Telecommunications

2011

MIEET 1º ano

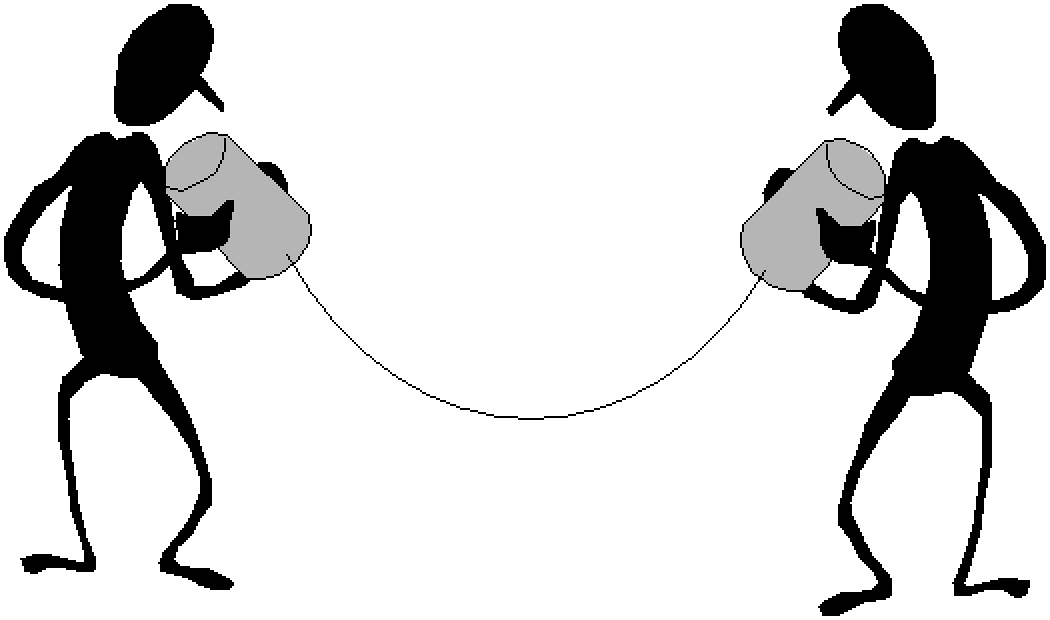
UAlg
UNIVERSIDADE DO ALGARVE

30 anos 1979 | 2009

Peter Stallinga UAlg 2011

Communication

Communication is the activity of conveying meaningful **information**.*



*: In the context of MIEET

Communication

(FYI) Official definition of 'Communication': *

communicate [k*uh*-**myoo**-ni-keyt] [?] Origin

com·mu·ni·cate [k*uh*-**myoo**-ni-keyt] [?] Show IPA ***verb,***
-cat·ed, -cat·ing.

verb (used with object)

1. to impart knowledge of; make known: *to communicate information; to communicate one's happiness.*

2. to give to another; impart; transmit: *to communicate a disease.*

3. to administer the Eucharist to.

4. *Archaic* . to share in or partake of.

verb (used without object)

5. to give or interchange thoughts, feelings, information, or the like, by writing, speaking, etc.: *They communicate with each other every day.*

6. to express thoughts, feelings, or information easily or effectively.

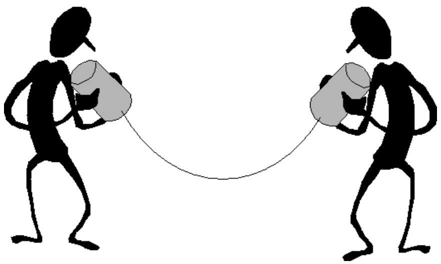
7. to be joined or connected: *The rooms communicated by means of a hallway.*

8. to partake of the Eucharist.

9. *Obsolete* . to take part or participate.

*: http://dictionary.reference.com/

Communication



- The information – the '**message**' – itself is not interesting for MIEET (If you don't like that, please transfer to the faculty of human and social sciences)

- The '**format**' and the '**channel**' of the message are very important for MIEET

- The sender and receiver can be separated in **time** and **space**

- The sender can be the same as the receiver

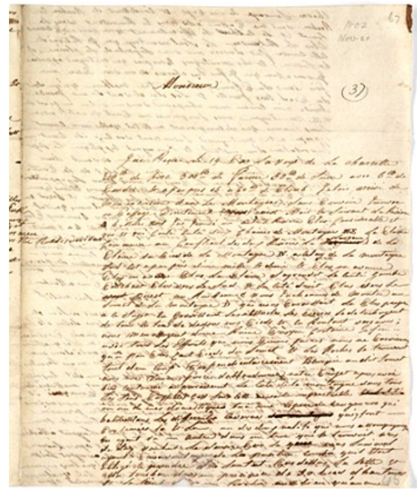
- Not necessarily persons

Oral

The oldest form of communication: **Oral**

- Same time and place. Format: acoustic. language

Written (visual)



Written language. Clay tablets / letters /books

- Different time and/or place. Format: optical. language

I can send a letter to the other end of the world. Or I can write something in a log book, to be accessed later by me or others.

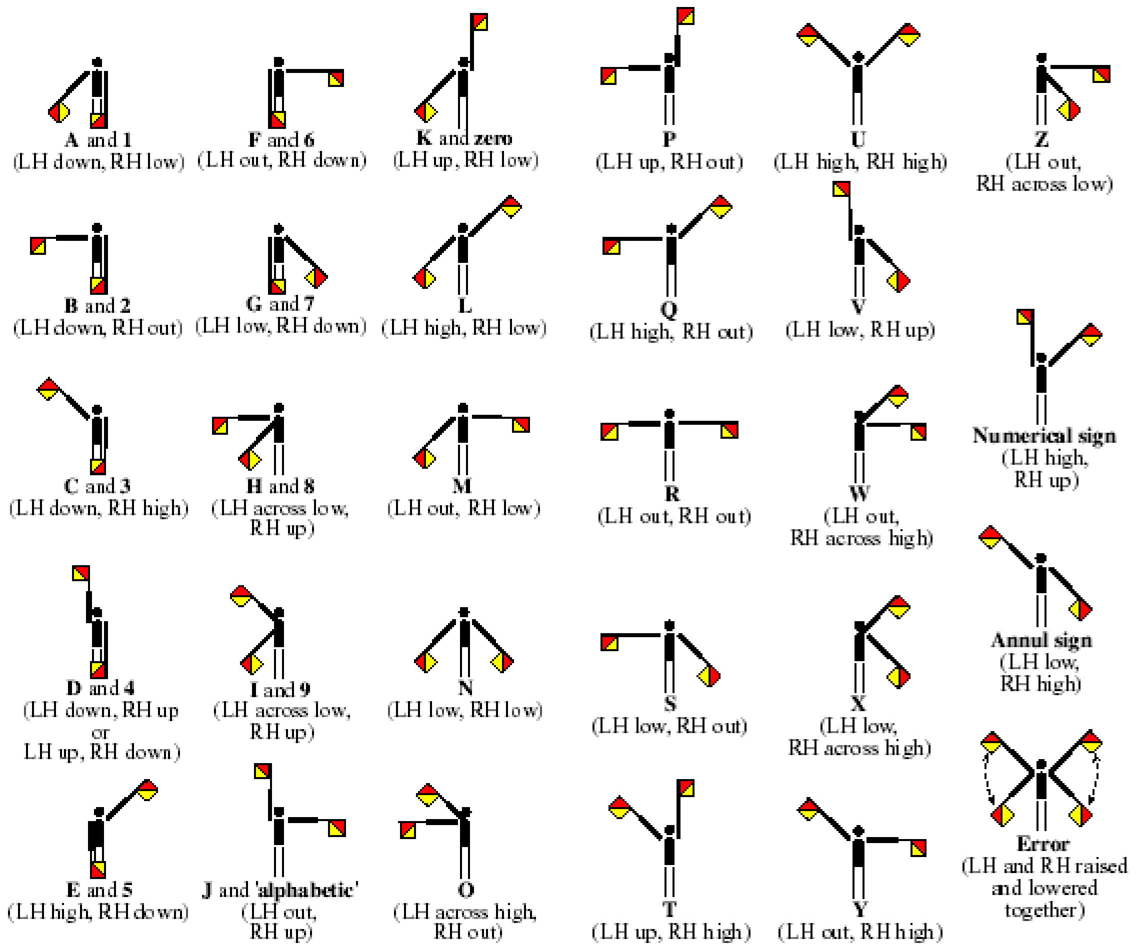
Etc.

Telecommunication

Telecommunication

- **Different** (distant) **place**, **same time**. (distant enough to not allow for direct communication)

- **Tele**phone, **tele**vision, etc.



Marine Semaphores. Smoke signals. Etc.
(communications normally win the war)

Modern Telecom: electronic

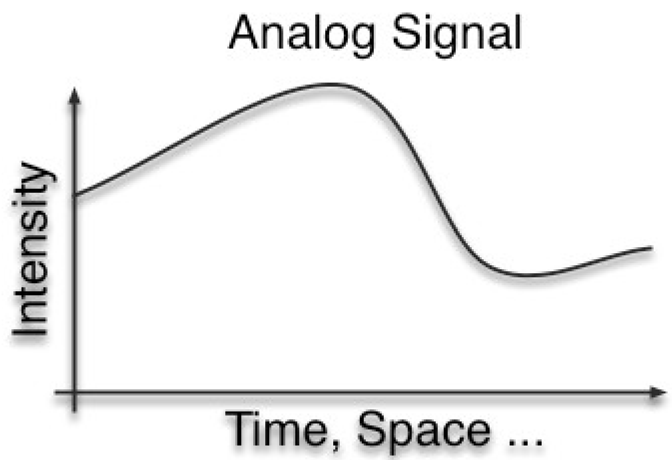
Television

Modern Telecom: electronic

Telephone

Analog

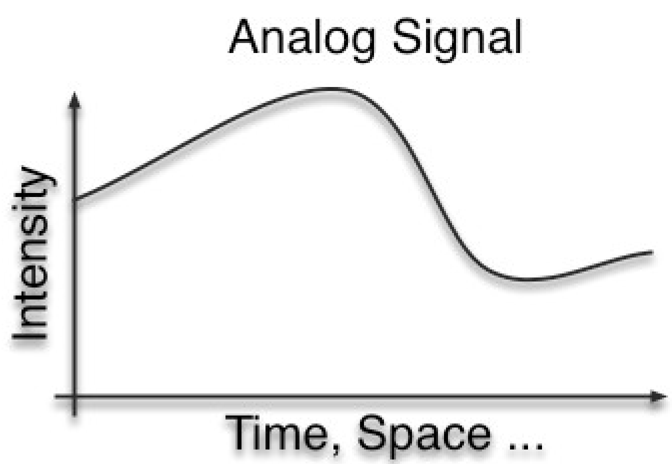
Analog: A signal that is **continuous** in **time** and **amplitude**



Signal can take **any value** at **any time** (like real numbers $\Re$)

Analog

Analog: A signal that is **continuous** in **time** and **amplitude**

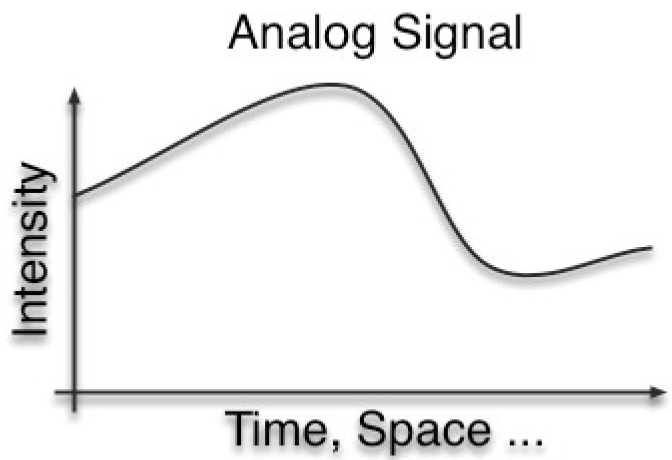


Signal can take **any value** at **any time** (like real numbers ℜ)

The information in the signal is theoretically **infinite**!
For example, I could use the convention (definition):
1.896719986783167 volt = “Peter Stallinga”
1.896719986783168 volt = “Pieter Stallinga”
1.896719986783169 volt = “Piter Stallinga”
etc.

Analog

Analog: A signal that is continuous in time and amplitude



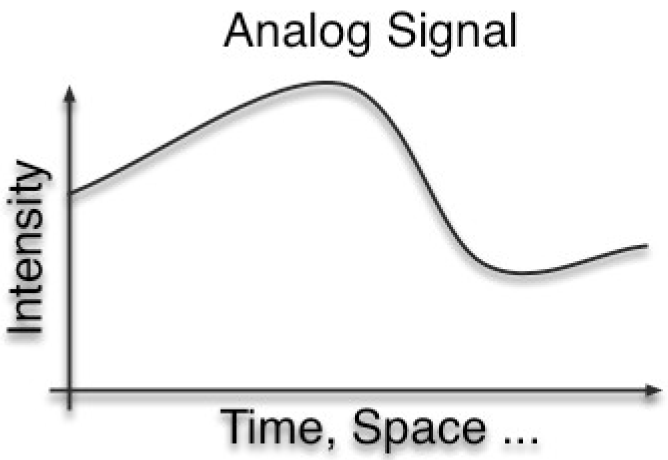
Signal can take **any value** at **any time** (like real numbers $\Re$)

at $t = 0$
1.896719986783167 volt = “Peter Stallinga”
at $t = 10^{-10}$ s
0.881763981227889 volt = “João da Silva”
at $t = 2\text{x}10^{-10}$ s
4.222719817654981 volt = “Nuno Rodrigues”

Analog is perfect!

0..5 V:
1.896719986783167 volt = “Peter Stallinga”
1.896719986783168 volt = “Pieter Stallinga”
1.896719986783169 volt = “Piter Stallinga”
$\Delta V = 10^{-15}$ V
every $\Delta t = 10^{-10}$ s

Total 5×10^{15} possibilities every 10^{-10} s



Bits and possibilities

Intermezzo:

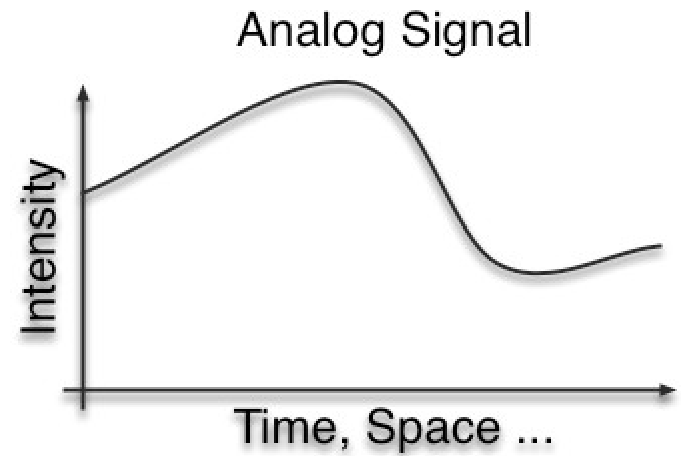
Total 5×10^{15} possibilities every 10^{-10} s
= **${}^{2}\mathrm{Log}(5 \times 10^{15})$ bits** / 10^{-10} s

1 bit of information has **2** possibilities, Ex. 0 or 1, yes/no, green/red, 0 volt/5 volt, man/woman
2 bits of information have **4** possibilities, Ex. 00, 01, 10, 11
3 bits of information have **8** possibilities, Ex. 000, 001, 010, 011, 100, 101, 110, 111
n bits have 2x2x2...x2 = **2^n** possibilities

Reverse: How many bits of information if we have x possibilities?
$m = {}^{2}\mathrm{Log}(x)$
Then $2^m = x$

Analog is perfect!

0..5 V:
1.896719986783167 volt = "Peter Stallinga"
1.896719986783168 volt = "Pieter Stallinga"
1.896719986783169 volt = "Piter Stallinga"
$\Delta V = 10^{-15}$ V
every $\Delta t = 10^{-10}$ s



Total $5\text{x}10^{15}$ possibilities every 10^{-10} s

= ${}^{2}\text{Log}(5\text{x}10^{15})$ bits / 10^{-10} s
= $5.2\text{x}10^{11}$ b/s (520 Gb/s) (Take that, Meo/Zon/Sapo! ....)

and if we want more, we just define more levels and time points
...... Analog is **without limits**!
(As long as the noise permits → Shannon [MIEET:Ftel])

Analog is better

Yes, your old record player is better than a CD player!!!
There is (potentially more information on a vinyl record)*

*: Except albums of '50 cents'; they contain no information whatsoever.

Digital

Why digital?

- Can be transmitted/copied **without** errors
- Errors can even be **corrected**

Digital / discrete

I'll send you 10 integer numbers, they are

10, 8, 9, 1, 8, 6, 3, 10, 7, 2 ← Information I am trying to transmit

Modulation → volt ↓ + noise caused by channel

10.0031, 7.9951, 8.7548, 1.0113, 8.004,
6.3456, 3.3346, 9.9981, 6.6767, 2.2981 ← How the information arrived

↓ Detection. Volt → number

10, 8, 9, 1, 8, 6, 3, 10, 7, 2 ← How it is interpreted

No errors!

Digital / discrete

I'll send you 10 integer numbers, they are

10, **8**, 9, 1, 8, 6, 3, 10, 7, 2 — Information I am trying to transmit

10.0031, 7.4121, 8.7548, 1.0113, 8.004,
6.3456, 3.3346, 9.9981, 6.6767, 2.2981 — How the information arrived

10, **7**, 9, 1, 8, 6, 3, 10, 7, 2 — How it is interpreted

Error

The noise cannot be bigger than the distance between two levels

Error detection

I'll send you 10 integer numbers, they are (together with sum)

10, 8, 9, 1, 8, 6, 3, 10, 7, 2, **sum 64** — Information I am trying to transmit

10.0031, 7.4121, 8.7548, 1.0113, 8.004,
6.3456, 3.3346, 9.9981, 6.6767, 2.2981, 63.9960 — How the information arrived

10, 7, 9, 1, 8, 6, 3, 10, 7, 2, sum 64 — How it is interpreted

sum 63 <> sum 64 !

Error. I **know** it!

By sending additional information we can detect errors

But we don't know **where** the errors are

Error correction: ARQ

In case of error, just ask for the data to be sent again

sender: 10, 8, 9, 1, 8, 6, 3, 10, 7, 2, sum 64
(sender: end-of-package, please acknowledge)

channel: 10.0031, 7.4121, 8.7548, 1.0113, 8.004, 6.3456, 3.3346, 9.9981, 6.6767, 2.2981, 63.9960

received: 10, 7, 9, 1, 8, 6, 3, 10, 7, 2, sum 64
receiver sends: Error! Not acknowledge. Please send again.

ARQ (Automatic repeat request) needs **bidirectional** communication!

Things can still go wrong
- imagine there is also error in check information, such that sum is correct
- imagine the signal is lost. We receive all 0's (sum 0)

Error correction: ARQ

Examples:

(Odd) Parity check: count number of 1's in data. Send additional 1 if even, 0 if odd
Example: 0000**1**, 10011**0**, 1111**1**

CRC (Cyclic redundancy check):
checking the remainder of the mathematical division of data.
(Parity is a special case of CRC)

Error correction: FEC (forward error correction)

FEC: Send the correction of the error anyway, before

Three bits. 1: information bit, 2: (even) parity bit, 3: correction bit (sum of I+P)

000: everything OK. No error.
Information: **0**

001: No error (parity bit is OK). Correction bit is wrong
Information: **0**

010: Parity check error. Correction bit tells us parity bit is wrong
Information: **0**

100: Parity check error. Correction bit tells us info bit is wrong
Information: **0**

011: Parity check error. Correction bit tells us info bit is wrong
Information: **1**

101: Parity check error. Correction bit tells us parity bit is wrong
Information: **1**

110: everything OK. No error.
Information: **1**

111: No error (parity bit is OK). Correction bit is wrong
Information: **1**

Error correction: FEC

Forward error correction. No request to acknowledge

FEC (forward error correction) can be used in **unidirectional** communication!

Examples:
- TDT (Digital Terrestrial Television)
- CD / DVD

Coding and transport of data

1: How the **human**-understandable information is translated into a **computer**-readable and processable signal
- (Digital/binary) **bit pattern**

2: How the digital format is translated into an **electronic** signal
- **Modulation**

3: How the electronic signal is **transported**
- Antennas, fiber-optic, cable, Wi-Fi

1: Bit pattern: ASCII

1: How the **human**-understandable information is translated into a **computer**-readable and processable signal
(Digital/binary) **bit pattern**

Example: ASCII*

- 26 letters in English alphabet
- UPPERCASE and lowercase
- Common used symbols (punctuation, math, etc.)
- Control codes (Note: ACK = acknowledge)

Total: 127 plus null character

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

*: American Standard Code for Information Interchange

1: Bit pattern: ASCII

Total: 128 characters / possibilities
Number of bits: $^{2}\text{Log}(128) = 7$
$(2^7 = 128)$

Example: 'A':
1000001 (binary)
41 (hexadecimal)*
65 (decimal)*

ASCII:
A **convention** agreeing which **character** is assigned which **bit pattern** (and value*)

First 3 bits

Last 4 bits

		000	001	010	011	100	101	110	111
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0000	**0**	NUL	DLE	space	0	@	P	`	p
0001	**1**	SOH	DC1 XON	!	1	A	Q	a	q
0010	**2**	STX	DC2	"	2	B	R	b	r
0011	**3**	ETX	DC3 XOFF	#	3	C	S	c	s
0100	**4**	EOT	DC4	$	4	D	T	d	t
0101	**5**	ENQ	NAK	%	5	E	U	e	u
0110	**6**	ACK	SYN	&	6	F	V	f	v
0111	**7**	BEL	ETB	'	7	G	W	g	w
1000	**8**	BS	CAN	(	8	H	X	h	x
1001	**9**	HT	EM	)	9	I	Y	i	y
1010	**A**	LF	SUB	*	:	J	Z	j	z
1011	**B**	VT	ESC	+	;	K	[	k	{
1100	**C**	FF	FS	,	<	L	\	l	|
1101	**D**	CR	GS	-	=	M	]	m	}
1110	**E**	SO	RS	.	>	N	^	n	~
1111	**F**	SI	US	/	?	O	_	o	del

*: See lectures on Digital Systems

1: Bit pattern: 8-bit ASCII

Modern ASCII: 8 bits

First half (00000000-01111111) standard
Second half (10000000-11111111) depending on country ('page code')

Very confusing. Sending text becomes non standard

Still one of most common problems in communications (Windows still uses MS-DOS code pages; Linux uses UTF-8)

*: See lectures on Digital Systems

1: Bit pattern: 8-bit ASCII, page code 860 (Pt)

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
8-	Ç 00C7 128	ü 00FC 129	é 00E9 130	â 00E2 131	ã 00E3 132	à 00E0 133	Á 00C1 134	ç 00E7 135	ê 00EA 136	Ê 00CA 137	è 00E8 138	Í 00CD 139	Ô 00D4 140	ì 00EC 141	Ã 00C3 142	Â 00C2 143
9-	É 00C9 144	À 00C0 145	È 00C8 146	ô 00F4 147	õ 00F5 148	ò 00F2 149	Ú 00DA 150	ù 00F9 151	Ì 00CC 152	Õ 00D5 153	Ü 00DC 154	¢ 00A2 155	£ 00A3 156	Ù 00D9 157	₧ 20A7 158	Ó 00D3 159
A-	á 00E1 160	í 00ED 161	ó 00F3 162	ú 00FA 163	ñ 00F1 164	Ñ 00D1 165	ª 00AA 166	º 00BA 167	¿ 00BF 168	Ò 00D2 169	¬ 00AC 170	½ 00BD 171	¼ 00BC 172	¡ 00A1 173	« 00AB 174	» 00BB 175
B-	░ 2591 176	▒ 2592 177	▓ 2593 178	│ 2502 179	┤ 2524 180	╡ 2561 181	╢ 2562 182	╖ 2556 183	╕ 2555 184	╣ 2563 185	║ 2551 186	╗ 2557 187	╝ 255D 188	╜ 255C 189	╛ 255B 190	┐ 2510 191
C-	└ 2514 192	┴ 2534 193	┬ 252C 194	├ 251C 195	─ 2500 196	┼ 253C 197	╞ 255E 198	╟ 255F 199	╚ 255A 200	╔ 2554 201	╩ 2569 202	╦ 2566 203	╠ 2560 204	═ 2550 205	╬ 256C 206	╧ 2567 207
D-	╨ 2568 208	╤ 2564 209	╥ 2565 210	╙ 2559 211	╘ 2558 212	╒ 2552 213	╓ 2553 214	╫ 256B 215	╪ 256A 216	┘ 2518 217	┌ 250C 218	█ 2588 219	▄ 2584 220	▌ 258C 221	▐ 2590 222	▀ 2580 223
E-	α 03B1 224	ß 00DF 225	Γ 0393 226	π 03C0 227	Σ 03A3 228	σ 03C3 229	µ 00B5 230	τ 03C4 231	Φ 03A6 232	Θ 0398 233	Ω 03A9 234	δ 03B4 235	∞ 221E 236	φ 03C6 237	ε 03B5 238	∩ 2229 239
F-	≡ 2261 240	± 00B1 241	≥ 2265 242	≤ 2264 243	⌠ 2320 244	⌡ 2321 245	÷ 00F7 246	≈ 2248 247	° 00B0 248	∙ 2219 249	· 00B7 250	√ 221A 251	ⁿ 207F 252	² 00B2 253	■ 25A0 254	NBSP 00A0 255

*: See lectures on Digital Systems

1: Bit pattern: 8-bit ASCII, page code 437 (Int)

8-	Ç 00C7 128	ü 00FC 129	é 00E9 130	â 00E2 131	ä 00E4 132	à 00E0 133	å 00E5 134	ç 00E7 135	ê 00EA 136	ë 00EB 137	è 00E8 138	ï 00EF 139	î 00EE 140	ì 00EC 141	Ä 00C4 142	Å 00C5 143
9-	É 00C9 144	æ 00E6 145	Æ 00C6 146	ô 00F4 147	ö 00F6 148	ò 00F2 149	û 00FB 150	ù 00F9 151	ÿ 00FF 152	Ö 00D6 153	Ü 00DC 154	¢ 00A2 155	£ 00A3 156	¥ 00A5 157	₧ 20A7 158	ƒ 0192 159
A-	á 00E1 160	í 00ED 161	ó 00F3 162	ú 00FA 163	ñ 00F1 164	Ñ 00D1 165	ª 00AA 166	º 00BA 167	¿ 00BF 168	⌐ 2310 169	¬ 00AC 170	½ 00BD 171	¼ 00BC 172	¡ 00A1 173	« 00AB 174	» 00BB 175
B-	░ 2591 176	▒ 2592 177	▓ 2593 178	│ 2502 179	┤ 2524 180	╡ 2561 181	╢ 2562 182	╖ 2556 183	╕ 2555 184	╣ 2563 185	║ 2551 186	╗ 2557 187	╝ 255D 188	╜ 255C 189	╛ 255B 190	┐ 2510 191
C-	└ 2514 192	┴ 2534 193	┬ 252C 194	├ 251C 195	─ 2500 196	┼ 253C 197	╞ 255E 198	╟ 255F 199	╚ 255A 200	╔ 2554 201	╩ 2569 202	╦ 2566 203	╠ 2560 204	═ 2550 205	╬ 256C 206	╧ 2567 207
D-	╨ 2568 208	╤ 2564 209	╥ 2565 210	╙ 2559 211	╘ 2558 212	╒ 2552 213	╓ 2553 214	╫ 256B 215	╪ 256A 216	┘ 2518 217	┌ 250C 218	█ 2588 219	▄ 2584 220	▌ 258C 221	▐ 2590 222	▀ 2580 223
E-	α 03B1 224	ß 00DF 225	Γ 0393 226	π 03C0 227	Σ 03A3 228	σ 03C3 229	µ 00B5 230	τ 03C4 231	Φ 03A6 232	Θ 0398 233	Ω 03A9 234	δ 03B4 235	∞ 221E 236	φ 03C6 237	ε 03B5 238	∩ 2229 239
F-	≡ 2261 240	± 00B1 241	≥ 2265 242	≤ 2264 243	⌠ 2320 244	⌡ 2321 245	÷ 00F7 246	≈ 2248 247	° 00B0 248	∙ 2219 249	· 00B7 250	√ 221A 251	ⁿ 207F 252	² 00B2 253	■ 25A0 254	NBSP 00A0 255
	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F

Remember: we are sending bit patterns and not characters. How the receiver is interpreting them is just a matter of convention!

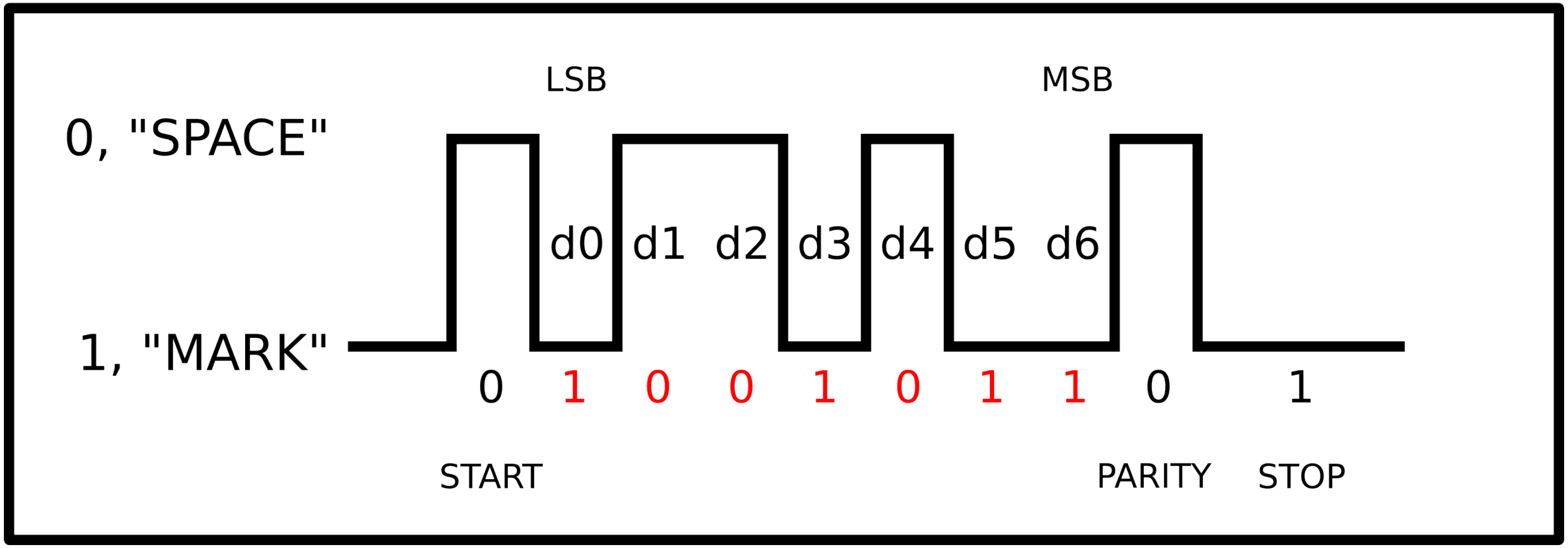
*: See lectures on Digital Systems

2: Electronic layer: RS232 (serial)

- '0', named 'SPACE': Voltage larger than +3 V
- '1', named 'MARK': Voltage below -3 V
- 7 bits sent in reverse order (LSB first)
- A start bit '0' added in the beginning
- parity bit added at end of character
- stop bit(s) '1' added at end of pattern

9600: 7E1

'i' (#105, $69): 1101001

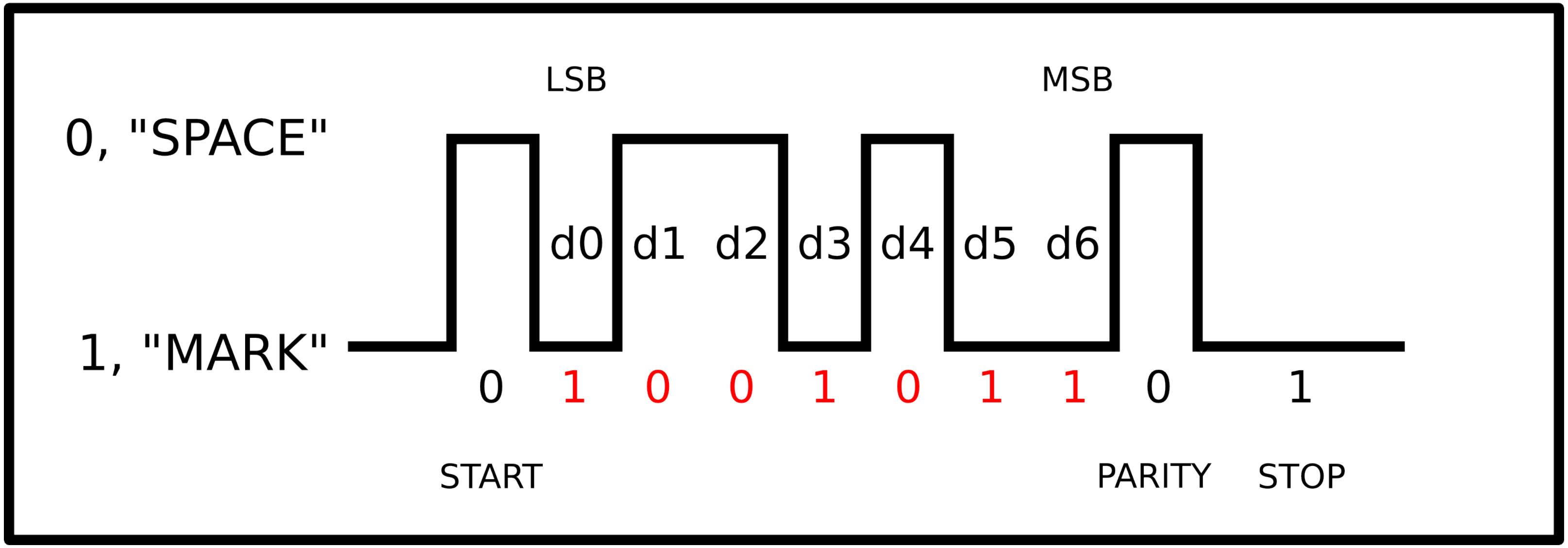


2: Electronic layer: RS232 (serial)

“9600E71”
means: 9600 bits/s (including start/stop/parity bits)
E = Even parity
7 = 7 Data bits
1 = 1 stop bit

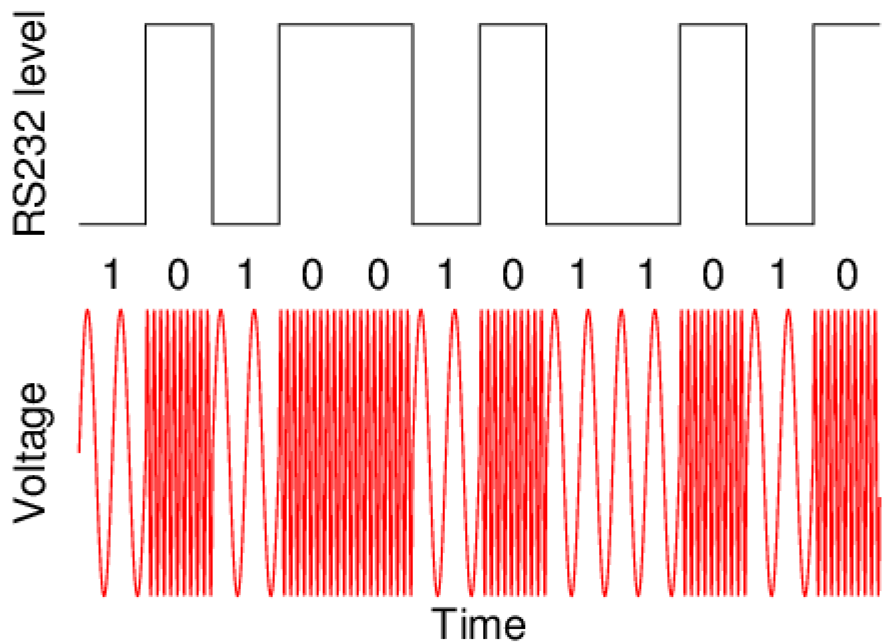
9600: 7E1

'i' (#105, $69): 1101001



2: Electronic layer: MoDem

Telephone lines are noisy. These RS232 bit patterns will arrive badly at other end (>1 km). We will modulation techniques:
Sender will **Mo**dulate it. 1 $\rightarrow$ 1 kHz, 0 $\rightarrow$ 5 kHz

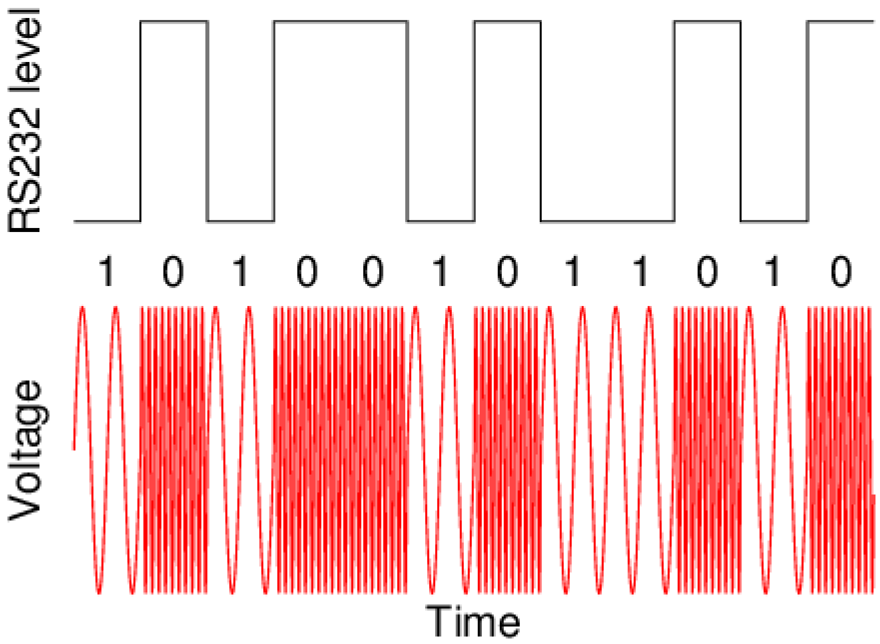


Receiver will **Dem**odulate it. 1 kHz $\rightarrow$ 1, 5 kHz $\rightarrow$ 0

MoDem: Modulator-Demodulator

MoDem: Modulator-Demodulator

2: Electronic layer: MoDem



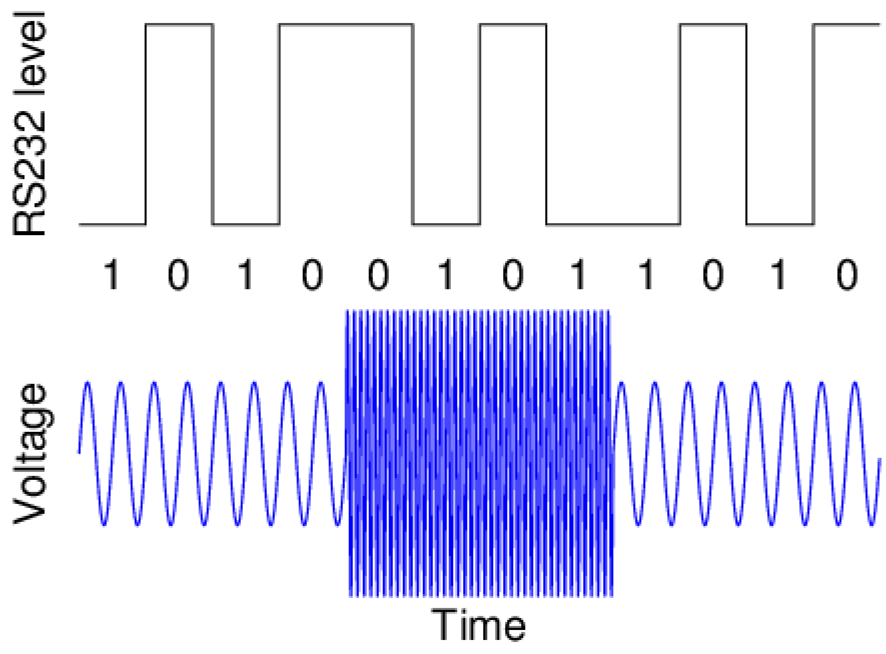
Telephone signal has bandwidth of 3 kHz
(LoFi voice has information up to that frequency)

Maximum bitrate: 2x3 kHz = 6 kb/s (Nyquist rate, $f = 2B$)

Advanced modulation techniques/ Ex. Two bits coding:

01 $\rightarrow$ 1 kHz. 12 volt,
10 $\rightarrow$ 1 kHz, 6 volt

11 $\rightarrow$ 5 kHz. 6 volt,
00 $\rightarrow$ 5 kHz, 12 volt



Maximum bitrate (Shannon-Hartley theorem): $f = B \times {}^2\mathrm{Log}(1+S/N)$

3: Transport. Cables, etc.

Earlier communication used **existing** telephone technology
Telephone landlines

- **Cheap** (because already exists)
- Very **noisy** (need for MoDem techniques)
- End of 'pay-per-call' idea (computers 24 hours connected)
- **Limited bandwidth** (telephone cut-off at 3 kHz; in simple protocol this would be 6 kb/s)

ADSL

:

:

Fiberoptics: (Nyquist rate, $f = 2B$)

Telephone (electrical) signal has bandwidth of 3 kHz

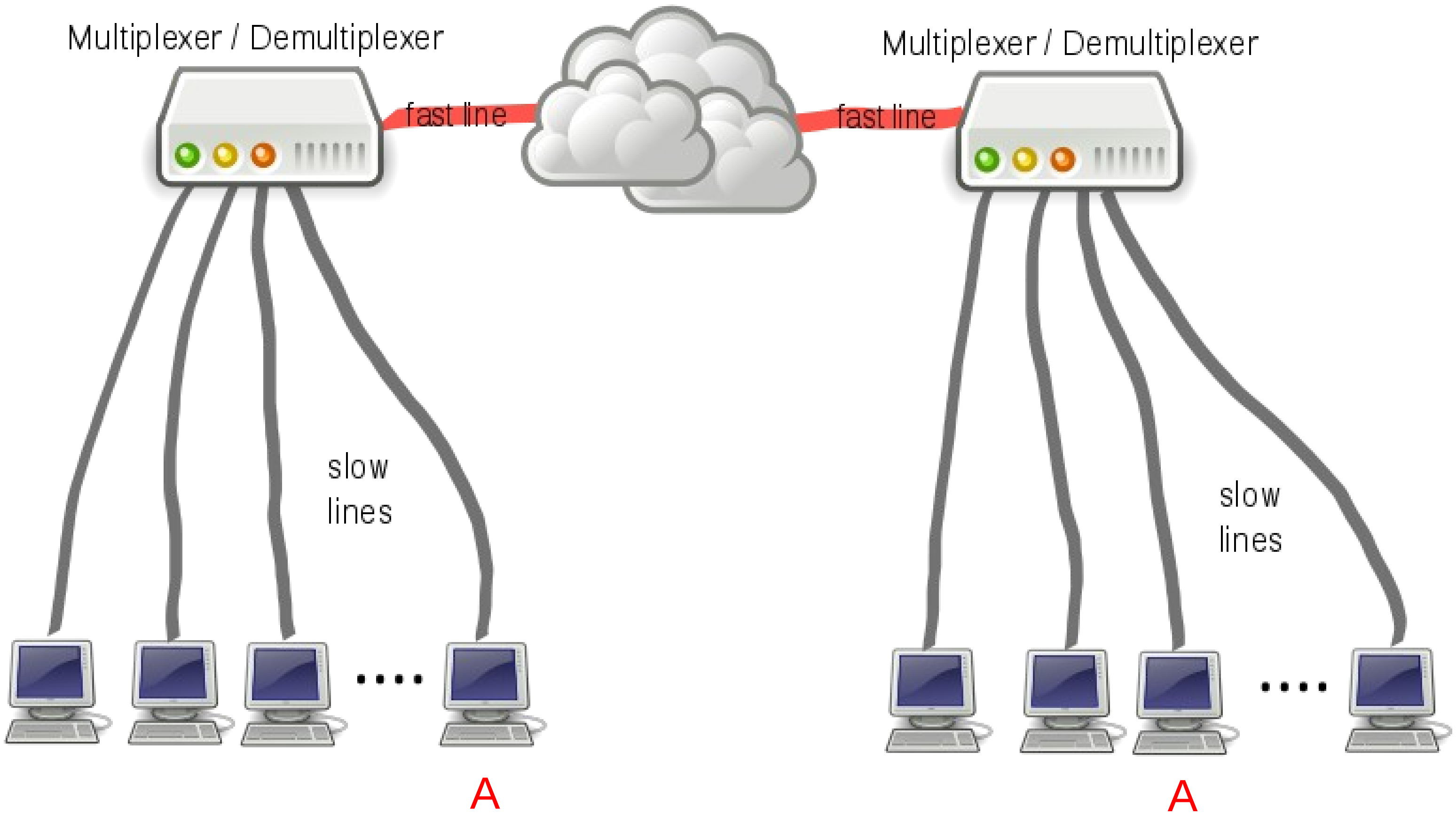
Maximum bitrate: 2x3 kHz = 6 kb/s

Fiberoptics (electromagnetic) have bandwidth of some THz (10^{12} Hz)

Maximum bitrate: 2×10^{12} Hz = 2 Tb/s

About a billion times faster. With one fiberoptics cable everybody in the world can effectively make a telephone call at the same time.

3: Transport. Multiplexing



Multiplexing: sharing a single physical medium by various sources/destinations

3: Transport. Multiplexing

Four ways of multiplexing:

SDM (Space division multiplexing)
Like audio cables. Many equipment connected to same amplifier. Not really multiplexing.
TDM (Time-division multiplexing)
Sequencing 'packets' from each individual input stream, one after the other
FDM (Frequency-division multiplexing)
Sending signals in several distinct frequency ranges over that medium.
CDM (Code-division multiplexing)
Everybody talks at the same time (like GPS satellites)

3: Transport. Multiplexing

TDM (Time-division multiplexing)
Sequencing 'packets' from each individual input stream, one after the other

FDM (Frequency-division multiplexing)
Sending signals in several distinct frequency ranges over that medium.