wired systems)





symmetric NRZ signal: $x(t) = \sum_{k=-\infty}^{+\infty} (2b_k - 1)\, p(t - kT)$

shaping pulse

For convenience, let us use a symmetric NRZ modulation

Pulse shaping (2/3)

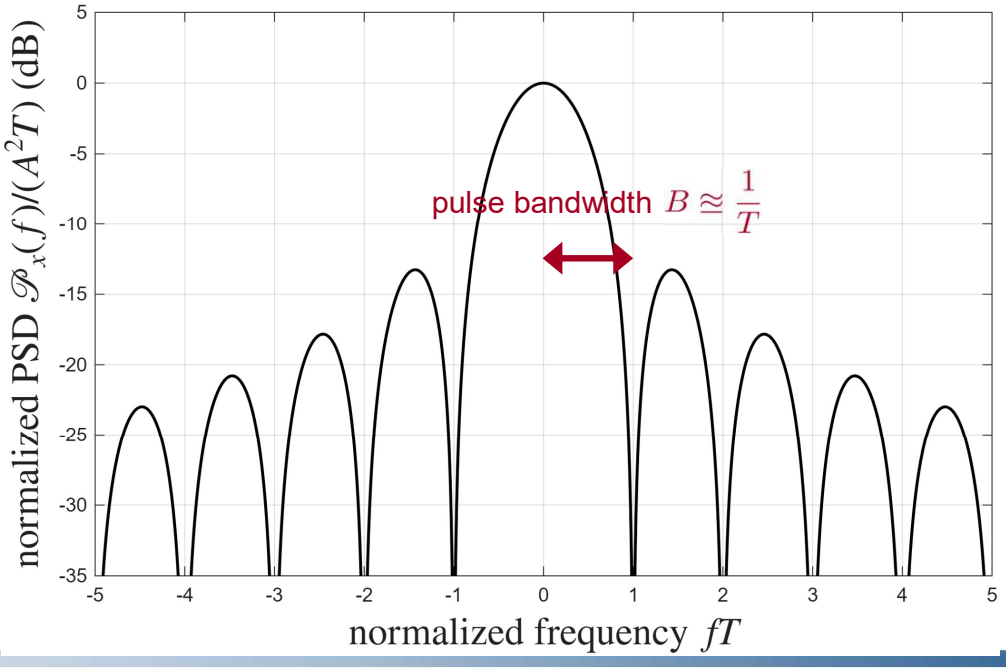
What is the (physical) bandwidth of a symmetric NRZ signal?

Let us consider the power spectral density (PSD) of $x(t)$:

$$\mathcal{P}_x(f) = \frac{A^2}{T}|P(f)|^2 = A^2\,T\,\mathrm{sinc}^2(fT)$$

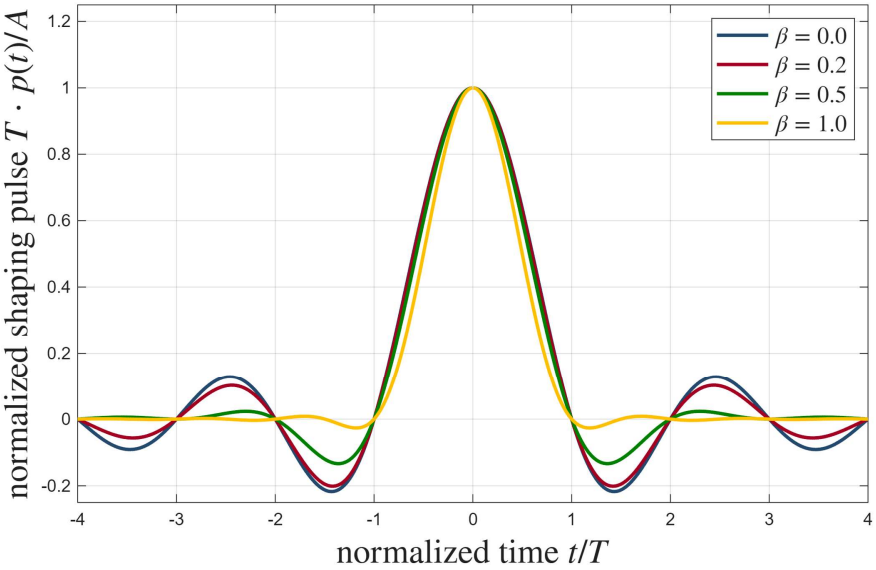
$$B \approx \frac{1}{T} = R = R_b$$

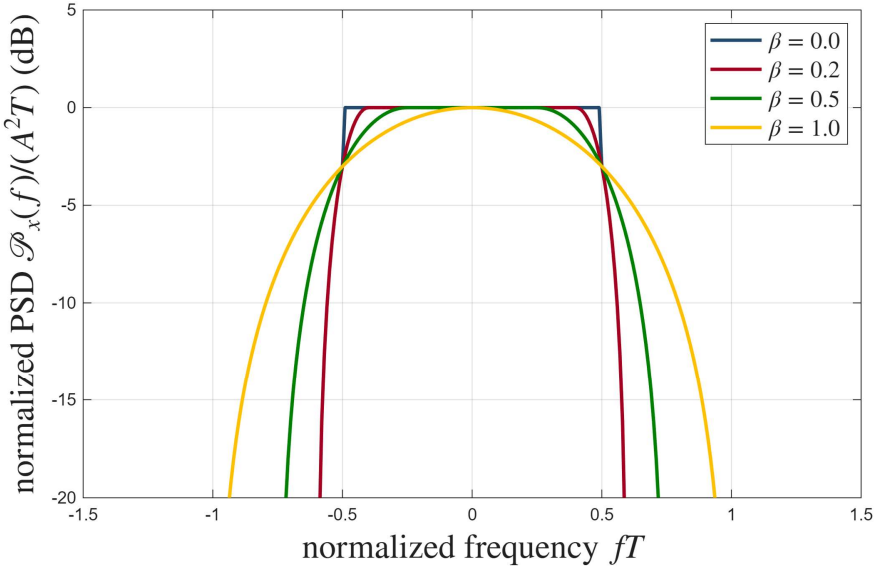
pulse bandwidth — symbol rate — bitrate



Pulse shaping (3/3)

Similar considerations can be drawn using other pulse formats, e.g, the square-root raised cosine (SRRC) with rolloff factor β:



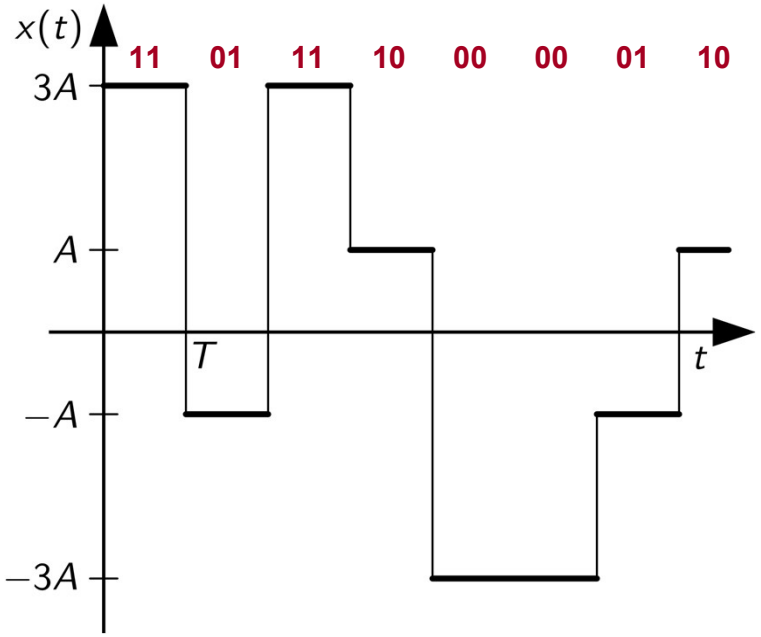


$$B \propto \frac{1}{T} = R = \cancel{R_b}$$

this is NOT true in generale!

Multilevel (baseband) modulations (1/2)

To improve the spectral efficiency R_b/B, we can map more bits into a single symbol:



the bitrate is now twice the symbol rate:

$$R_b = 2R = 2 \cdot \frac{1}{T} \approx 2B$$

$$T_b = \frac{1}{R_b} = \frac{T}{2}$$



Multilevel (baseband) modulations (2/2)

In general, when using an *M*-level mapping (i.e., when using $N_b = \log_2 M$ bits per symbol), we have

$$R_b = \log_2 M \cdot R = N_b \cdot R \approxeq N_b \cdot B$$

The spectral efficiency becomes

$$\eta_s \triangleq \frac{R_b}{B} \approxeq N_b \qquad [\text{b/s/Hz}]$$

This is particularly relevant in wireless communications, where the frequency spectrum is scarce!



Carrier modulation (1/4)

Why do we need to modulate?

- baseband is not sustainable, in terms of physical feasibility of the components: e.g., the antenna size needs to have the same order of magnitude of the signal wavelength λ (typical size: $\lambda/2$)



- bandpass yields additional benefits, e.g., in terms of multiple access

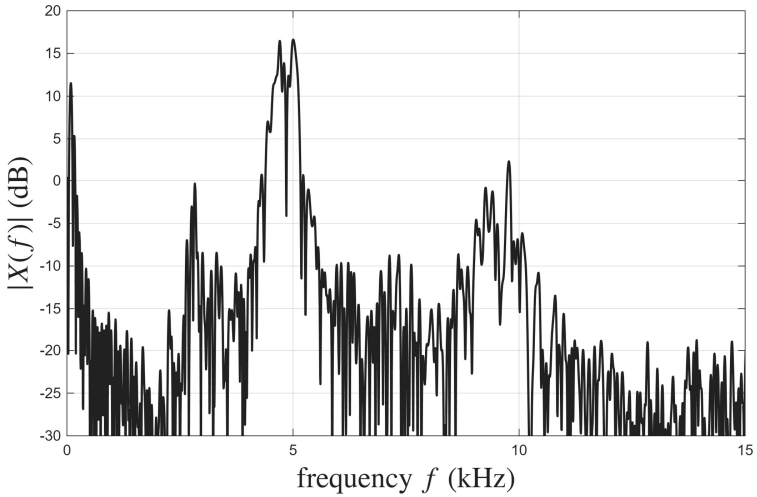
Carrier modulation (2/4)

Signal wavelength:

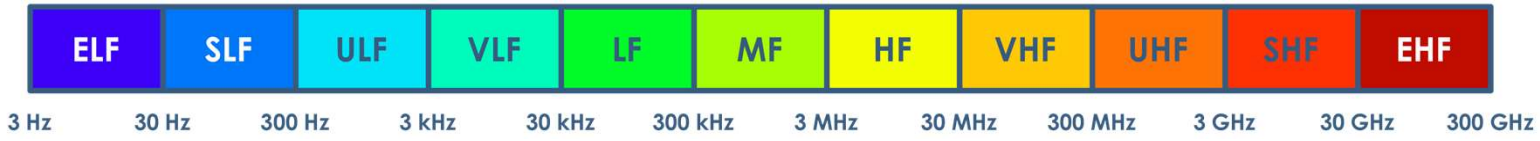
$$\lambda = \frac{c}{f}$$

speed of light, $3 \cdot 10^8$ m/s

frequency

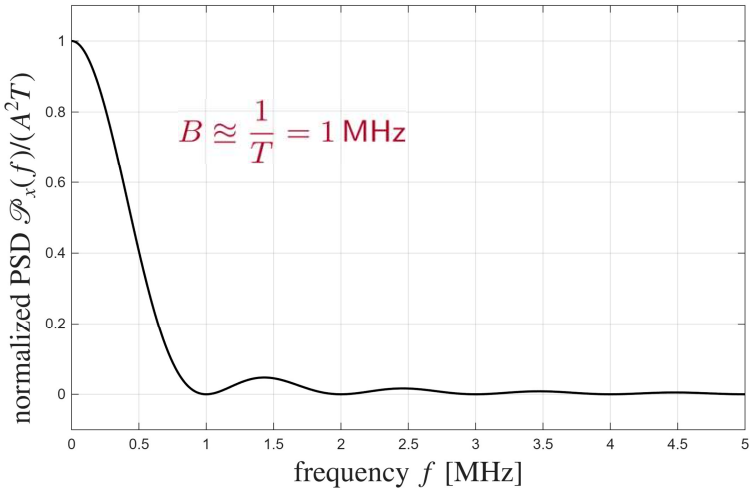


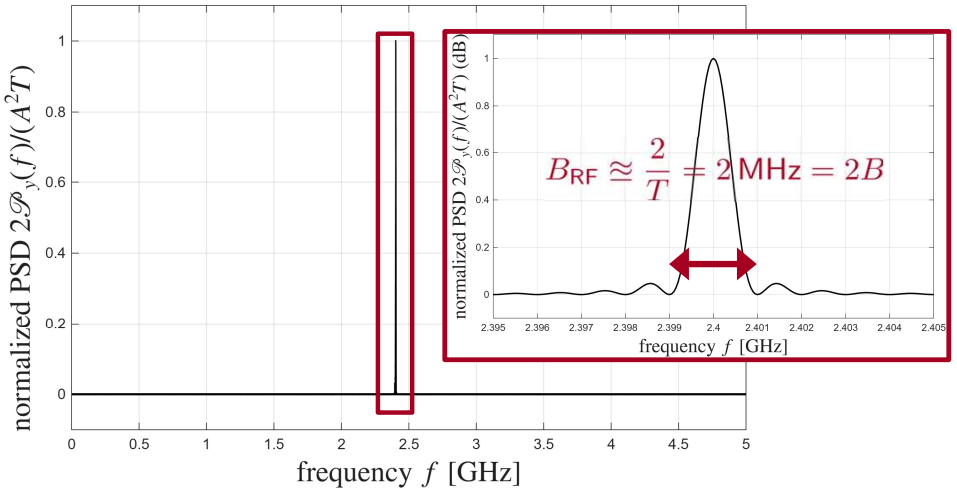
Wireless radio spectrum:



Carrier modulation (3/4)

$$x(t) = \sum_{k=-\infty}^{+\infty} (2b_k - 1)\, p(t - kT), \quad T = 1\,\mu\text{s} \qquad y(t) = x(t)\cos(2\pi f_0 t), \quad f_0 = 2.4\,\text{GHz}$$

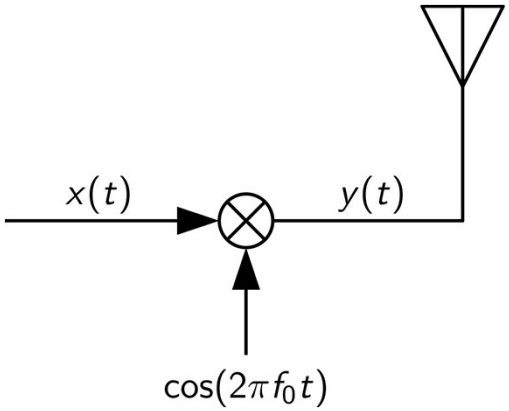


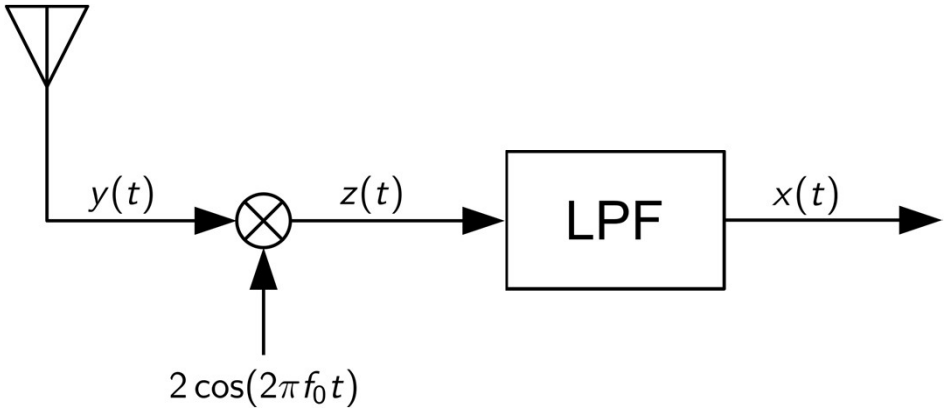


Modulation theorem: $x(t)\cos(2\pi f_0 t) \Leftrightarrow \dfrac{X(f - f_0) + X(f + f_0)}{2}$

Carrier modulation (4/4)

How can we demodulate the signal $x(t)$?





$$y(t) = x(t)\cos(2\pi f_0 t)$$

$$2\cos^2(\alpha) = 1 + \cos(2\alpha)$$

$$z(t) = y(t)2\cos(2\pi f_0 t) = 2x(t)\cos^2(2\pi f_0 t)$$

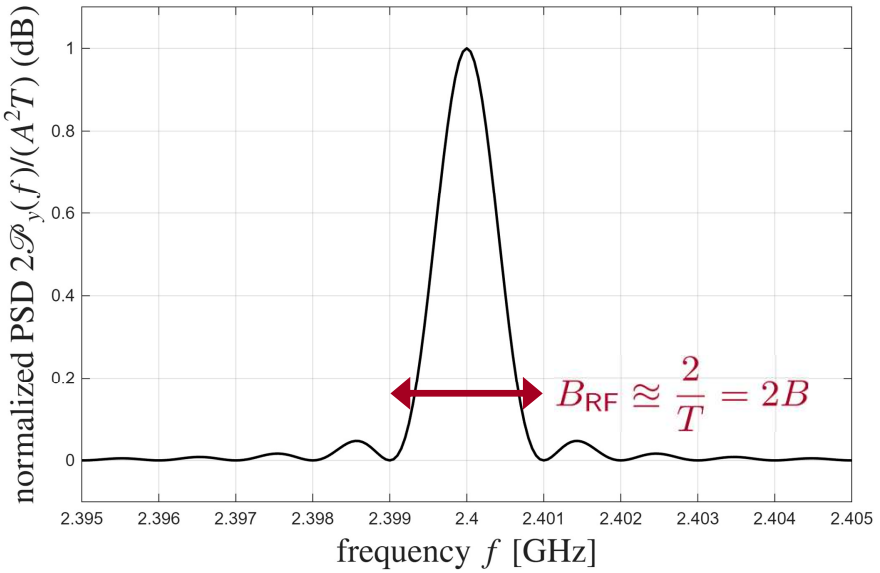
$$= x(t) + x(t)\cos\left[2\pi(2f_0)t\right]$$

I/Q modulation (1/4)

Modulation is the key to transmit a signal via radio frequency (RF) but… when moving to the RF, we decrease our spectral efficiency by a factor 2:

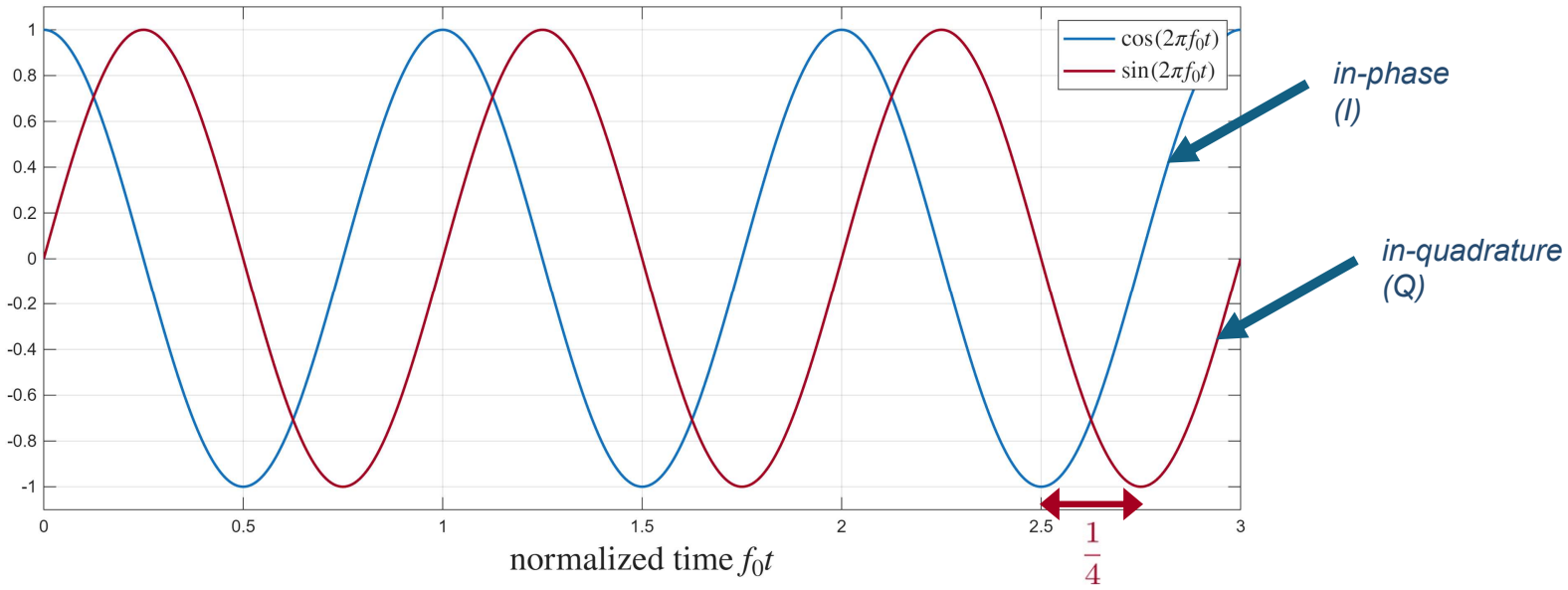
$$\eta_{\mathsf{s}} \triangleq \frac{R_b}{B_{\mathsf{RF}}} \approx \frac{N_b}{2} \qquad [\mathrm{b/s/Hz}]$$

How can we improve η_{s}, while exploiting the intuition of multiple bits per symbol?



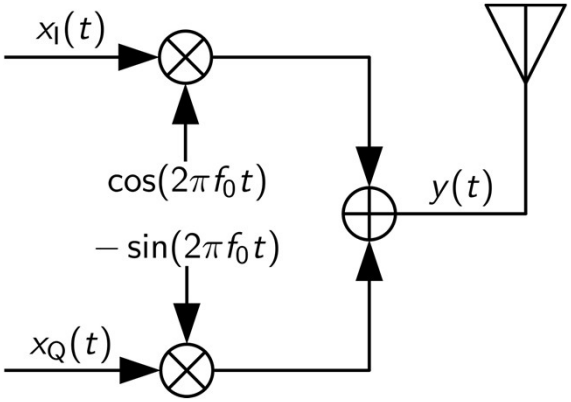
I/Q modulation (2/4)

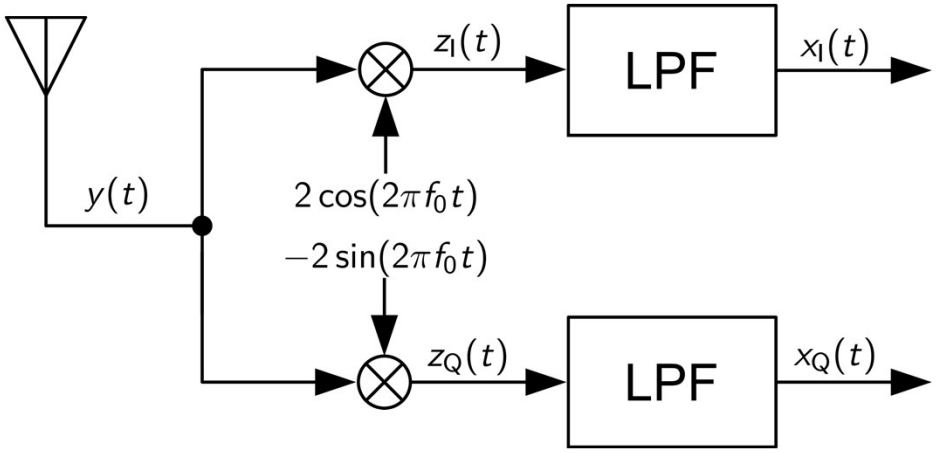
Let us multiplex two streams on the same carrier, by leveraging two orthogonal waveforms:



I/Q modulation (3/4)

Let us feed the modulator with two independent streams, each with bitrate R_b:





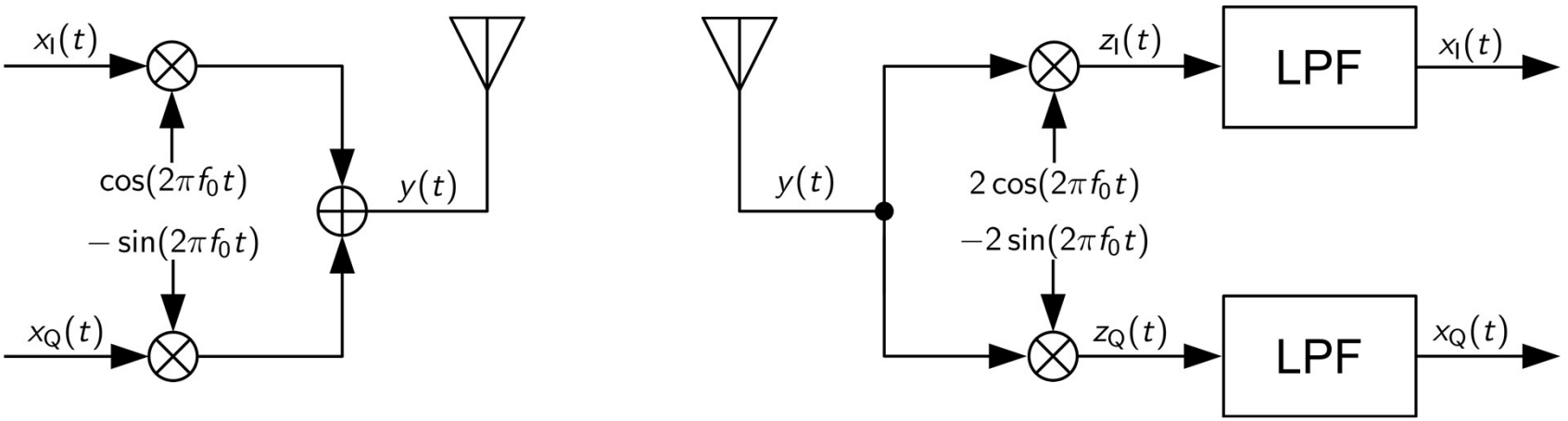
$$y(t) = x_I(t)\cos(2\pi f_0 t) - x_Q(t)\sin(2\pi f_0 t)$$

$$2\cos^2(\alpha) = 1 + \cos(2\alpha)$$
$$2\sin^2(\alpha) = 1 - \cos(2\alpha)$$
$$2\sin(\alpha)\cos(\alpha) = \sin(2\alpha)$$

$$z_I(t) = y(t)2\cos(2\pi f_0 t) = 2x_I(t)\cos^2(2\pi f_0 t) - 2x_Q(t)\sin(2\pi f_0 t)\cos(2\pi f_0 t)$$
$$= x_I(t) + x_I(t)\cos[2\pi(2f_0)t] - x_Q(t)\sin[2\pi(2f_0)t]$$
$$z_Q(t) = -y(t)2\sin(2\pi f_0 t) = 2x_Q(t)\sin^2(2\pi f_0 t) - 2x_I(t)\sin(2\pi f_0 t)\cos(2\pi f_0 t)$$
$$= x_Q(t) - x_Q(t)\cos[2\pi(2f_0)t] - x_I(t)\sin[2\pi(2f_0)t]$$

I/Q modulation (4/4)

Let us feed the modulator with two independent streams, each with bitrate R_b:



$$\eta_s \triangleq \frac{R_{b,I} + R_{b,Q}}{B_{RF}} = \frac{2R_b}{2B} \approx N_b \qquad [b/s/Hz]$$

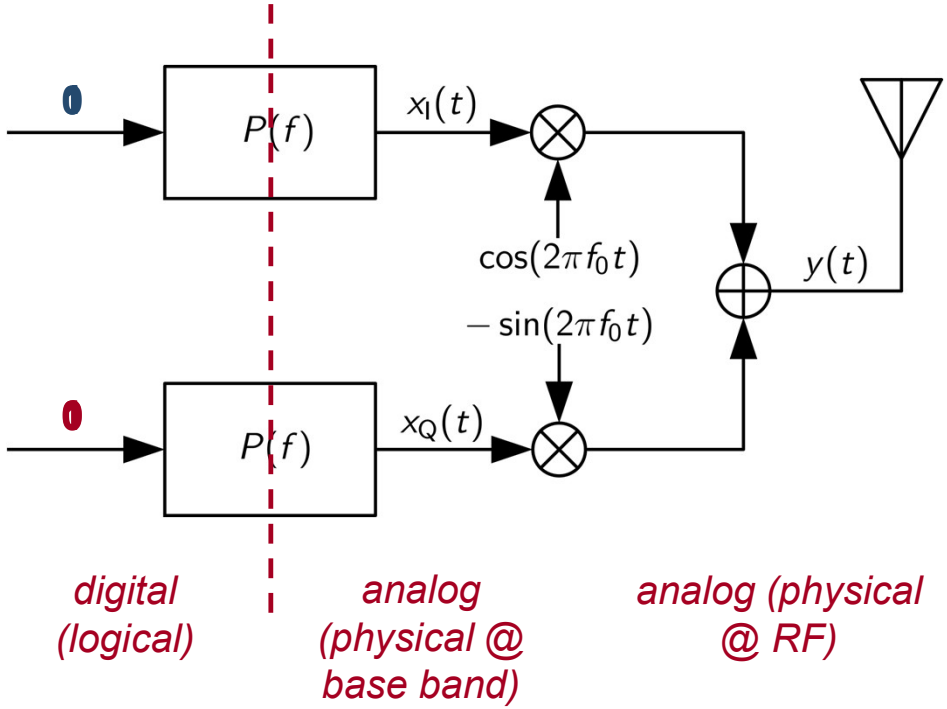
PSK and QAM constellations (1/5)

To cast one single binary stream into the I/Q scheme, for instance we can assign odd bits to the I branch and even bits to the Q branch:

01101110101100…

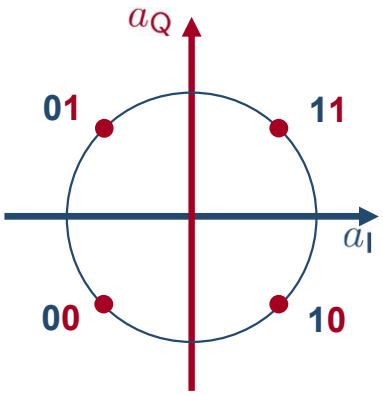
$$x_{\mathrm{I}}(t) = \sum_{\ell=-\infty}^{+\infty} (2b_{2\ell} - 1)\, p(t - \ell T)$$

$$x_{\mathrm{Q}}(t) = \sum_{\ell=-\infty}^{+\infty} (2b_{2\ell+1} - 1)\, p(t - \ell T)$$

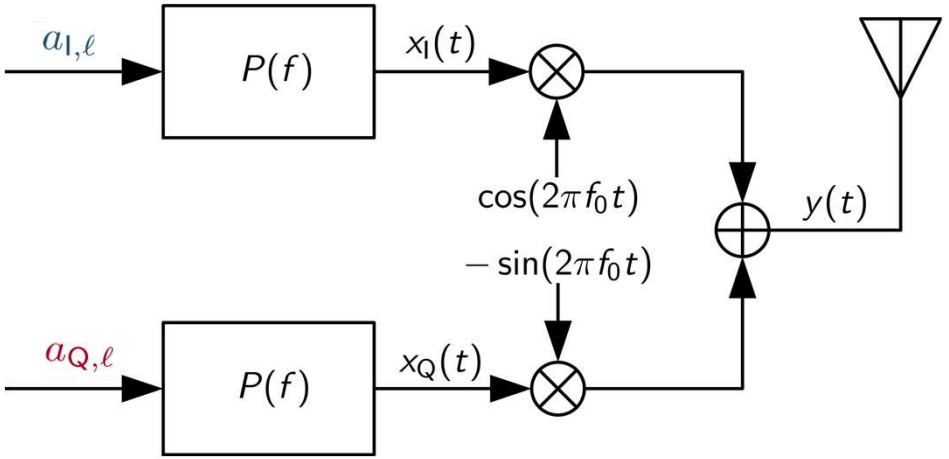


PSK and QAM constellations (2/5)

This is equivalent to map the input bits in complex symbols over the I/Q plane:



$b_{2\ell}$ $b_{2\ell+1}$	$a_{I,\ell}$	$a_{Q,\ell}$
00	-1	-1
01	-1	+1
10	+1	-1
11	+1	+1

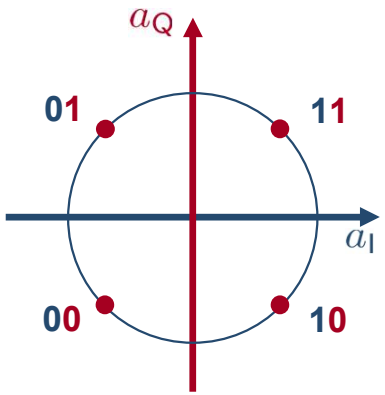


$$x_I(t) = \sum_{\ell=-\infty}^{+\infty} (2b_{2\ell} - 1)\, p(t - \ell T) = \sum_{\ell=-\infty}^{+\infty} a_{I,\ell}\, p(t - \ell T)$$
$$x_Q(t) = \sum_{\ell=-\infty}^{+\infty} (2b_{2\ell+1} - 1)\, p(t - \ell T) = \sum_{\ell=-\infty}^{+\infty} a_{Q,\ell}\, p(t - \ell T)$$

With this scheme, we can send N_b=$\log_2 4$=2 bits per symbol, thus achieving η_s=2 b/s/Hz

PSK and QAM constellations (3/5)

This scheme goes under the name of quaternary phase shift keying (QPSK):



$b_{2\ell}\, b_{2\ell+1}$	$a_{I,\ell}$	$a_{Q,\ell}$
00	-1	-1
01	-1	+1
10	+1	-1
11	+1	+1

Question: Can we generalize to an arbitrary number of bits (including odd N_b's)?

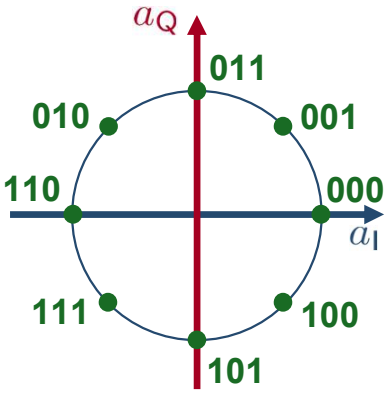
Answer: Yes, using phase shift keying (PSK) and quadrature amplitude modulation (QAM) constellations!

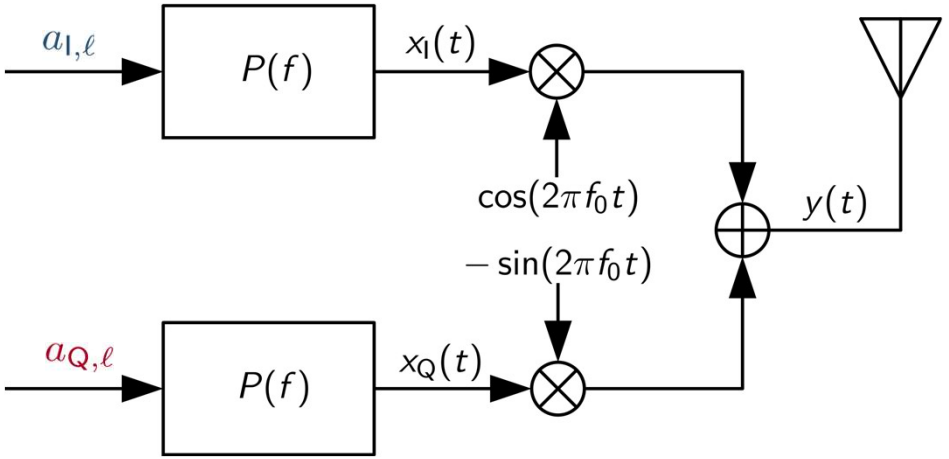
PSK and QAM constellations (4/5)

Important: The I/Q modulator block diagram remains the same!

M=8 (N_b=3)

→ 8-PSK:





$b_{3\ell}b_{3\ell+1}b_{3\ell+2}$	$a_{I,\ell}$	$a_{Q,\ell}$	$b_{3\ell}b_{3\ell+1}b_{3\ell+2}$	$a_{I,\ell}$	$a_{Q,\ell}$
000	$+1$	0	100	$+1/\sqrt{2}$	$-1/\sqrt{2}$
001	$+1/\sqrt{2}$	$+1/\sqrt{2}$	101	0	-1
010	$-1/\sqrt{2}$	$+1/\sqrt{2}$	110	-1	0
011	0	$+1$	111	$-1/\sqrt{2}$	$-1/\sqrt{2}$

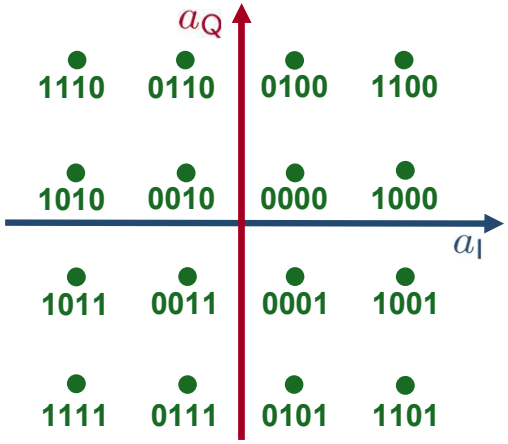
$$x_I(t) = \sum_{\ell=-\infty}^{+\infty} a_{I,\ell}\, p(t-\ell T)$$
$$x_Q(t) = \sum_{\ell=-\infty}^{+\infty} a_{Q,\ell}\, p(t-\ell T)$$

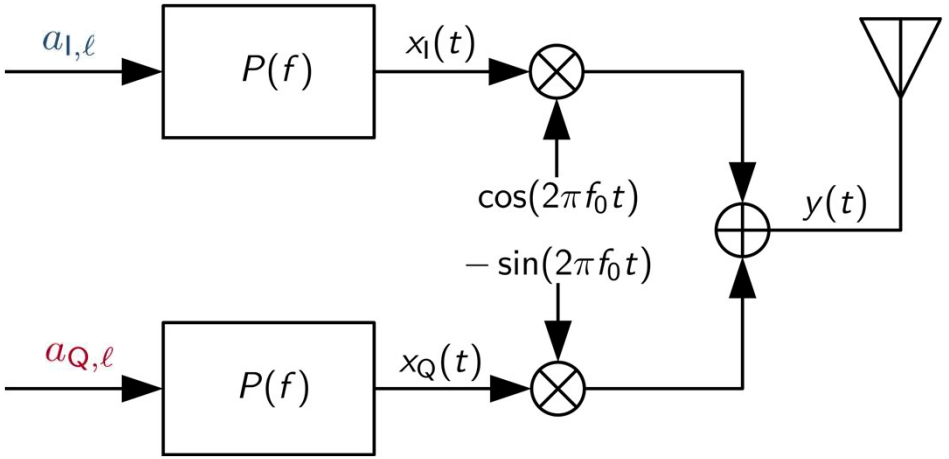
PSK and QAM constellations (5/5)

Important: The I/Q modulator block diagram remains the same!

M=16 (N_b=4)

→ 16-QAM:

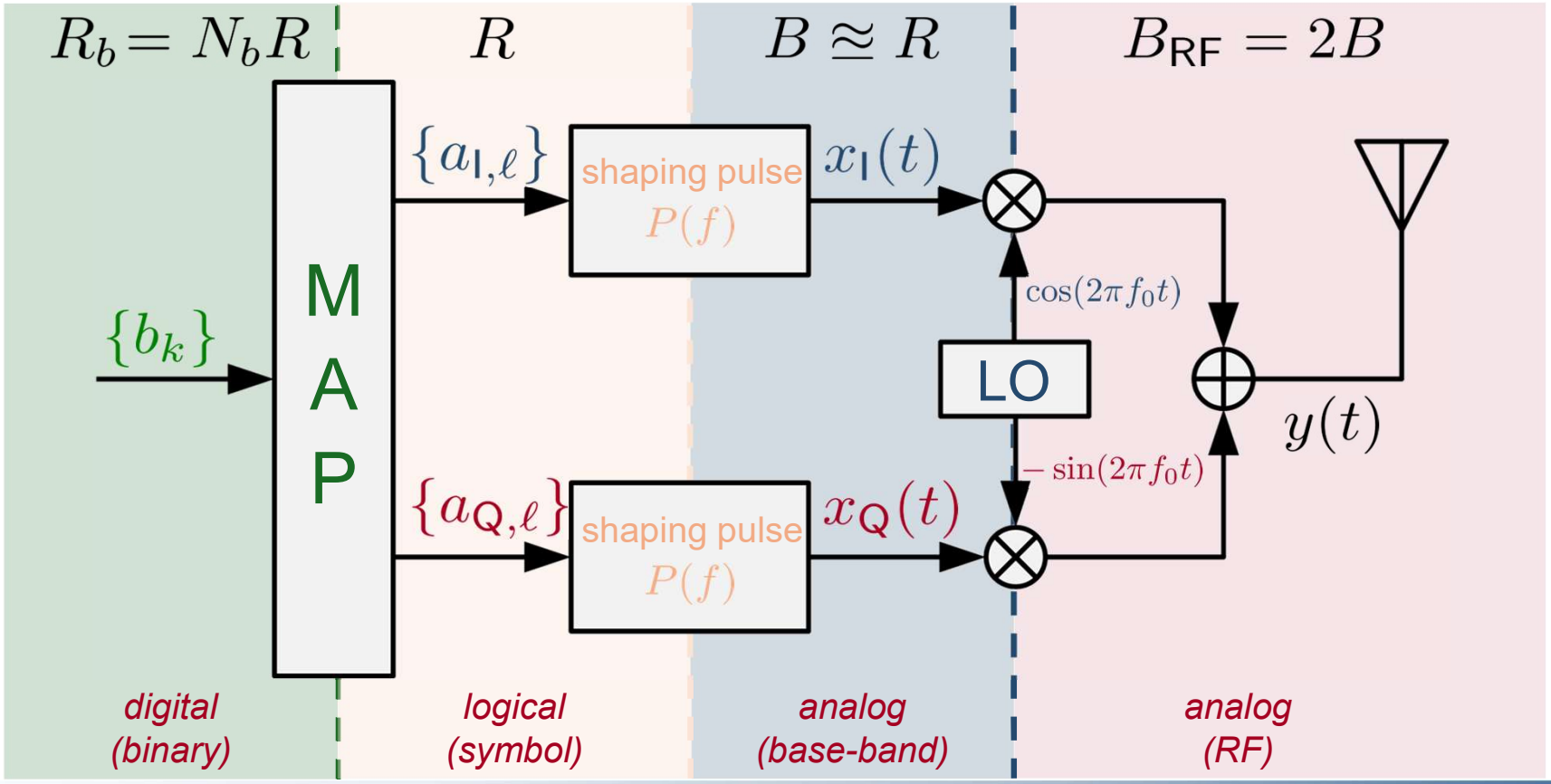




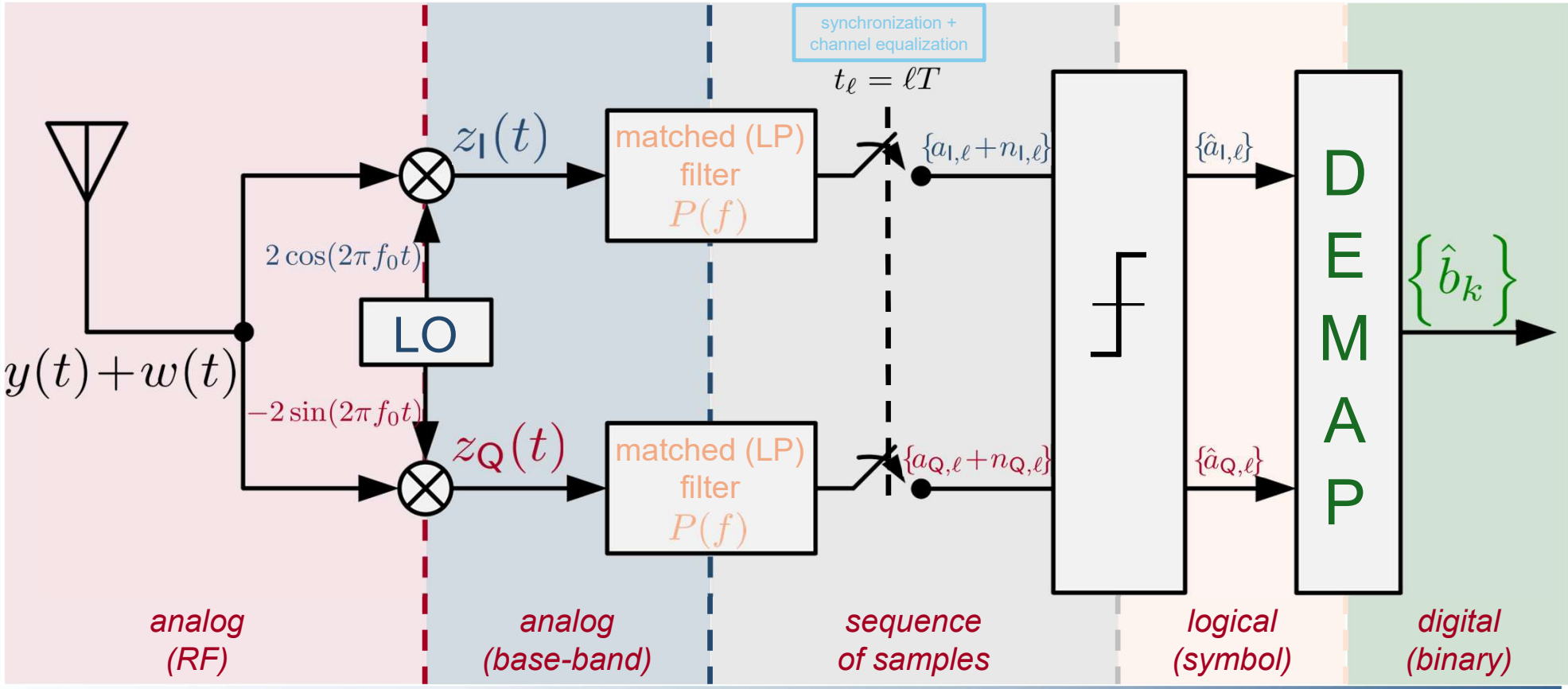
$$x_I(t) = \sum_{\ell=-\infty}^{+\infty} a_{I,\ell}\, p(t - \ell T)$$
$$x_Q(t) = \sum_{\ell=-\infty}^{+\infty} a_{Q,\ell}\, p(t - \ell T)$$

Exercise: Derive the mapping for the binary PSK (BPSK)

The I-Q modulator

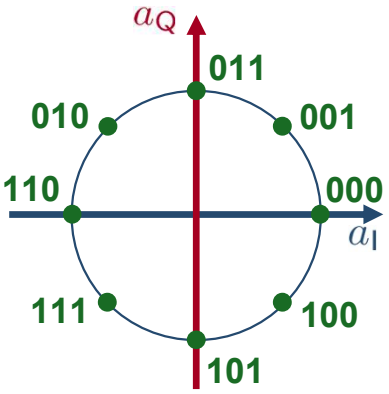


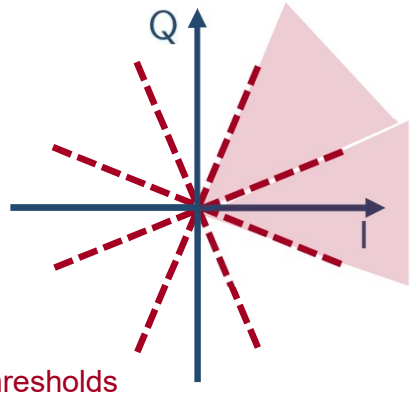
The I-Q demodulator (1/4)



The I-Q demodulator (2/4)

Example of threshold-based detection: let us suppose to use 8-PSK





decision thresholds

any sample with coordinates $(a_{I,\ell}+n_{I,\ell}, a_{Q,\ell}+n_{Q,\ell})$ falling in this region gives the symbol $\hat{a}_{I,\ell} = 1/\sqrt{2}, \hat{a}_{Q,\ell} = 1/\sqrt{2}$

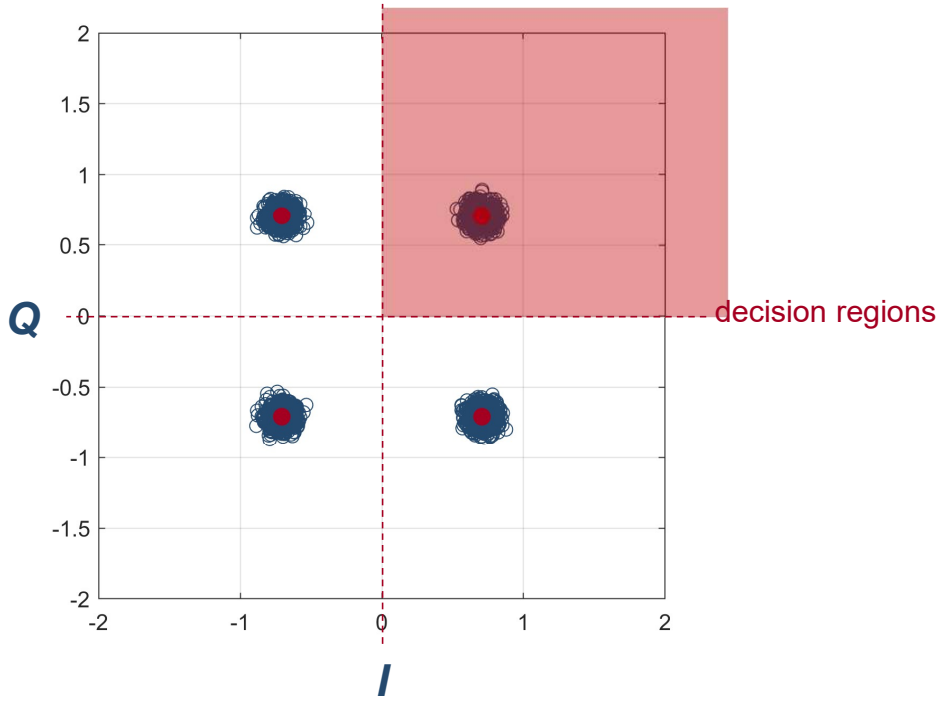
after de-mapping, we have **001**

any sample with coordinates $(a_{I,\ell}+n_{I,\ell}, a_{Q,\ell}+n_{Q,\ell})$ falling in this region gives the symbol $\hat{a}_{I,\ell} = 1, \hat{a}_{Q,\ell} = 0$

after de-mapping, we have **000**

The I-Q demodulator (3/4)

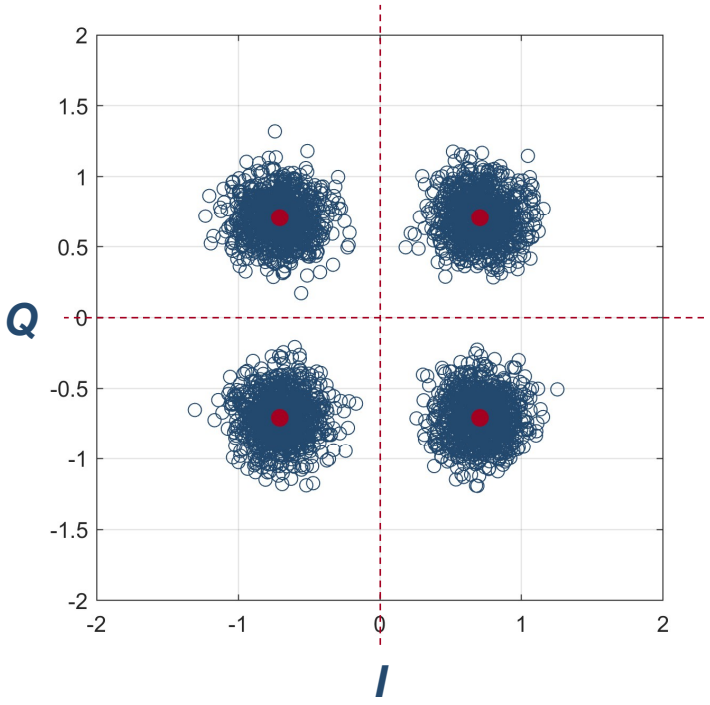
What happens when using a QPSK modulation in a high signal-to-noise ratio (SNR) scenario?



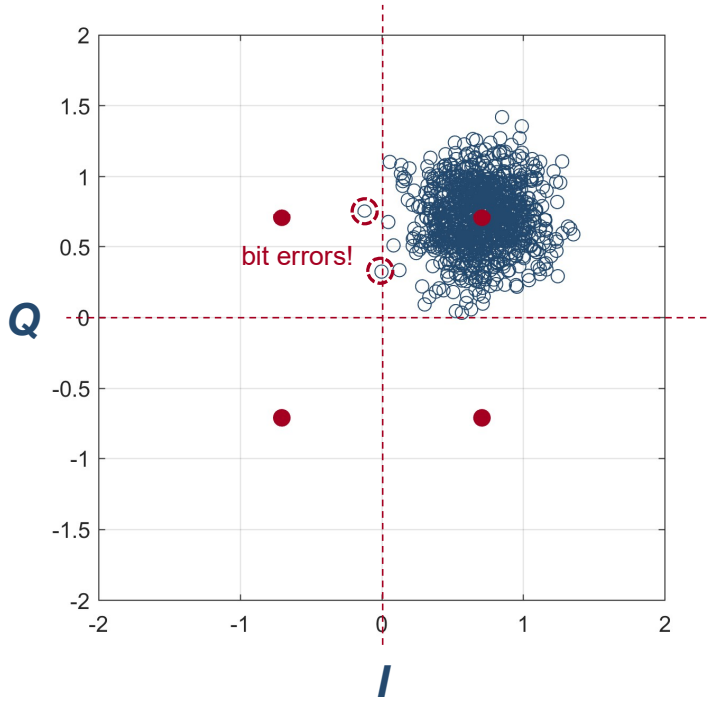


The I-Q demodulator (4/4)

medium-SNR scenario



low-SNR scenario



Impact of additive white Gaussian noise (AWGN) (1/5)

How can we quantify the impact of AWGN in the demodulation performance?

The main figure of merit becomes the signal-to-noise ratio (SNR), in the form of

$$\frac{E_b}{N_0} = \frac{S}{N} \cdot \frac{B}{R_b}$$

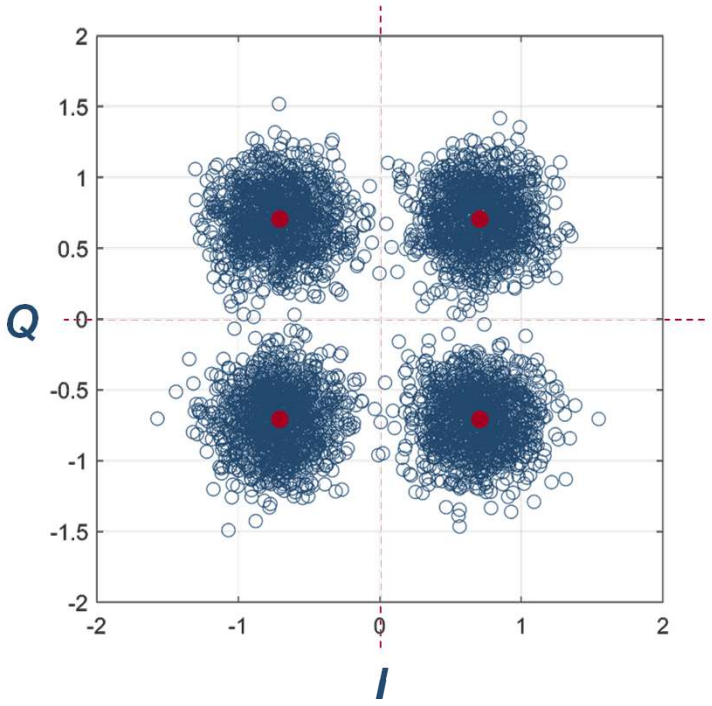
- E_b: energy per bit
- N_0: noise power spectral density
- S: useful signal power
- N: noise power
- B: signal bandwidth
- R_b: bit rate

The performance of a digital communication system can be assessed measuring the bit error rate (BER) – which occurs when $\hat{b}_k \neq b_k$ – and the symbol error rate (SER) – which occurs when $(\hat{a}_{\mathrm{I},\ell} + j\hat{a}_{\mathrm{Q},\ell}) \neq (a_{\mathrm{I},\ell} + ja_{\mathrm{Q},\ell})$

Impact of additive white Gaussian noise (AWGN) (2/5)

QPSK @

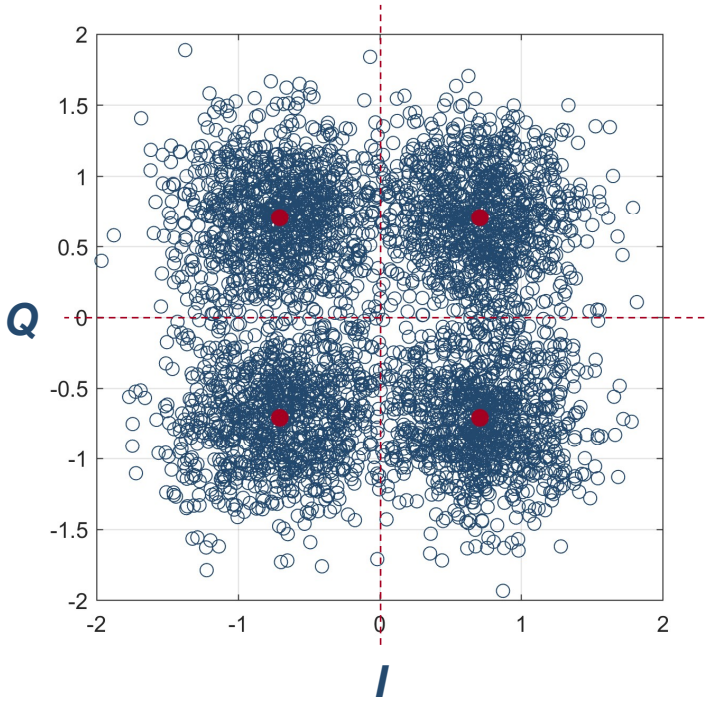
$E_b/N_0 = 10$ dB



Impact of additive white Gaussian noise (AWGN) (3/5)

The situation becomes dramatic when either increasing the AWGN power…

QPSK @
$E_b/N_0 = 3\,\mathrm{dB}$

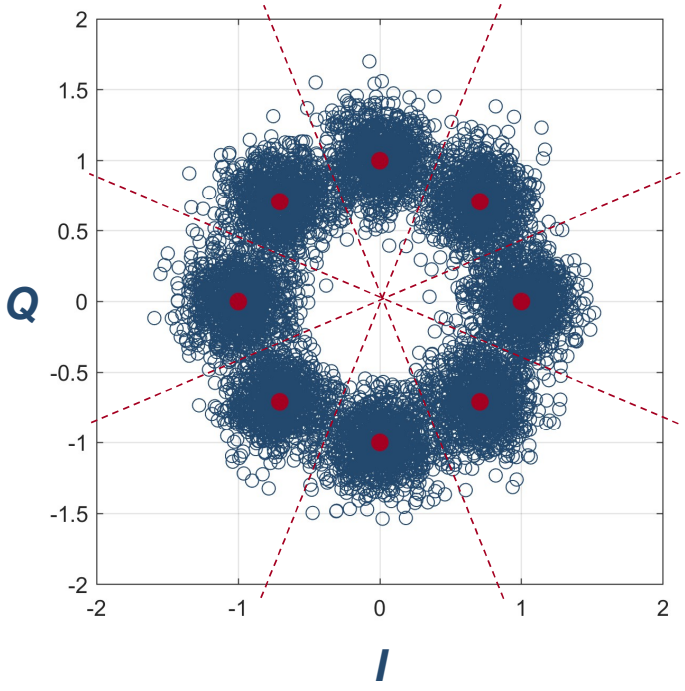


Impact of additive white Gaussian noise (AWGN) (4/5)

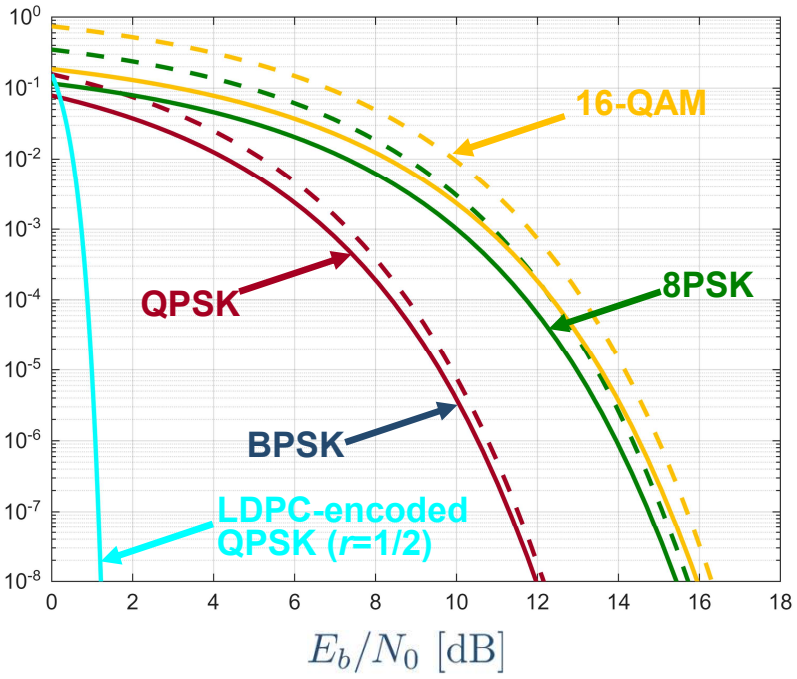
…or when increasing the modulation order

8PSK @

$E_b/N_0 = 7\,\mathrm{dB}$



Impact of additive white Gaussian noise (AWGN) (5/5)



—— *BER* p_e

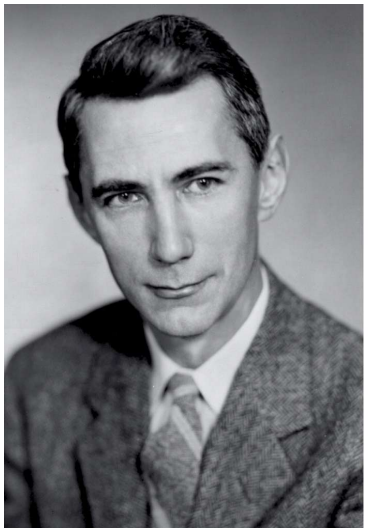
-------- *SER* $\approx N_b p_e$

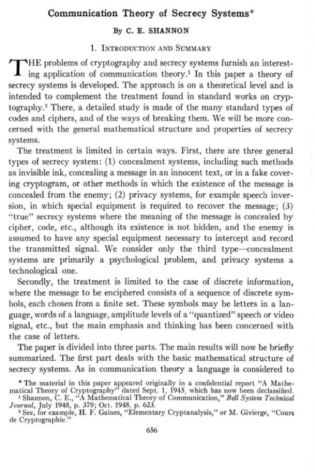
Shannon capacity (1/3)

How can we evaluate the performance of a system?

$$R_b \leq \mathcal{C} = B \cdot \log_2\left(1 + \frac{S}{N}\right) = B \cdot \log_2\left(1 + \frac{E_b}{N_0} \cdot \frac{R_b}{B}\right)$$

Shannon capacity





Communication Theory of Secrecy Systems*

By C. E. SHANNON

1. Introduction and Summary

THE problems of cryptography and secrecy systems furnish an interesting application of communication theory.[1] In this paper a theory of secrecy systems is developed. The approach is on a theoretical level and is intended to complement the treatment found in standard works on cryptography.[2] There, a detailed study is made of the many standard types of codes and ciphers, and of the ways of breaking them. We will be more concerned with the general mathematical structure and properties of secrecy systems.

The treatment is limited in certain ways. First, there are three general types of secrecy system: (1) concealment systems, including such methods as invisible ink, concealing a message in an innocent text, or in a fake covering cryptogram, or other methods in which the existence of the message is concealed from the enemy; (2) privacy systems, for example speech inversion, in which special equipment is required to recover the message; (3) "true" secrecy systems where the meaning of the message is concealed by cipher, code, etc., although its existence is not hidden, and the enemy is assumed to have any special equipment necessary to intercept and record the transmitted signal. We consider only the third type—concealment systems are primarily a psychological problem, and privacy systems a technological one.

Secondly, the treatment is limited to the case of discrete information, where the message to be enciphered consists of a sequence of discrete symbols, each chosen from a finite set. These symbols may be letters in a language, words of a language, amplitude levels of a "quantized" speech or video signal, etc., but the main emphasis and thinking has been concerned with the case of letters.

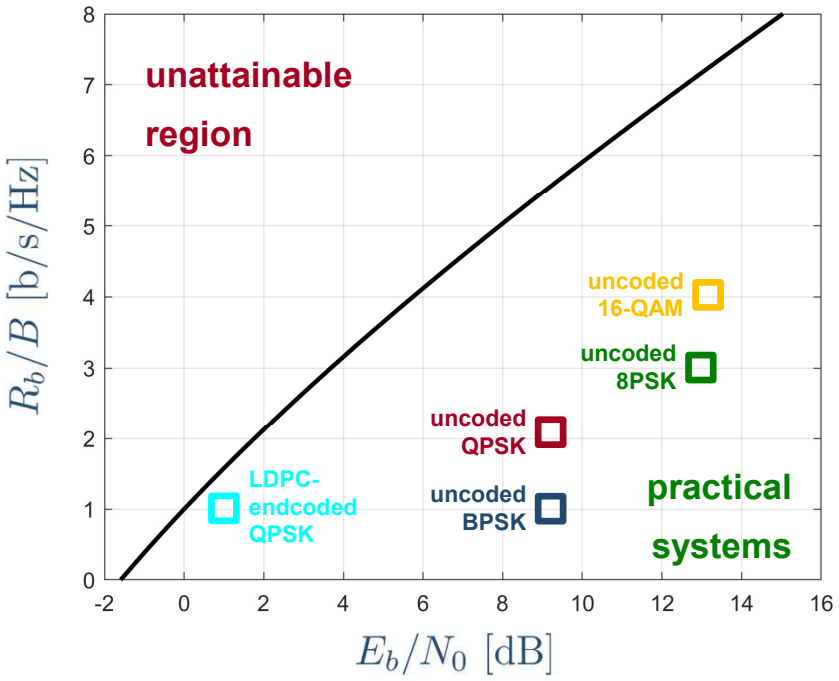
The paper is divided into three parts. The main results will now be briefly summarized. The first part deals with the basic mathematical structure of secrecy systems. As in communication theory a language is considered to

* The material in this paper appeared originally in a confidential report "A Mathematical Theory of Cryptography" dated Sept. 1, 1945, which has now been declassified.
[1] Shannon, C. E., "A Mathematical Theory of Communication," *Bell System Technical Journal*, July 1948, p. 379; Oct. 1948, p. 623.
[2] See, for example, H. F. Gaines, "Elementary Cryptanalysis," or M. Givierge, "Cours de Cryptographie."

656

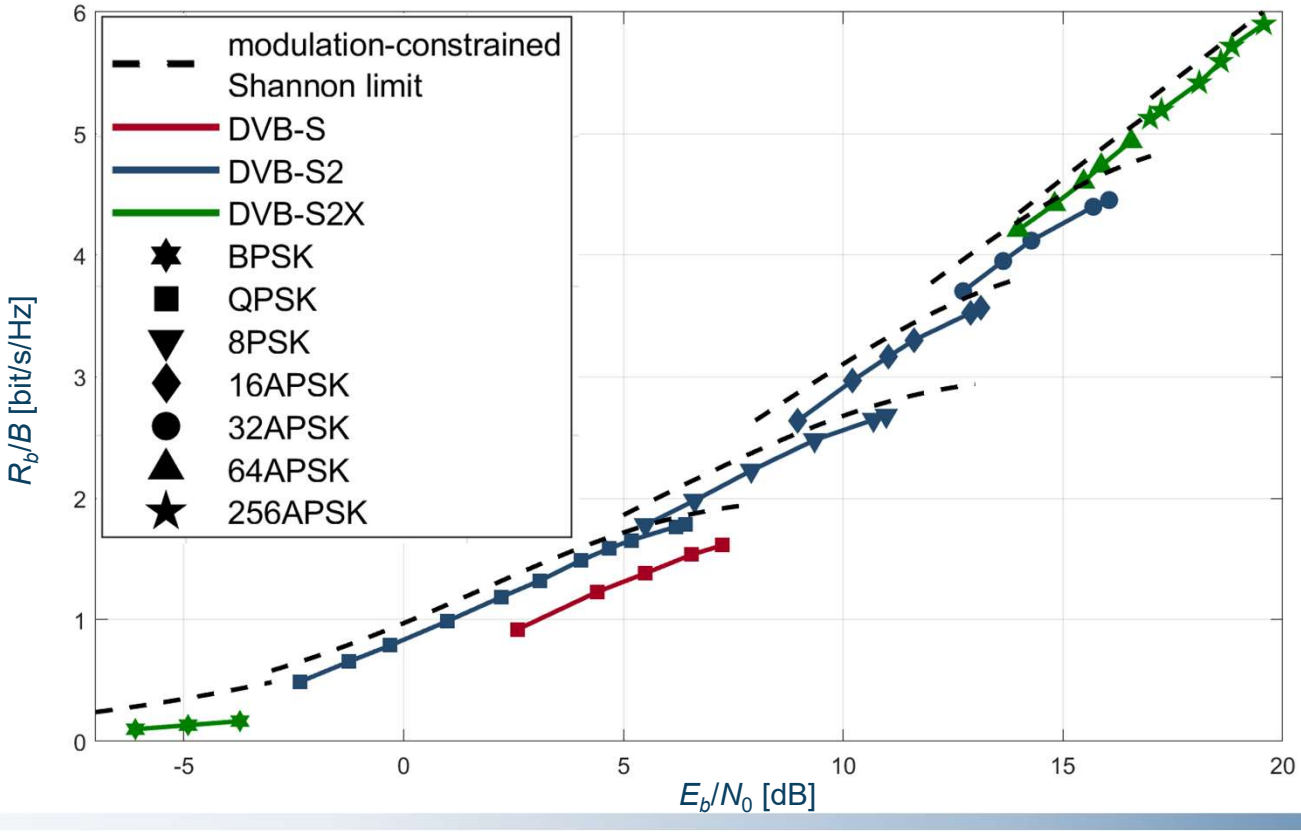


Shannon capacity (2/3)



Shannon capacity (3/3)

Example: adaptive coding and modulation used by satellite standards

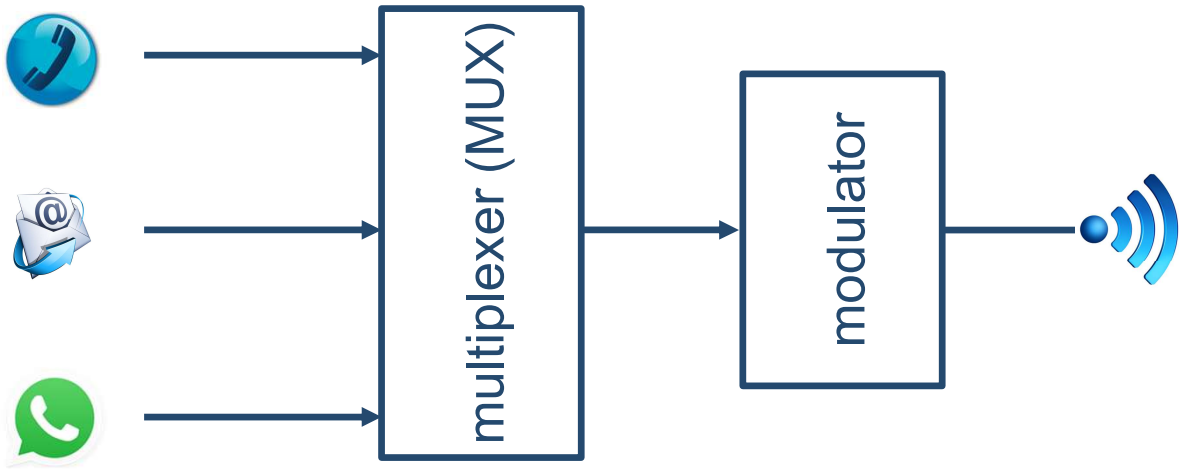




Multiplexing

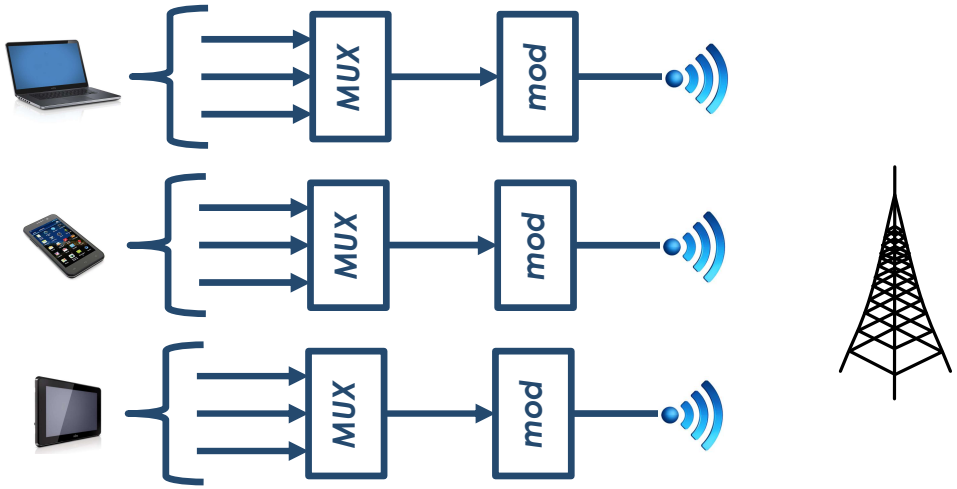
Multiplexing vs. multiple access (1/3)

Multiplexing: separating different flows at the same transmit side



Multiplexing vs. multiple access (2/3)

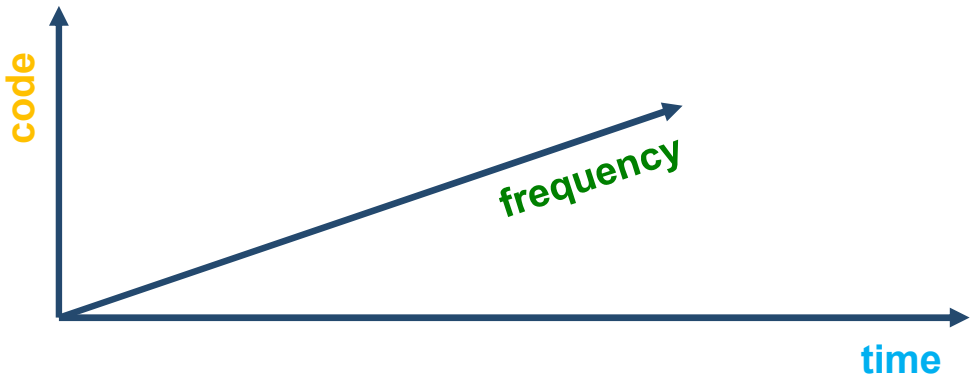
Multiple access: separating different users at the receiver side





Multiplexing vs. multiple access (3/3)

We can exploit several degrees of freedom: frequency, time, space, codes, etc.

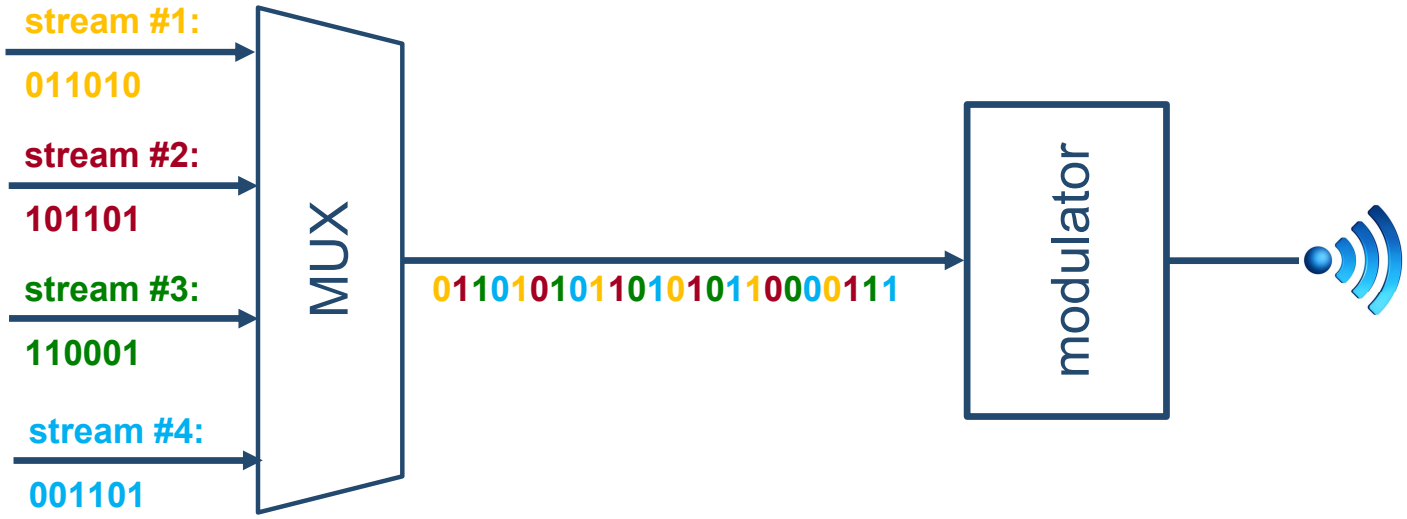




Time division multiplexing (TDM)

Time division multiplexing (TDM) (1/2)

In practice, TDM works as follows:



$R = 1/T$ $R^{(\mathrm{MUX})} = 4/T$

Time division multiplexing (TDM) (2/2)

Each stream makes use of the whole bandwidth using a round robin scheduling, with time slot duration $T^{(\mathrm{MUX})} = T/N$:





Practical applications of TDM

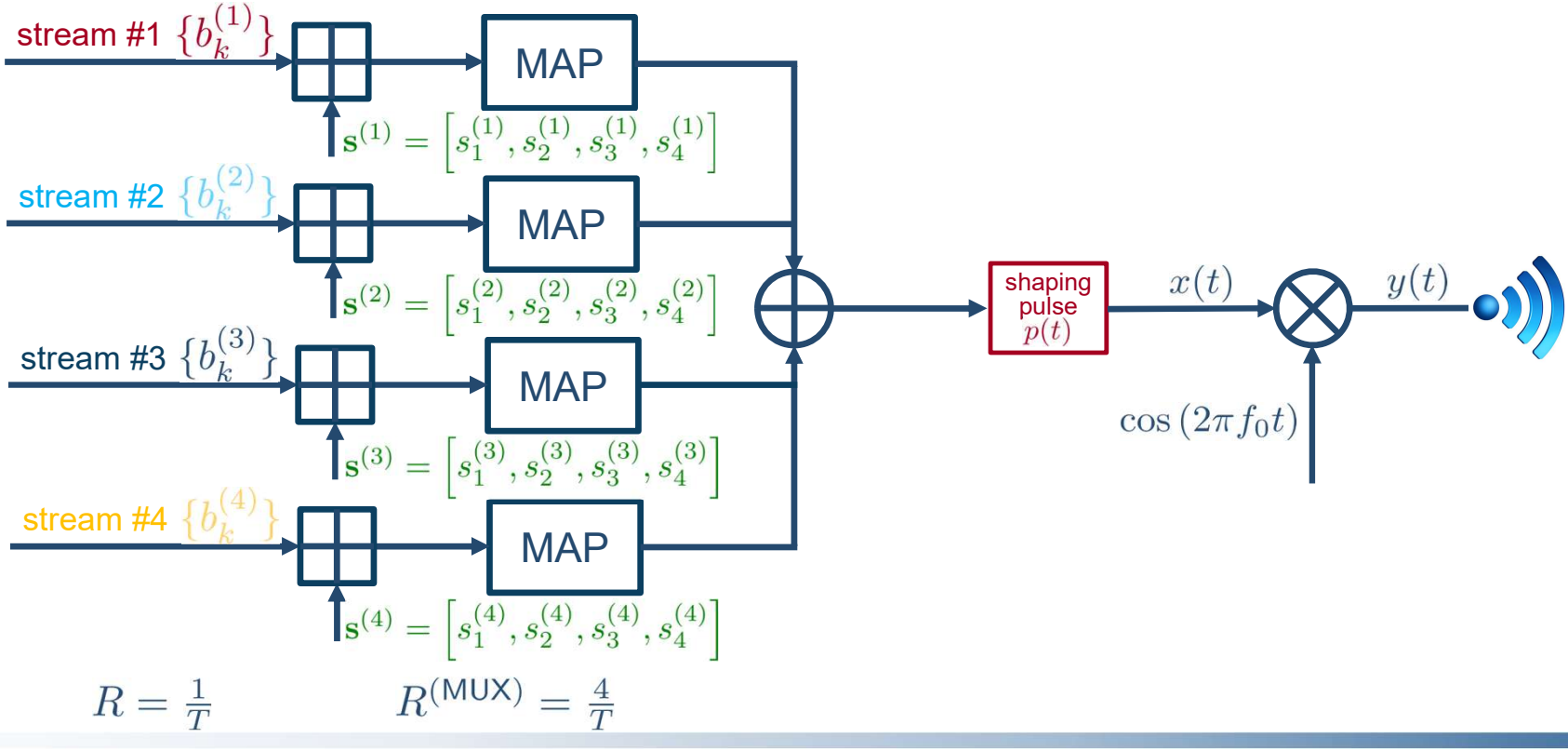
- E1 (Europe) & T1 (USA and Japan): byte-based TDM
- NRZ binary signals, with $R = 64\,\mathrm{kb/s}$ and $T = 15.625\,\mu\mathrm{s}$ are grouped byte by byte: $T_B = 8T = 125\,\mu\mathrm{s}$
- The E1 multiplex includes *N*=32 streams (30: data plane, 2: control plane)
- The TDM signal shows $R^{(\mathrm{MUX})} = N \cdot R = 2.048\,\mathrm{Mb/s}$ and $T^{(\mathrm{MUX})} = T/N \simeq 0.49\,\mu\mathrm{s}$ (for a re-clocked byte time $\simeq 3.9\,\mu\mathrm{s}$)



Code division multiplexing (CDM)

Code division multiplexing (CDM) (1/7)

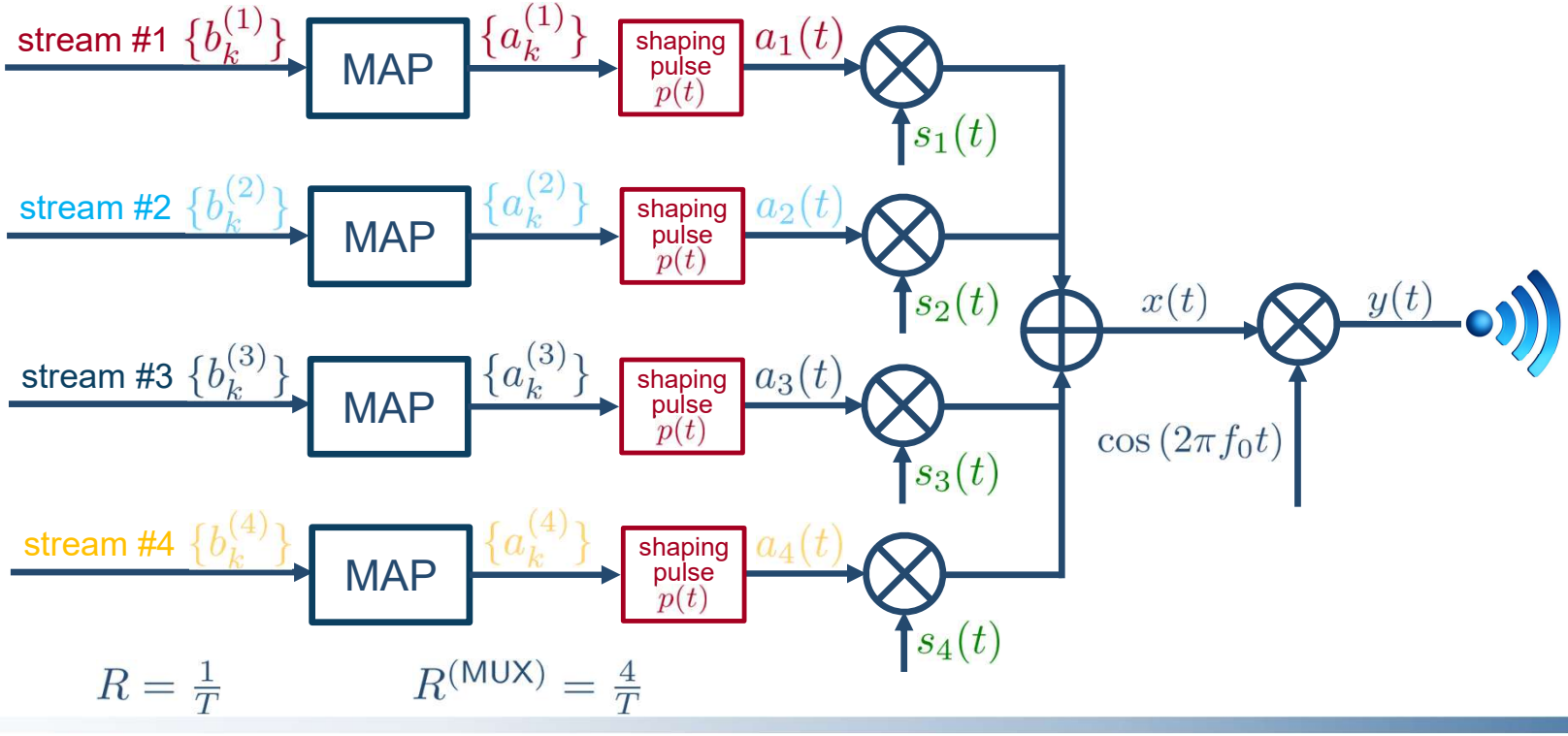
In practice, CDM works as follows:



$R = \frac{1}{T}$ $\quad R^{(\mathrm{MUX})} = \frac{4}{T}$

Code division multiplexing (CDM) (2/7)

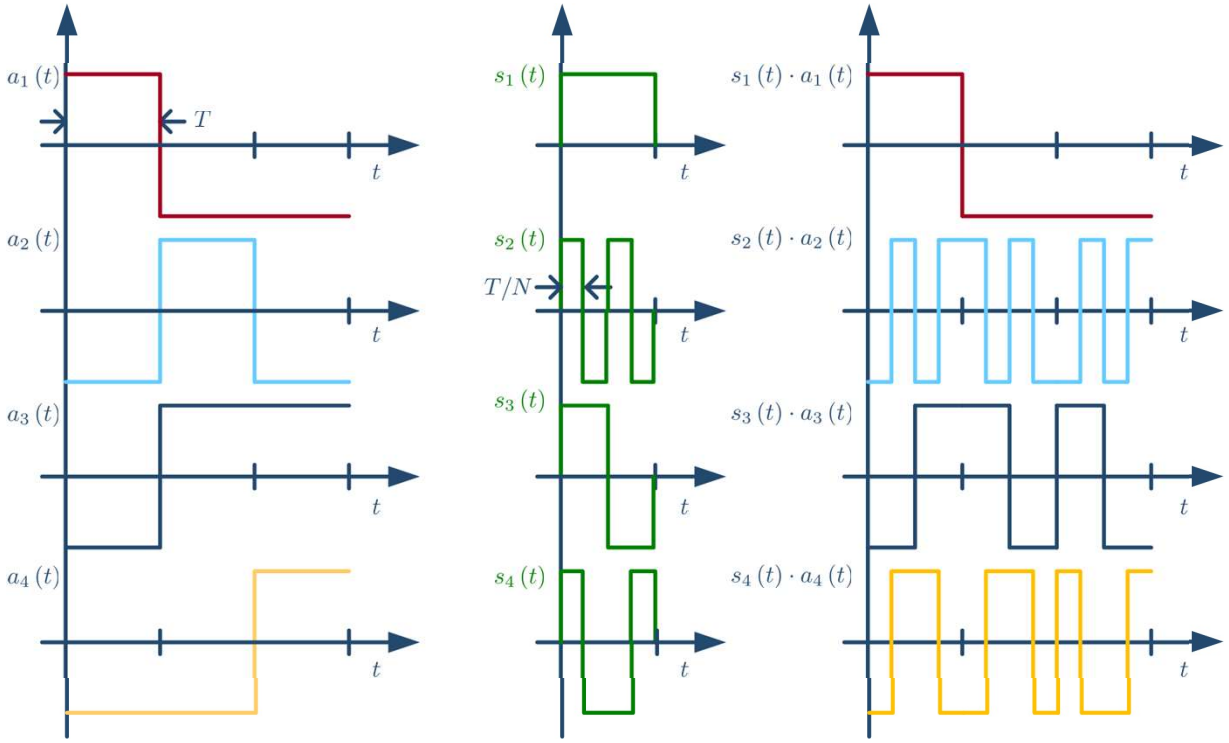
To better visualize it, let us use the following equivalent scheme:



$R = \frac{1}{T}$ $\quad$ $R^{(\mathrm{MUX})} = \frac{4}{T}$

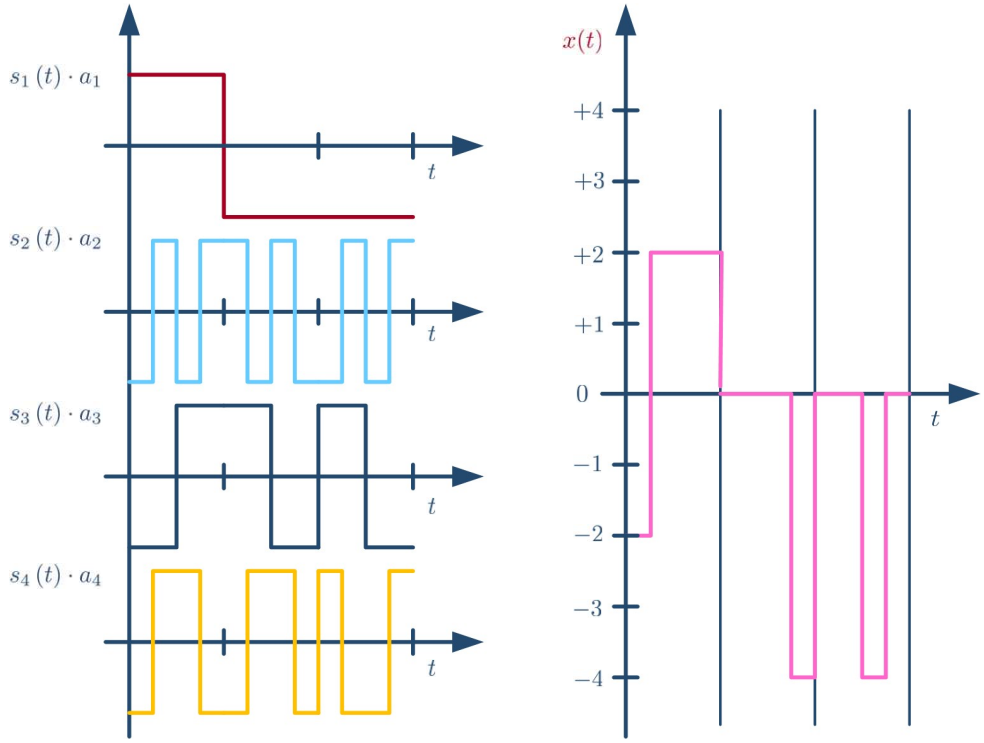
Code division multiplexing (CDM) (3/7)

In CDM, signatures are given by a code set $\{s_n(t)\}_{n=1}^{N}$:



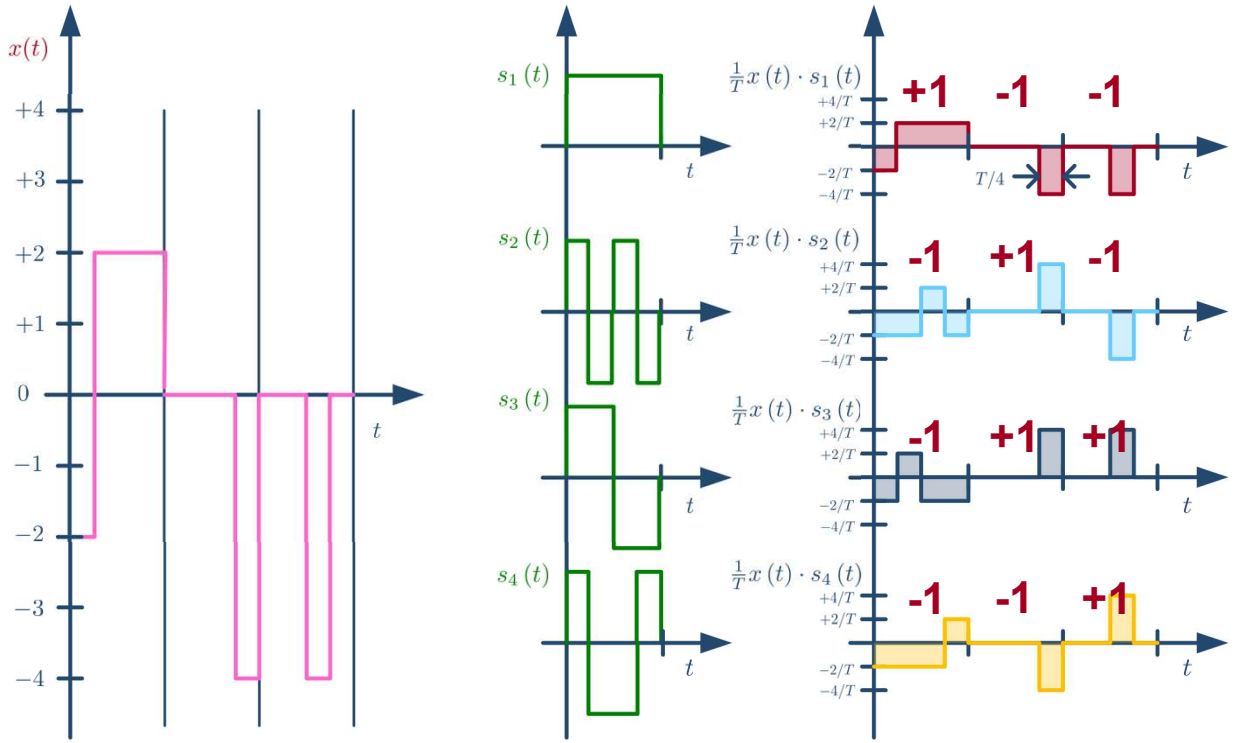
Code division multiplexing (CDM) (4/7)

The CDM-multiplexed signal is the summation of all coded streams:



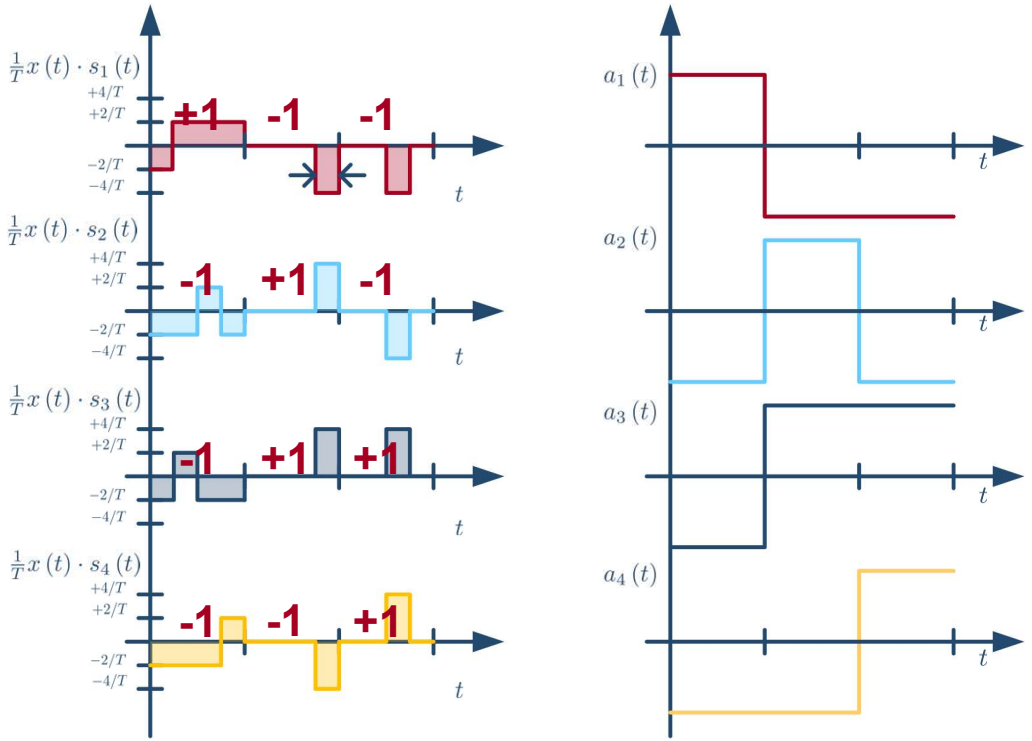
Code division multiplexing (CDM) (5/7)

To decode the *n*th stream, the receiver needs to know the code set:



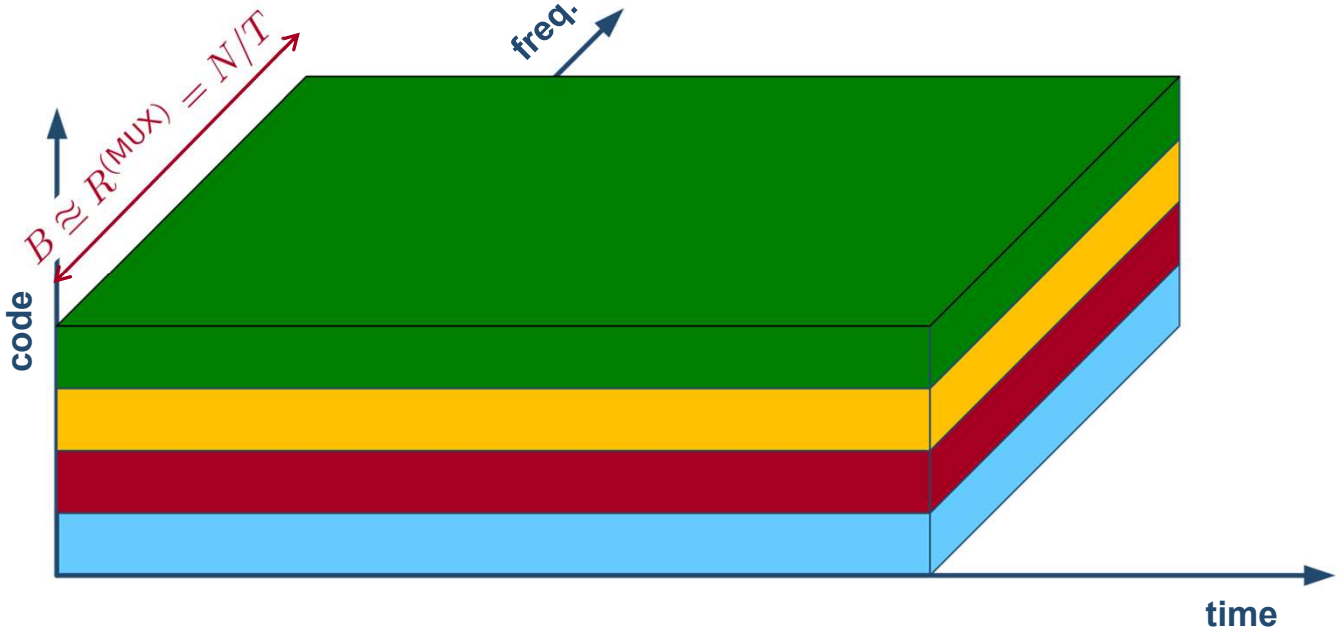
Code division multiplexing (CDM) (6/7)

To decode the *n*th stream, the receiver needs to know the code set:



Code division multiplexing (CDM) (7/7)

This holds true for each CDM signal obtained with orthogonal codes: it makes continuous use of the whole bandwidth:



Walsh-Hadamard codes (1/3)

A useful set of signature codes for CDM is the Walsh-Hadamard (WH) code set

The *n*th code of the WH set is represented by the *n*th row of the *N*-order Hadamard matrix $\mathbf{H}_N$:

$$\mathbf{H}_4 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix} \begin{matrix} \mathbf{s}^{(1)} \\ \mathbf{s}^{(2)} \\ \mathbf{s}^{(3)} \\ \mathbf{s}^{(4)} \end{matrix}$$



Walsh-Hadamard codes (2/3)

The *N*-order WH code set, with cardinality $N = 2^W$, with $W \in \mathbb{N}$, can be obtained in a recursive fashion:

$$\mathbf{H}_2 \triangleq \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}, \quad \mathbf{H}_N = \mathbf{H}_2 \otimes \mathbf{H}_{N/2} = \begin{bmatrix} \mathbf{H}_{N/2} & \mathbf{H}_{N/2} \\ \mathbf{H}_{N/2} & \overline{\mathbf{H}}_{N/2} \end{bmatrix}$$

where $\otimes$ is the Kronecker product, and $\overline{\mathbf{H}}$ is the modulo-2 complement of the matrix $\mathbf{H}$

Walsh-Hadamard codes (3/3)

Examples:

$$\mathbf{H}_2 \triangleq \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}$$

$$\mathbf{H}_4 = \begin{matrix} \mathbf{H}_2 \\ \mathbf{H}_2 \end{matrix} \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix} \begin{matrix} \mathbf{H}_2 \\ \overline{\mathbf{H}}_2 \end{matrix}$$

$$\mathbf{H}_8 = \begin{matrix} \mathbf{H}_4 \\ \mathbf{H}_4 \end{matrix} \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix} \begin{matrix} \mathbf{H}_4 \\ \overline{\mathbf{H}}_4 \end{matrix}$$

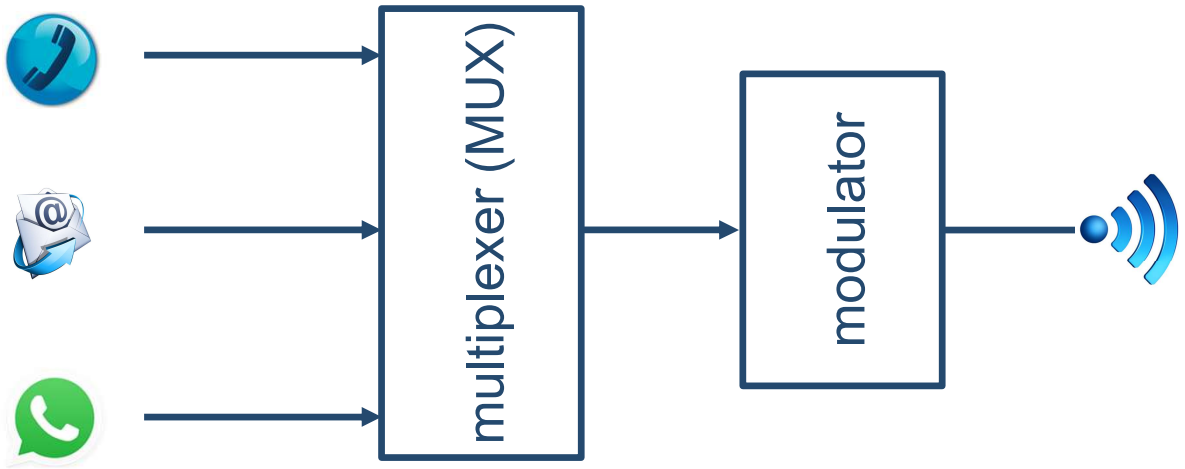


Multiple access



Multiplexing vs. multiple access

Multiplexing: separating different flows at the same transmit side



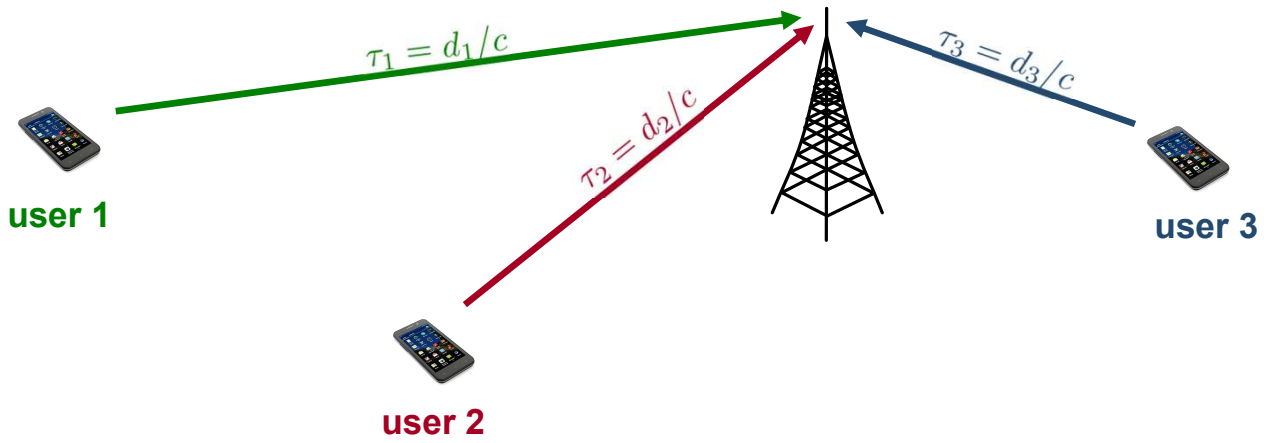


Time division multiple access (TDMA)

Time division multiple access (TDMA) (1/3)

To extend the TDM approach to multiple users, we need to account for a number of additional issues

Example: a typical cellular network





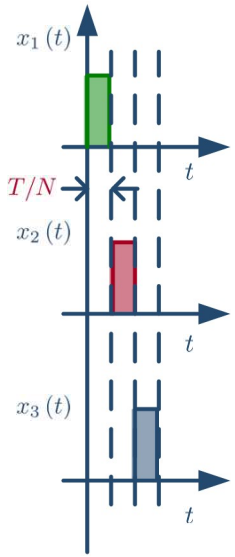
Time division multiple access (TDMA) (2/3)

- Users need to receive their own time slot allocation, assigned by the network through the control plane

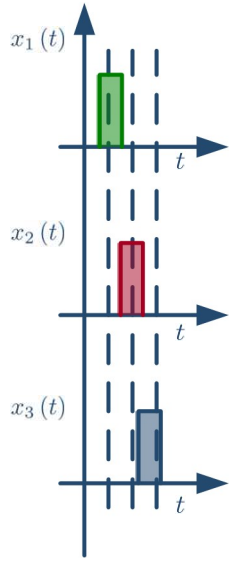
- There is the need for network synchronization to align the users, which can be geographically sparse (and hence with different propagation delays)

Time division multiple access (TDMA) (3/3)

Desired situation:



Actual situation:



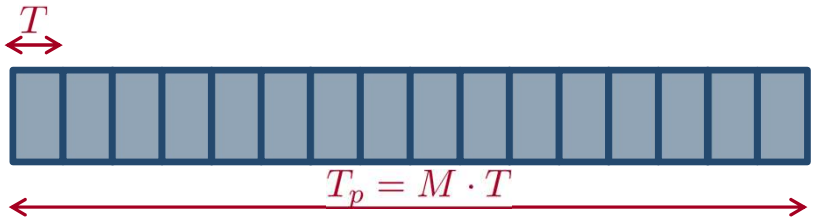
signals from different users overlap due to different propagation delays



Burst-based TDMA (1/2)

To simplify the synchronization tasks, time slots are grouped per burst instead of bit by bit

Each data burst is composed by a packet with duration $T_p = M \cdot T$, where M is the number of bits per packet

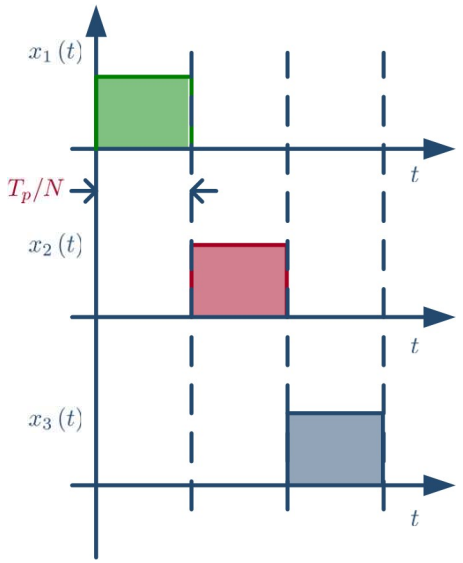


This approach allows the propagation times to be compared with T_p/N rather than with T/N (and hence there is a factor M)

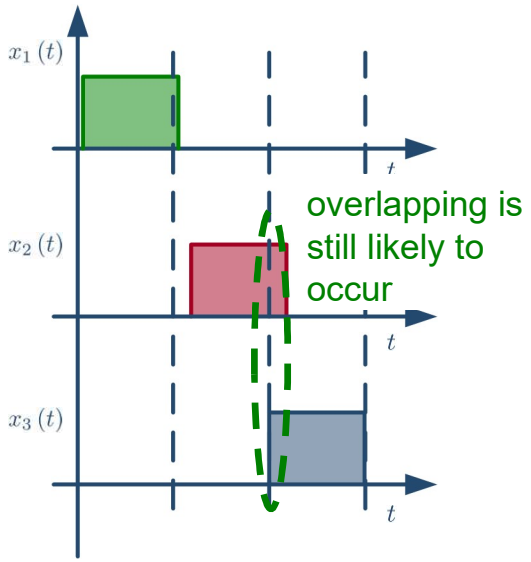
Burst-based TDMA (2/2)

The (centralized) network synchronization feeds back timing information to let the users anticipate or postpone transmissions

Desired situation:

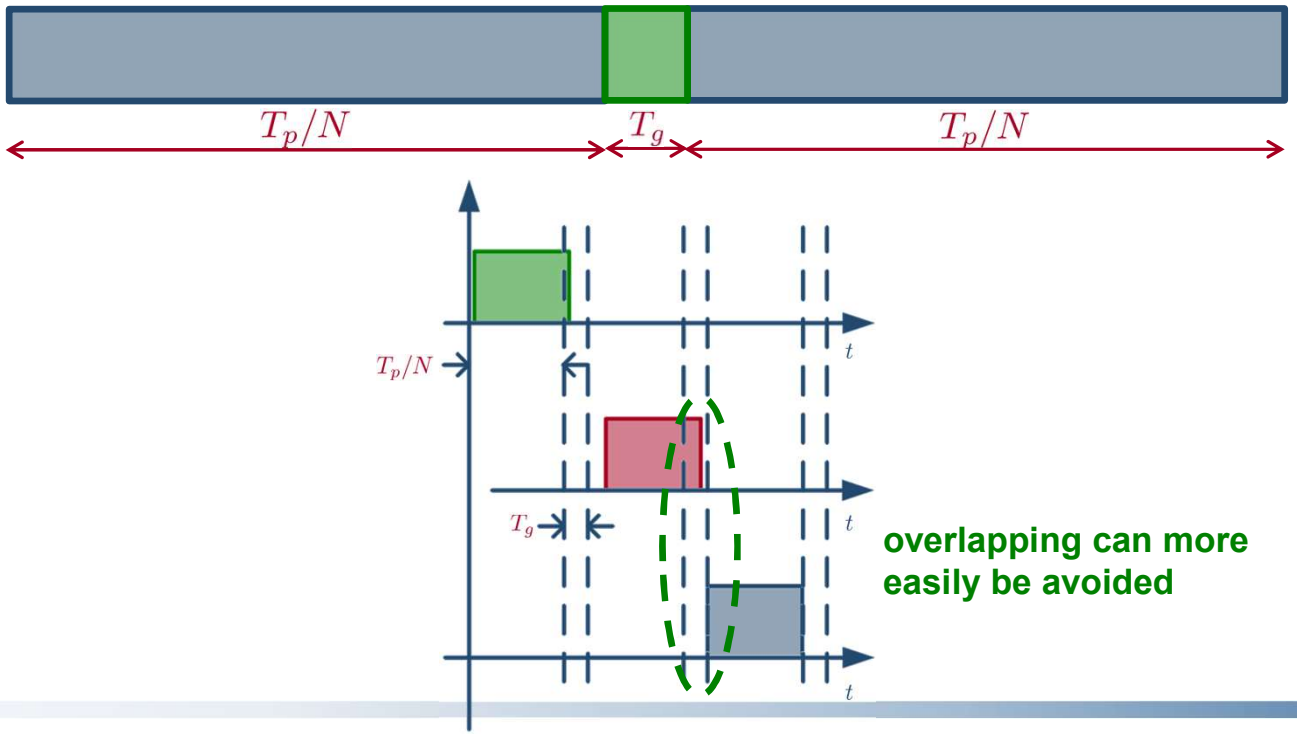


Actual situation:



Introducing a guard interval

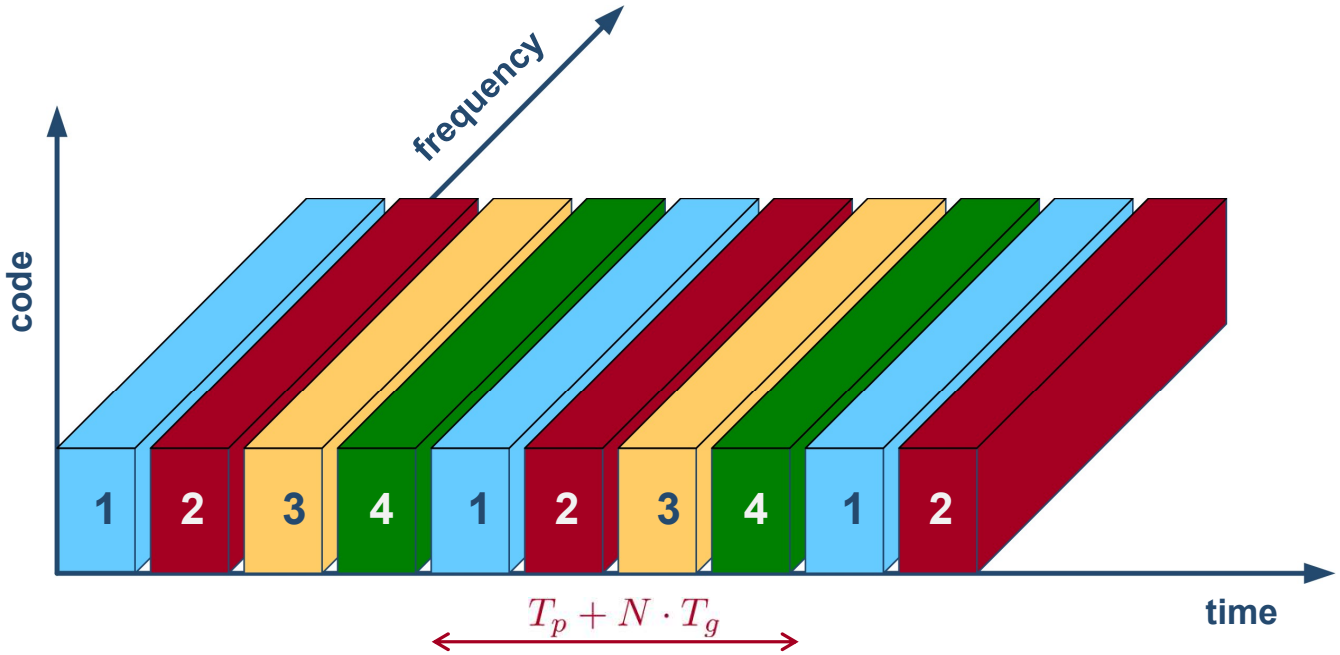
To relax the network synchronization requirements, we can introduce a guard interval T_g between each burst



overlapping can more easily be avoided

TDMA in 3D resource plan

The guard interval is a tradeoff between synchronization performance and optimal usage of network resources



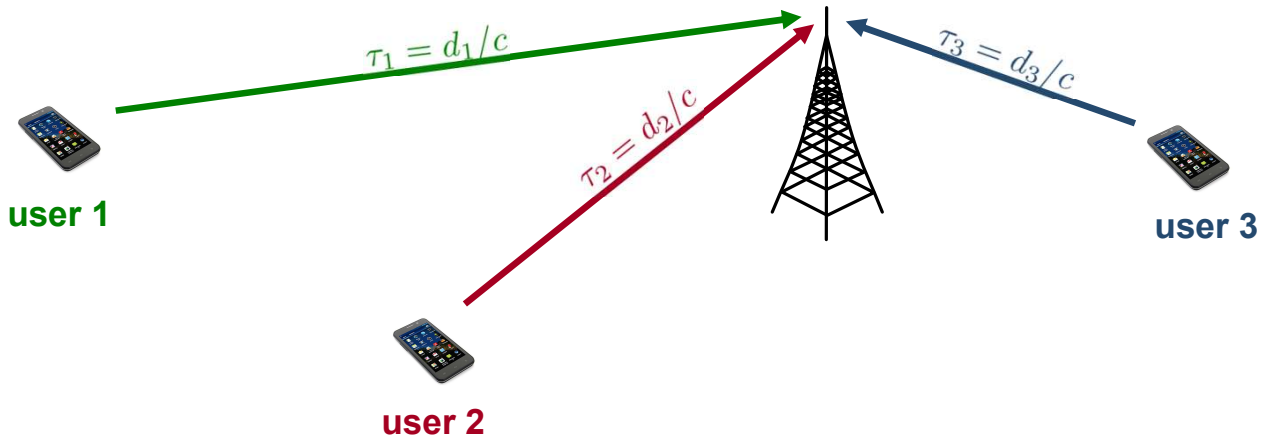


Code division multiple access (CDMA)

Code division multiple access (CDMA) (1/2)

The main difference between CDM and CDMA is due to uncoordinated times of arrival across the network users

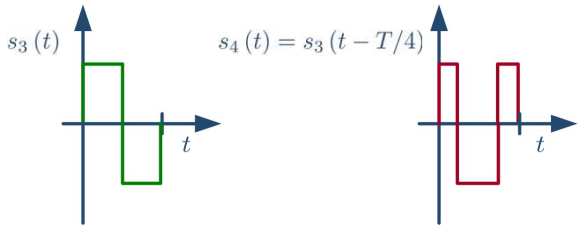
Similarly to the TDMA scenario, different users experience different propagation delays:





Code division multiple access (CDMA) (2/2)

When considering orthogonal codes (such as the WH code set), cross-correlation is negligible if and only if $\Delta\tau \ll T/N = T_c$



This means that, in practice, the requirement is $\Delta\tau \leq T_c/10$

Example: in UMTS (3G), $T_c \approxeq 0.26\,\mu\text{s}$, which implies $\Delta\tau \leq 26\,\text{ns}$

This accuracy is not viable, especially if compared with that required by TDMA, in which $\Delta\tau \leq T_g \approx T$ (typically, accuracy is two-three orders of magnitude higher in CDMA)



Non-orthogonal codes (1/3)

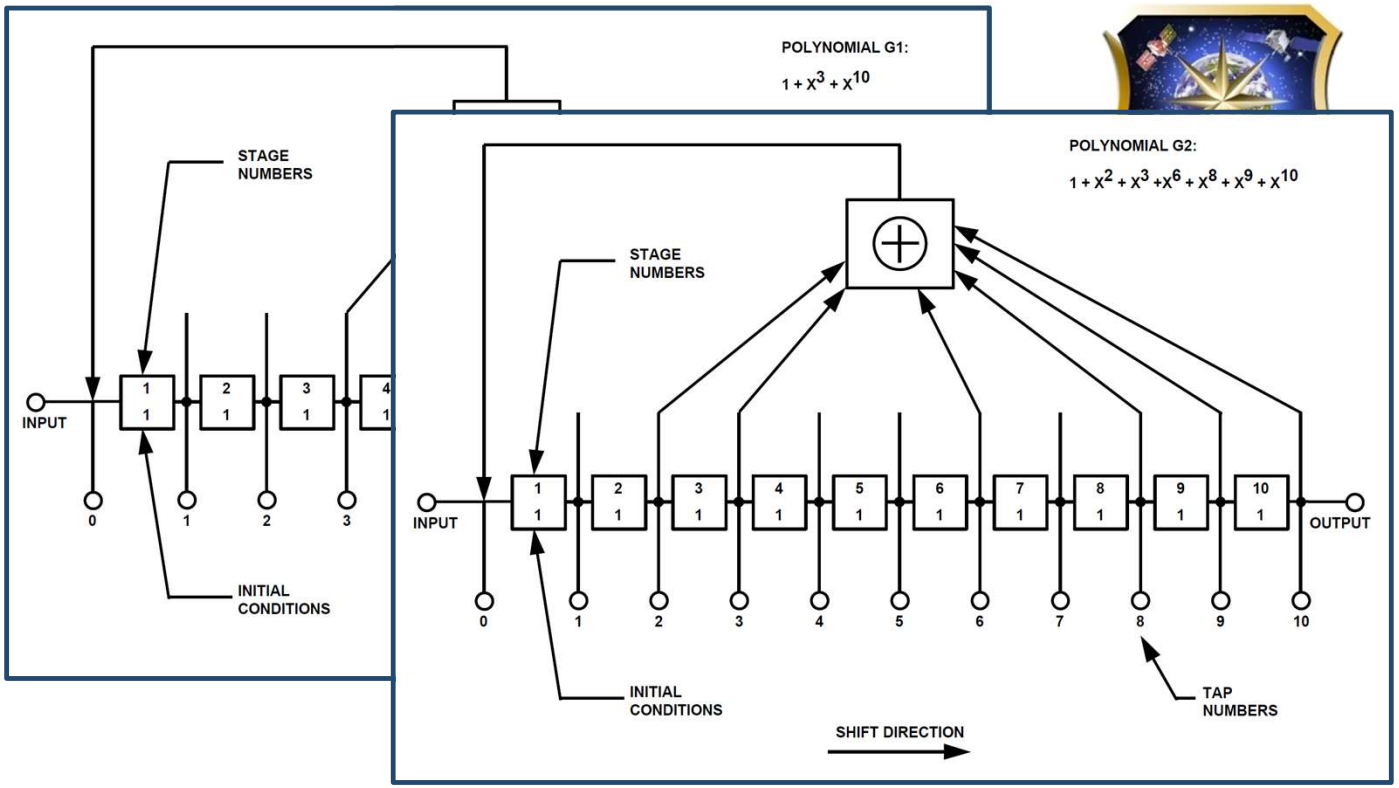
In this context, we need to identify signature codes that work pretty well in the asynchronous scenario

The class of pseudo-random noise (PRN) sequences, such as the Gold codes used in many systems, provide good performance in fully uncoordinated transmission patterns

PRN codes show similarities with the AWGN (e.g., the PSD is almost flat in the frequency domain), due to the pseudo-random nature of the chips; however, they are generated according to a well-defined algorithm, known to both transmit and receive sides

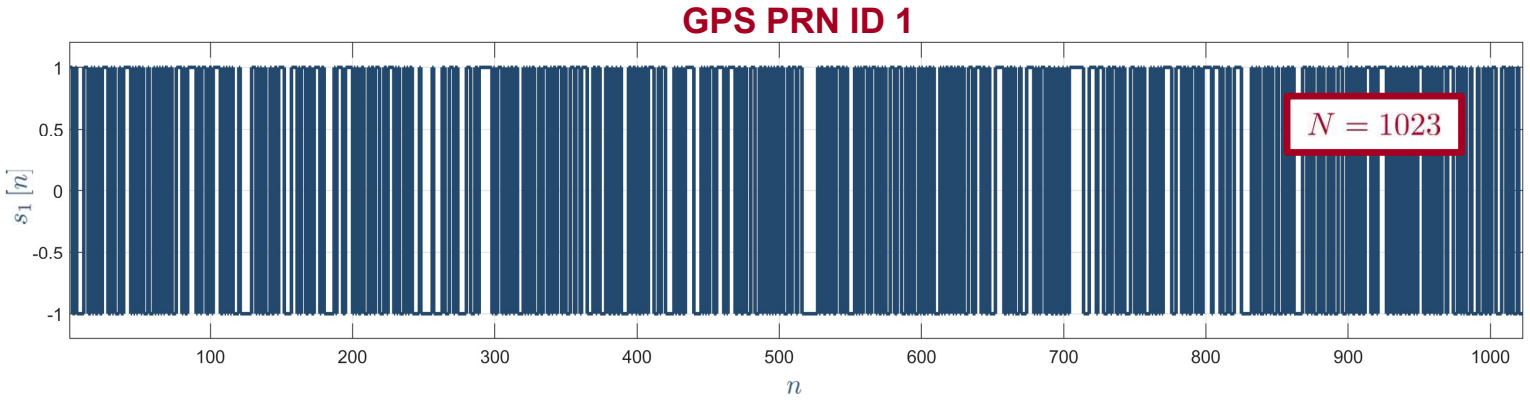
Non-orthogonal codes (2/3)

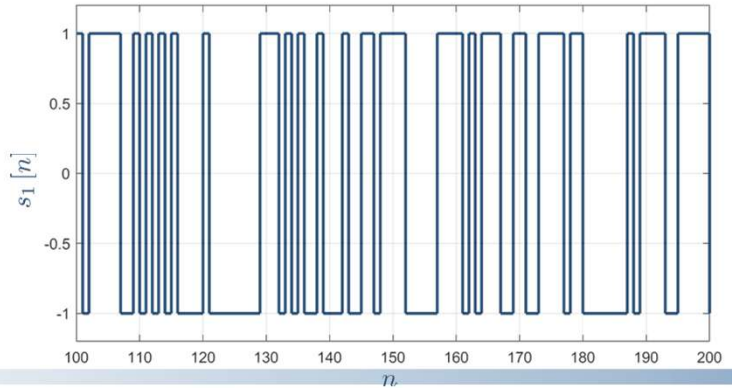
Example: PRN code generation in global positioning system (GPS)



Non-orthogonal codes (3/3)

Example: PRN code generation in global positioning system (GPS)

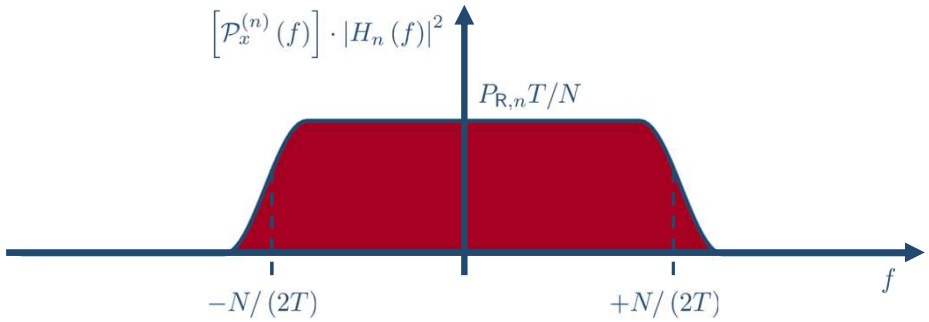






Performance of CDMA (1/4)

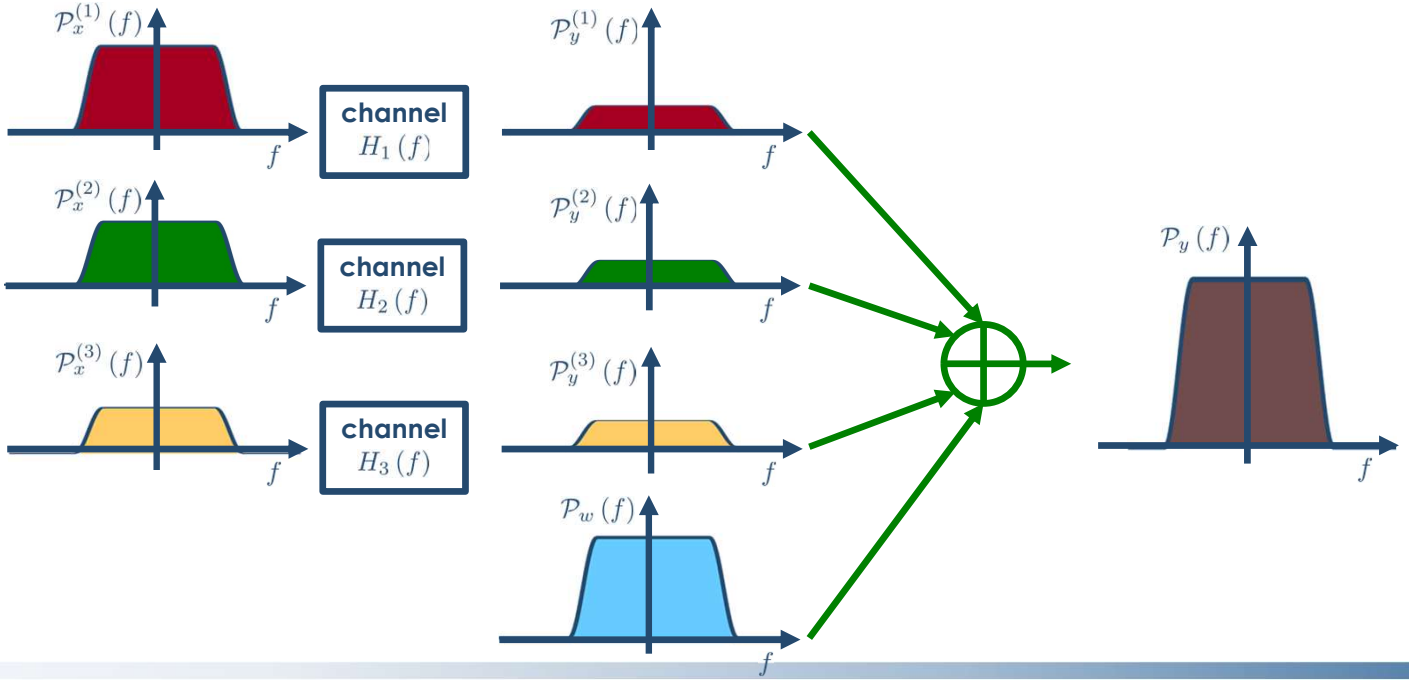
Thanks to the ad-hoc code chip generation, PRN codes show a flat PSD:



where $H_n(f)$ is the DFT of the channel experienced by user n, and $P_{\mathrm{R},n}$ is user n's received power

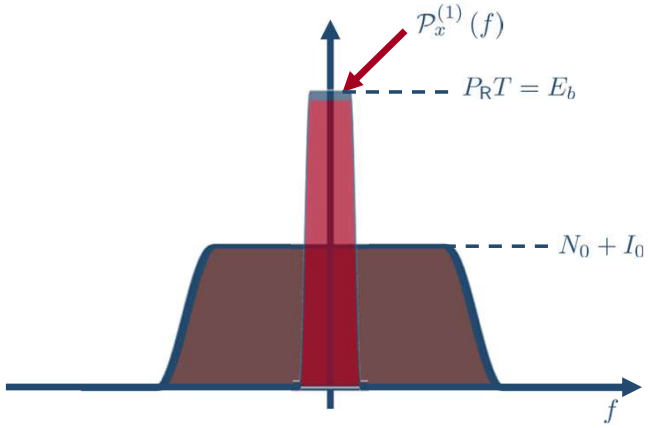
Performance of CDMA (2/4)

Hypothesis: our CDMA system adopts a closed-loop power contol that equalizes all received powers $\{P_{\mathrm{R},n}\}_{n=1}^{N_A} = P_{\mathrm{R}}$, where N_A is the number of users simultaneously active (note: $N_A \lessgtr N$)



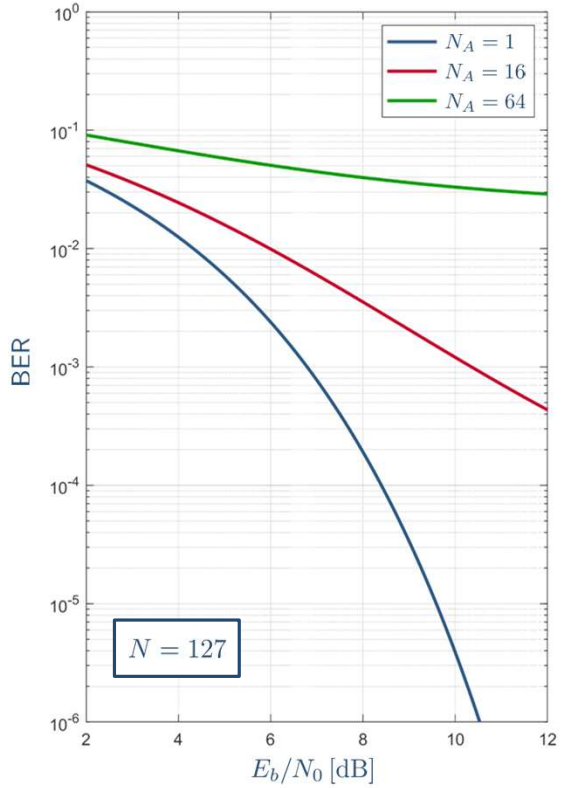
Performance of CDMA (3/4)

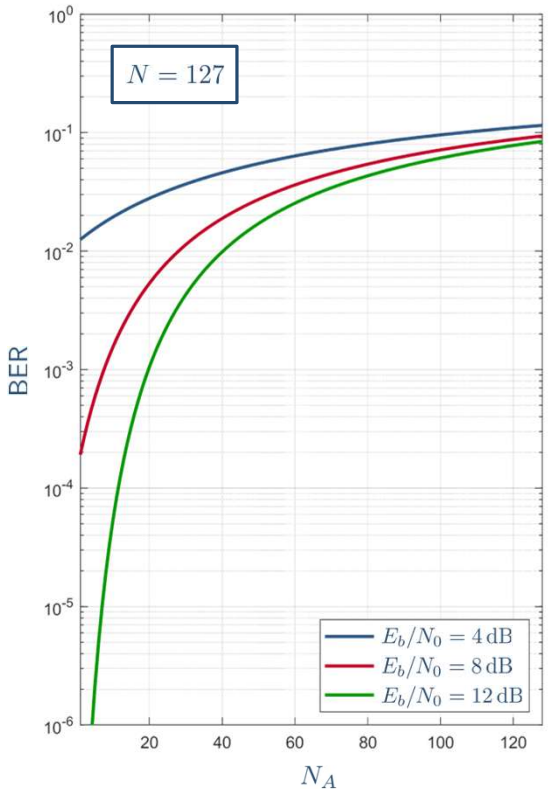
Decoding the *n*th user by applying the matched-filter approach is equivalent to extract only the desired spectrum out of the compound signal:



where the interference to the other users (the so-called multiple access interference, MAI) can be considered as a white Gaussian process as soon as $N_A \gg 1$ (say, $N_A \geq 10$)

Performance of CDMA (4/4)







Differences wrt TDMA

Non-orthogonality of the signature codes significantly degrades the performance of the network

However, CDMA allows users to access the network in a truly random manner, without the need for synchronization across the users

This places CDMA closer to statistical-based multiple-access techniques, such as packet-oriented ones



Frequency division multiple access (FDMA)



Frequency division multiple access (FDMA) (1/3)

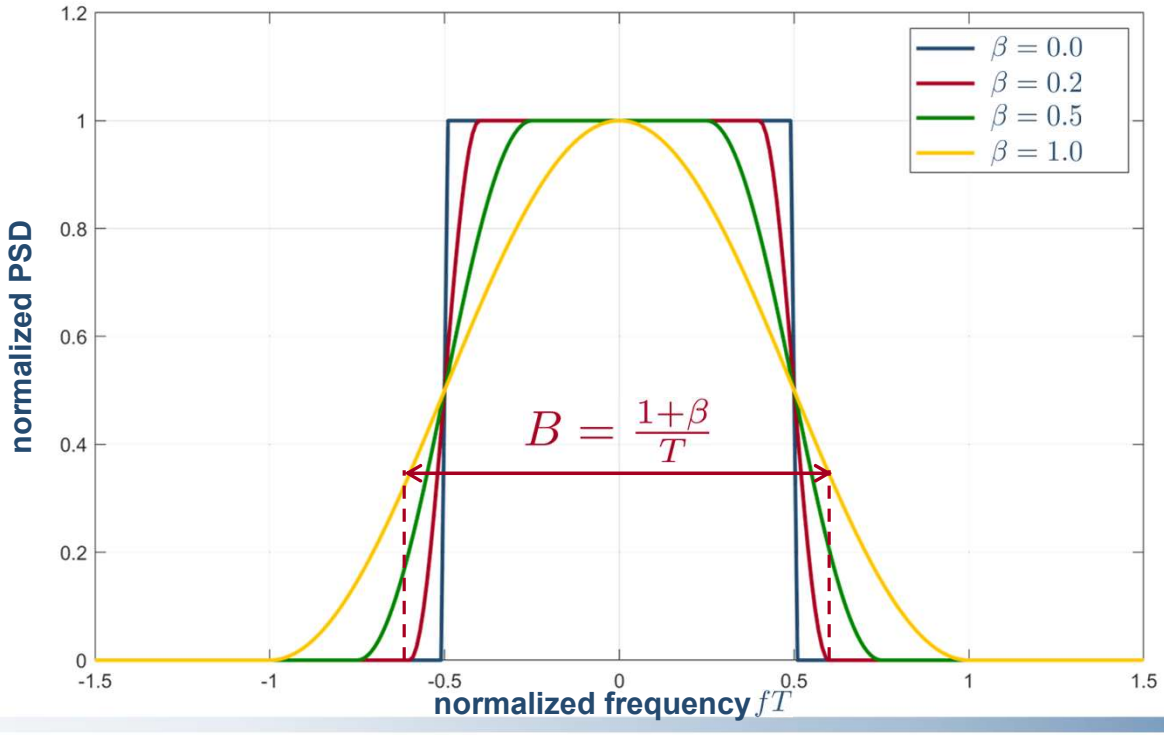
Historically, the first multiple access technique adopted in radio communications is based on FDMA

Constraint: each signal must be strictly band-limited

Note that there are no requirements in terms of time or phase alignment across different channels

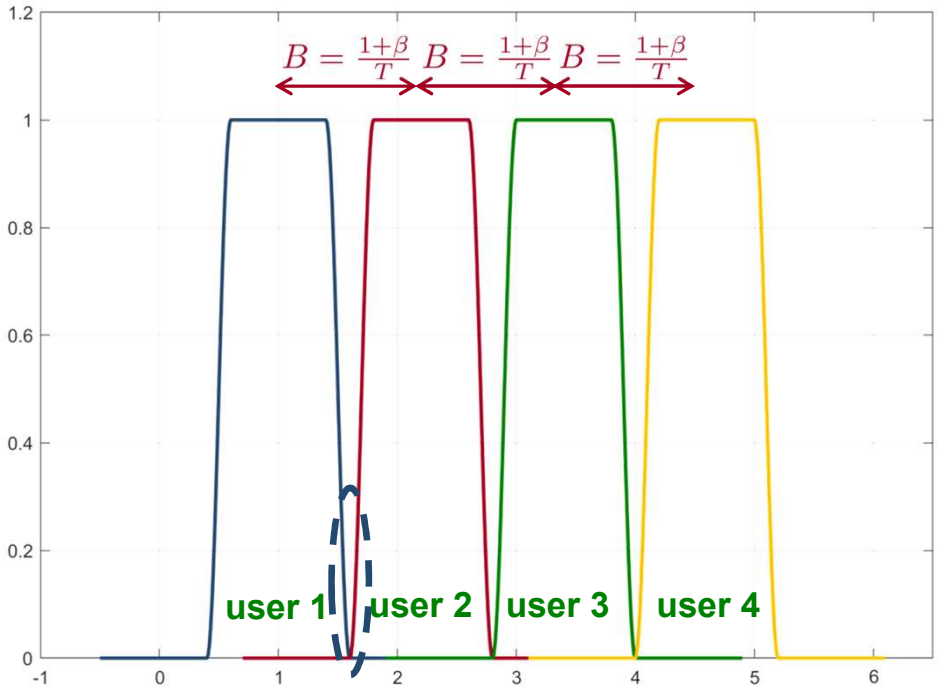
Frequency division multiple access (FDMA) (2/3)

Let us assume to consider a QAM signal modulated with a square root raised cosine (SRRC) shaping pulse:



Frequency division multiple access (FDMA) (3/3)

In the frequency domain, the multiple-access signal takes the form

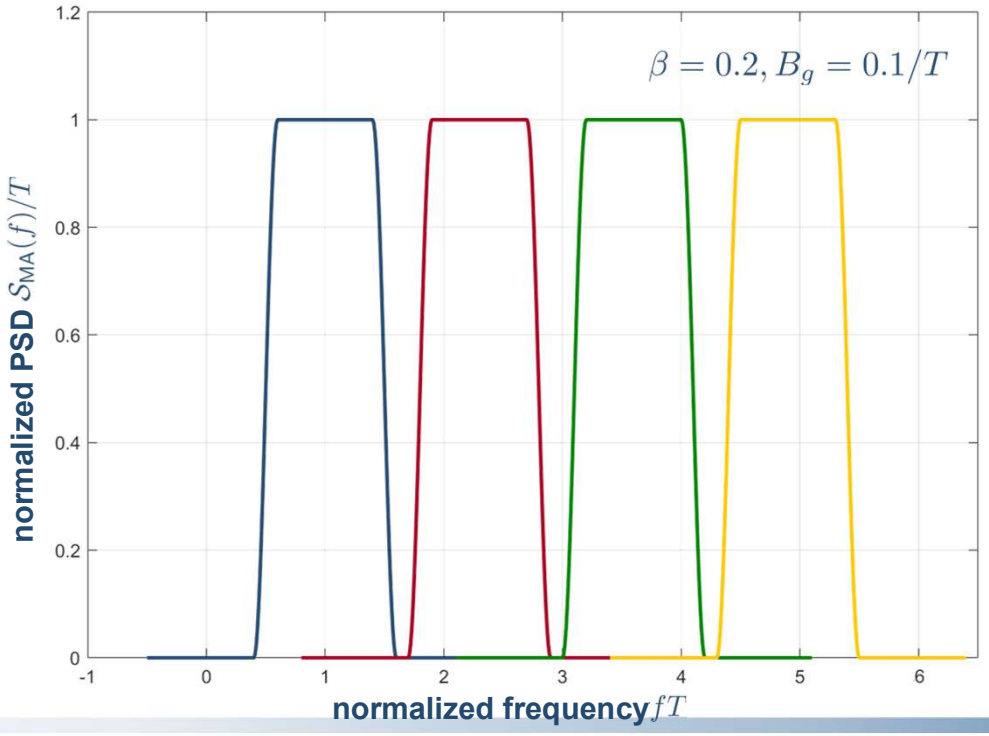


To mitigate carrier frequency instabilities, we can add a guard bandwidth B_g



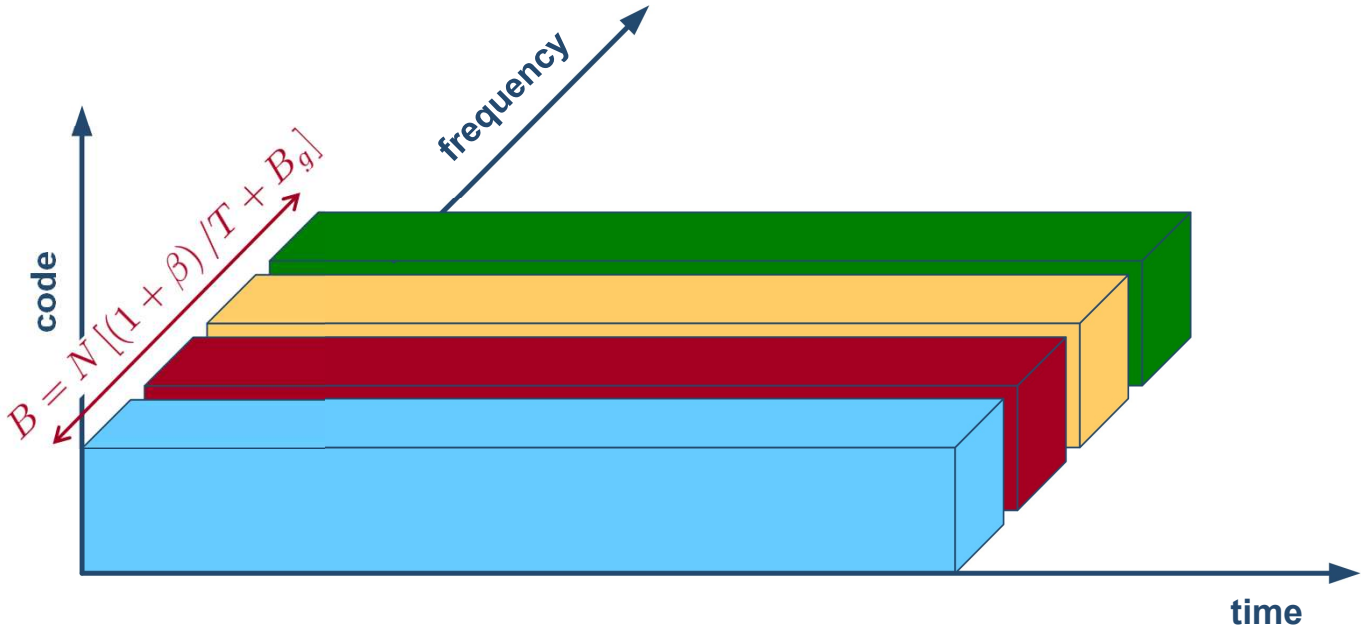
Guard bandwidth (1/2)

Increased separation (and hence easier architecture at both tx and rx sides) comes at the cost of additional resource waste



Guard bandwidth (2/2)

FDMA in 3D resource plan can be viewed as





Packet-oriented multiple access



What is packet-oriented multiple access? (1/2)

Connection-oriented multiple access:

- coordinated access, such as in TDMA, FDMA, and, to some extent, CDMA
- deterministic approach, well suited for constant-bitrate communications

Packet-oriented multiple access:

- uncoordinated access
- statistical approach, well suited for bursty-traffic communications



What is packet-oriented multiple access? (2/2)

Typical example of bursty/impulsive traffic: the IoT scenario









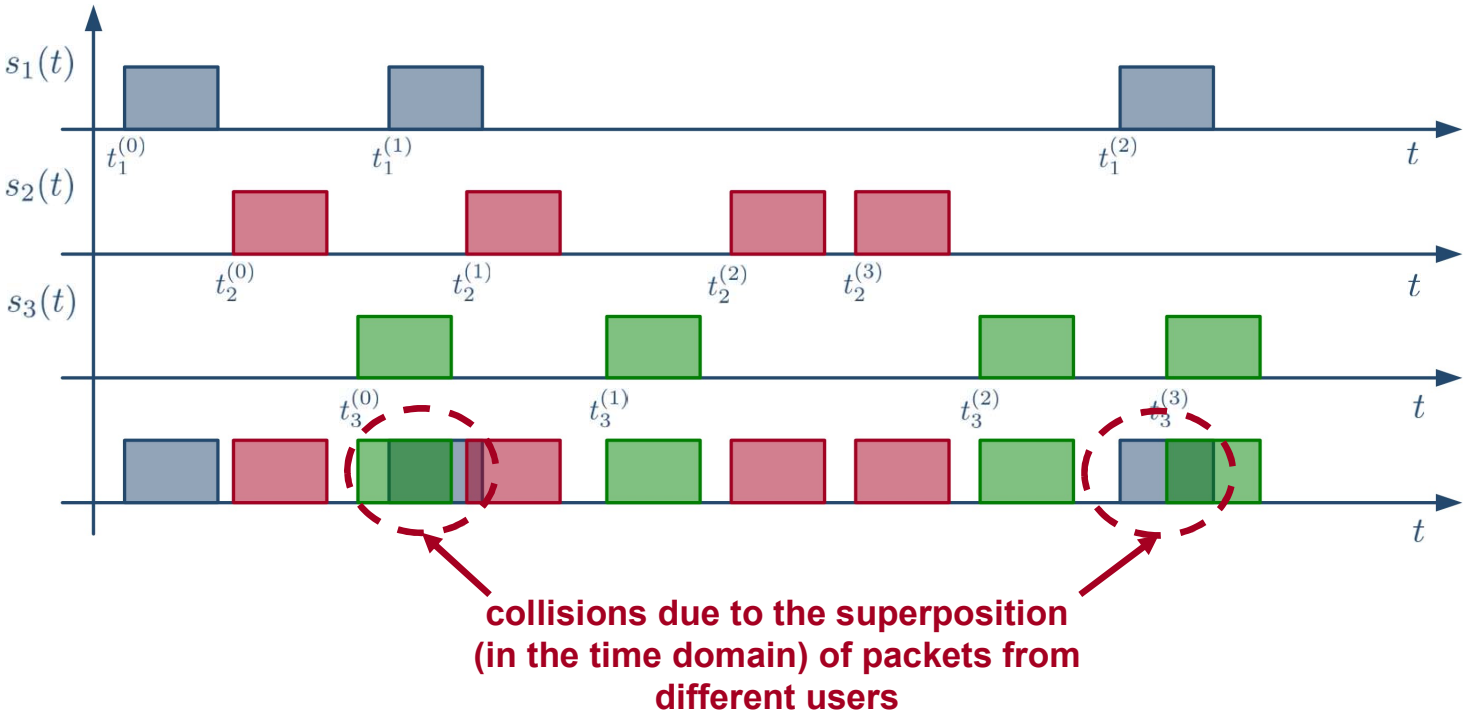


The ALOHA protocol (1/3)

Historically, the first protocol to address this issue was the ALOHAnet, or simply ALOHA, devised in the early 1970s

The main goal was to connect the research centers at the University of Hawaii, spread across the Hawaiian islands, with the main Oahu campus

The ALOHA protocol (2/3)





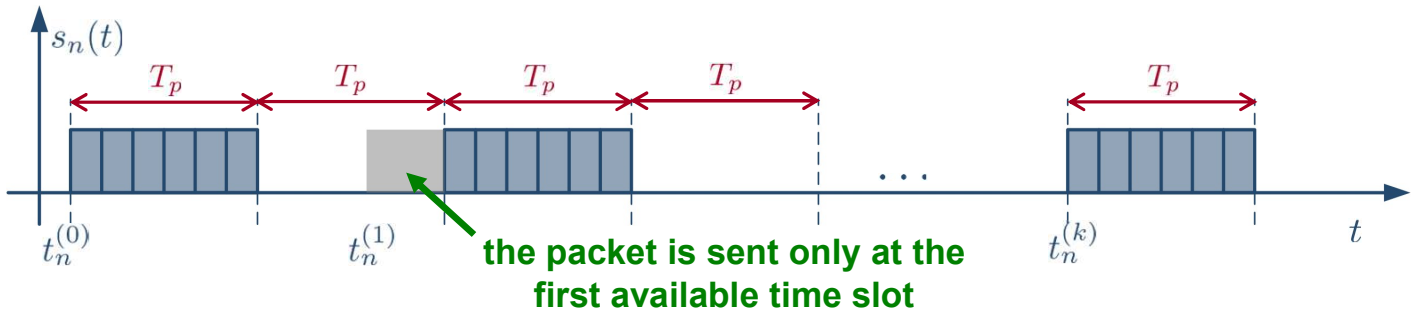
The ALOHA protocol (3/3)

Unlike connection-oriented multiple-access scheme, the ALOHA protocol needs a return channel, to receive feedback from the receiver (using acknowledgment of correct reception)

In case of a missing acknowledgement (corresponding to an event of collision), the re-transmission follows a specific backoff time (based on a statistical approach – out of scope in this lecture)

The slotted-ALOHA (S-ALOHA) protocol (1/2)

To reduce the occurrence of collision events, we can apply a centralized network synchronization similar to TDMA:

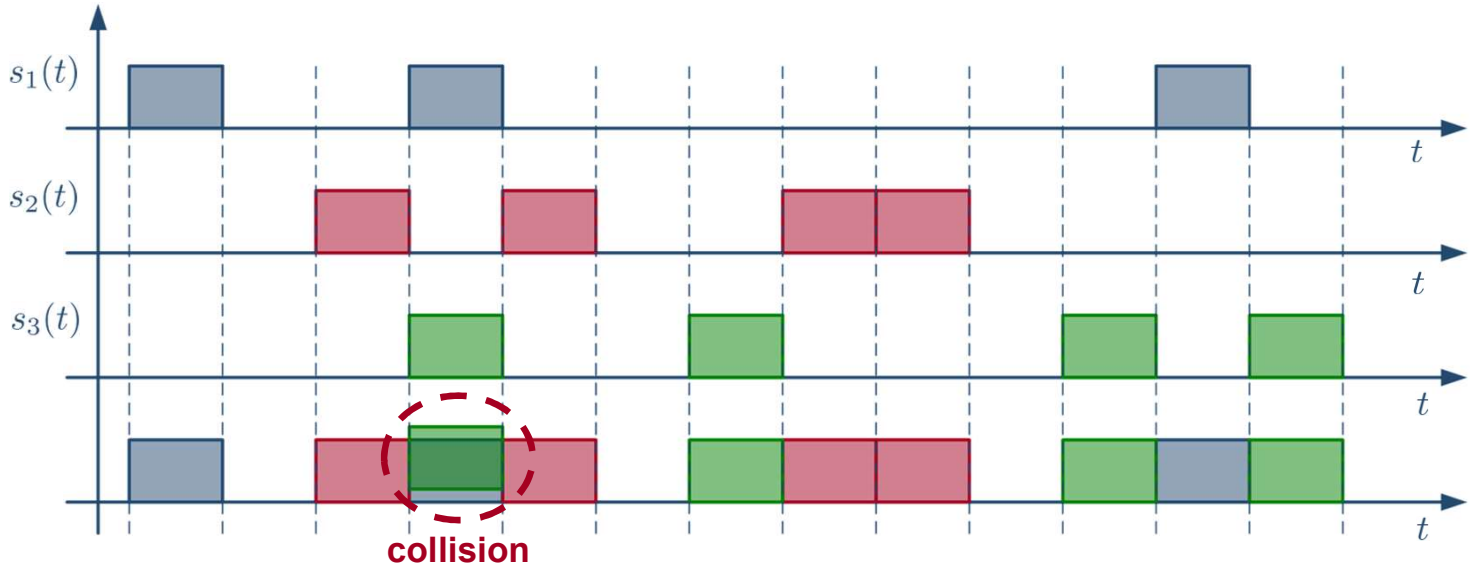


This modified version of ALOHA is called slotted ALOHA (S-ALOHA)

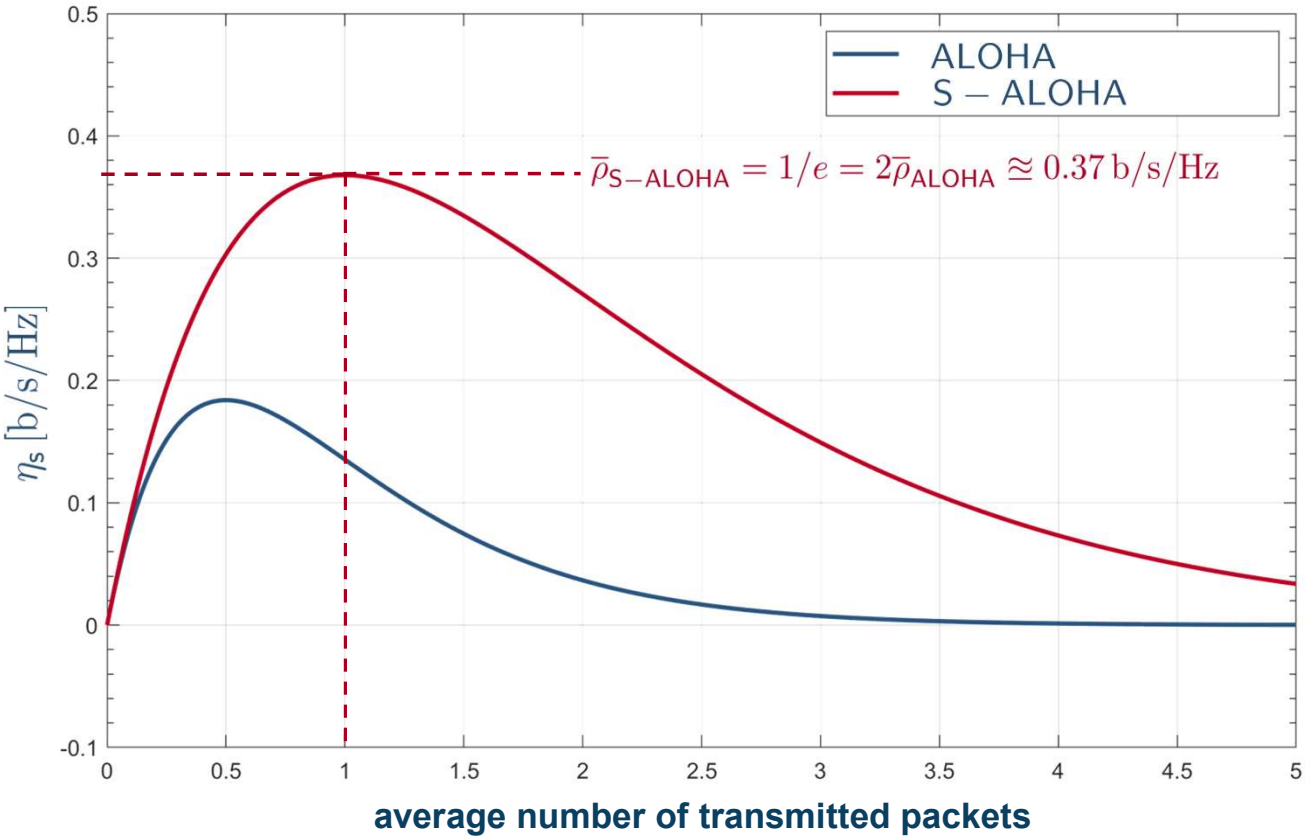


The slotted-ALOHA (S-ALOHA) protocol (2/2)

Collisions occur only when more than one source selects the same time slot to transmit the packet (i.e., partial collisions do not exist anymore)



ALOHA vs. S-ALOHA





Efficiency comparison across multiplexing and multiple access techniques



Efficiency of deterministic-based approaches

It is now interesting to compare the packet-oriented and the connection-oriented approaches in terms of spectral efficiency:

$$\eta_s = \frac{\text{bitrate successfully delivered}}{\text{used bandwidth}}$$



Efficiency of multiplexing techniques

When considering frequency / time / code division multiplexing:

- used bandwidth: $B = N/T$
- bitrate successfully delivered: $R_b = N \cdot 1/T = N/T$

As a consequence,

$$\eta_{\mathrm{s,TDM}} = \eta_{\mathrm{s,CDM}} = 1\,\mathrm{b/s/Hz}$$

To sum up: multiplexing has unitary efficiency, thanks to its orthogonality



Efficiency of FDMA

When dealing with FDMA, we need to account for the guard bandwidth B_g:

- used bandwidth: $B = N \cdot [(1+\beta)/T + B_g]$
- bitrate successfully delivered: $R_b = N \cdot 1/T = N/T$

As a consequence,

$$\eta_{\text{s,FDMA}} = \frac{N/T}{N \cdot [(1+\beta)/T + B_g]} = \frac{1}{1+\beta+B_g T}$$

Example: with $\beta = 0.2$ and $B_g = 0.1/T$, we get $\eta_{\text{s,FDMA}} \approx 0.77\,\text{b/s/Hz}$



Efficiency of TDMA

Similarly, when dealing with TDMA, efficiency is reduced by the guard interval T_g:

- used bandwidth: $B = N/T$
- bitrate successfully delivered: $R_b = \frac{N}{T} \cdot \frac{T_p/N}{T_p/N + T_g}$

This implies that

$$\eta_{\text{s,TDMA}} = \frac{N/T \cdot (MT/N)}{N/T \cdot (MT/N + T_g)} = \frac{1}{1 + NT_g/(MT)}$$

Example: with $T_g = 8.25T/N$ and $M = 148$ (GSM), we get $\eta_{\text{s,TDMA}} \approxeq 0.95\,\text{b/s/Hz}$



Efficiency of CDMA (1/3)

With CDMA, the efficiency can be measured as follows:

- used bandwidth: $B = N/T$
- bitrate successfully delivered: $R_b = N_A \cdot 1/T$

which implies

$$\eta_{\mathsf{s,CDMA}} = \frac{N_A}{N}$$

However, deriving the number of simultaneously active users N_A is not straight-forward, as it depends on the desired quality of service (QoS), which impacts the selected SNR E_b/N_0

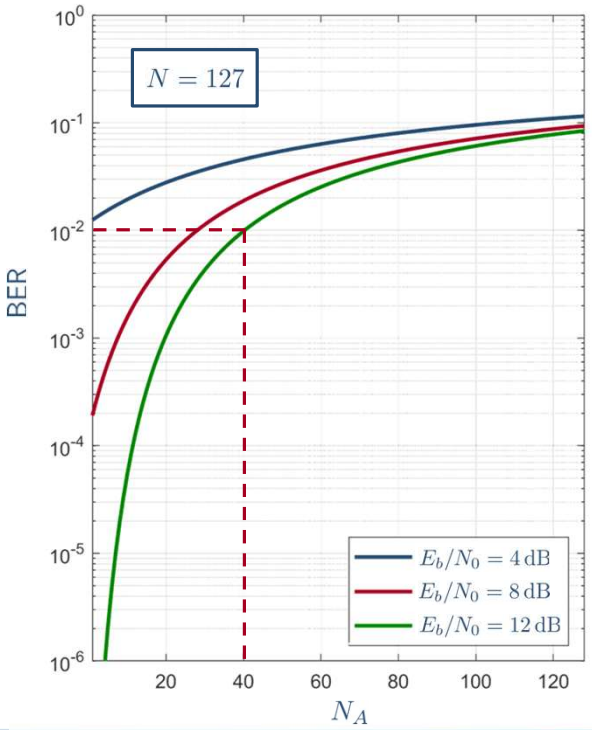
Efficiency of CDMA (2/3)

Example: $E_b/N_0 = 12\,\mathrm{dB}$, that yields $\mathrm{BER} < 10^{-7}$ in the single-user case

Target BER in the multi-user case (voice applications):

$\mathrm{BER} = 10^{-2}$

$$\eta_{\mathrm{s,CDMA}} = \frac{N_A}{N} = \frac{40}{127} \approxeq 0.31\,\mathrm{b/s/Hz}$$



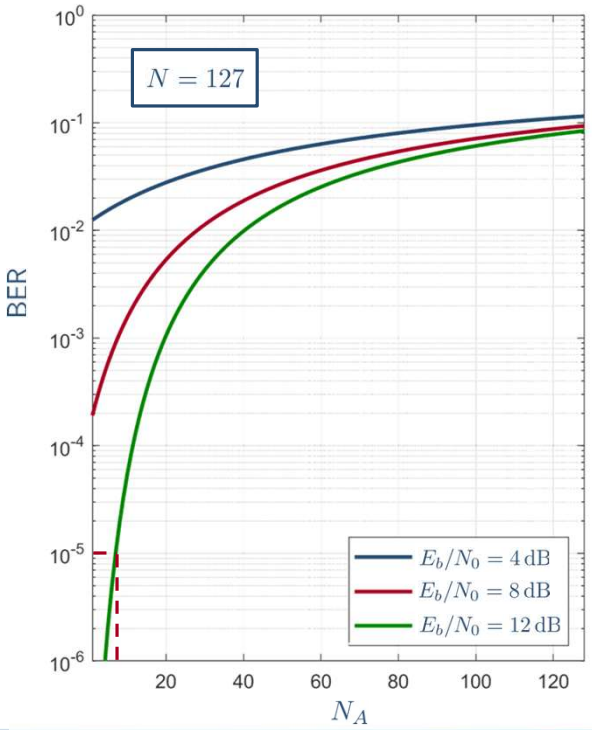
Efficiency of CDMA (3/3)

Example: $E_b/N_0 = 12\,\mathrm{dB}$, that yields $\mathrm{BER} < 10^{-7}$ in the single-user case

Target BER in the multi-user case (video applications):

$\mathrm{BER} = 10^{-5}$

$$\eta_{\mathrm{s,CDMA}} = \frac{N_A}{N} = \frac{7}{127} \approxeq 0.06\,\mathrm{b/s/Hz}$$





Efficiency of coded CDMA (1/3)

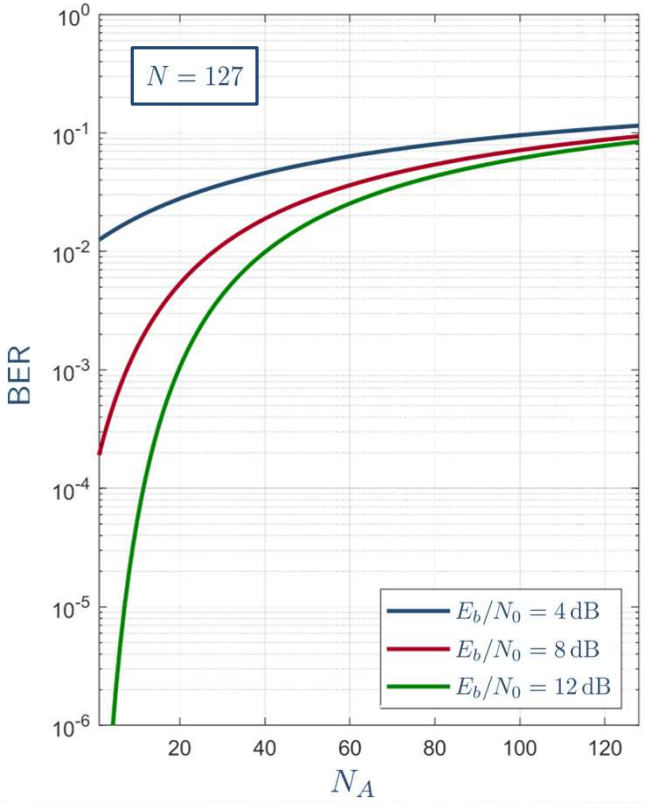
The impact of MAI is a very limiting factor for the efficiency of a CDMA-based system

We can improve the efficiency performance by adopting channel coding techniques

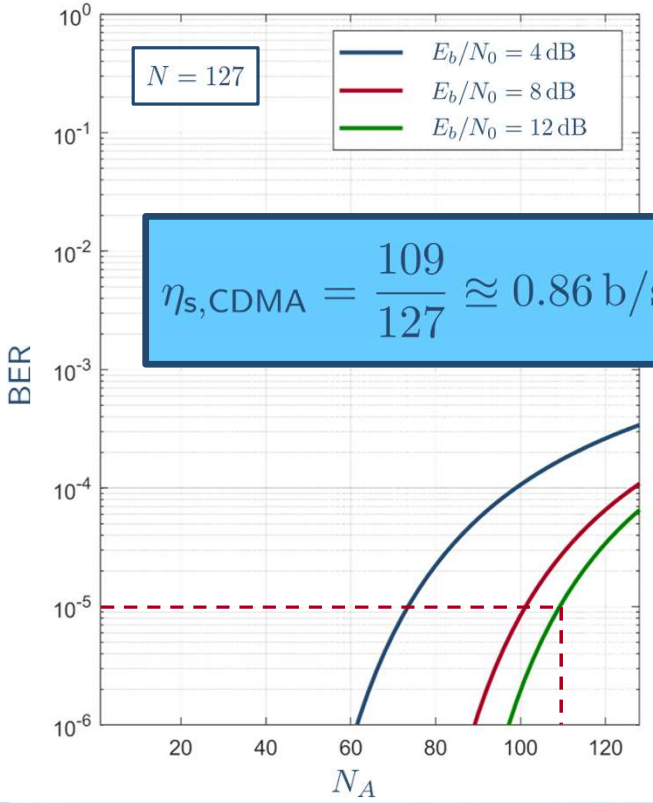
Example: low-density parity check (LDPC) code with rate 1/2, keeping the chip time equal to the uncoded case (and hence the used bandwidth)

Efficiency of coded CDMA (2/3)

uncoded case



coded case



$$\eta_{\mathrm{s,CDMA}} = \frac{109}{127} \approx 0.86\,\mathrm{b/s/Hz}$$



Efficiency of coded CDMA (3/3)

With channel coding, with CDMA we can attain efficiencies in the same order as TDMA and FDMA (i.e., systems with orthogonal multiple access) (by relaxing the QoS requirements, even $\eta_{s,\mathrm{CDMA}} > 1\,\mathrm{b/s/Hz}$ are possible!)

Please note that using channel coding with TDMA and FDMA does not further improve the efficiency, yet increasing the energy efficiency of the system given a certain QoS

Channel coding can also be applied to packet-based multiple access schemes, thus originating additional variants to the ALOHA protocol (e.g., spread-spectrum ALOHA)



Bibliography

[01] B. Sklar and F. Harris, *Digital Communications* , 3rd ed. Upper Saddle River, NJ, US: Prentice Hall, 2021.

[02] M. Luise, *Lecture notes on communication technologies.* 2022. [Online] https://docenti.ing.unipi.it/m.luise/ComTech/main_ComTech.pdf

[03] A. V. Oppenheim, A. S. Willsky, S. Hamid, *Signals and Systems*, 2nd ed. Pearson Education, 2017.

[04] J. G. Proakis, D. K. Manolakis, *Digital Signal Processing*, 5th ed. Pearson Education Limited, 2021.

[05] F. T. Ulaby, A. E. Yagle, *Signals and Systems: Theory and Applications*, Michigan Publishing, 2018. [Online] https://ss2-2e.eecs.umich.edu/

[06] R. Gonzalez, R. Woods, *Digital Image Processing*, 4th ed. Pearson Education Limited, 2018.

[07] H. Benoit, *Digital Television: Satellite, Cable, Terrestrial, IPTV, Mobile TV in the DVB Framework*, 3rd ed. Burlington, MA: Elsevier, 2008.

[08] T.S. Rappaport, *Wireless Communications: Principles and Practice,* 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.

[09] A.J. Goldsmith, *Wireless Communications*. Cambridge, UK: Cambridge Univ. Press, 2005.

[10] A.F. Molisch, *Wireless Communications*. West Sussex, UK: J. Wiley & Sons, 2005.