Voice path, PCM system, Line code
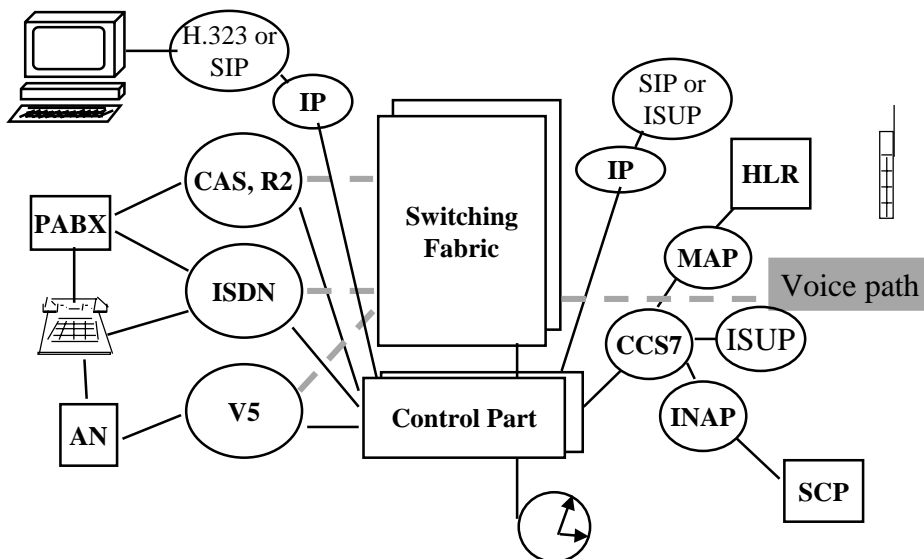
✔ **PCM ~ Pulse Code Modulation**
  › **Sampling**
  › **Quantizing**
    • **Linear**
    • **Non-linear**
  › **Quantizing error**

✔ **TDM- time division multiple access**
  › **PCM-frame structure, CRC4 -multi-frame**
  › **PCM 30, PCM 120, PCM 480, PCM 1920 (see also course s38.118)**
  › **PCM-line code**

Summary of course scope

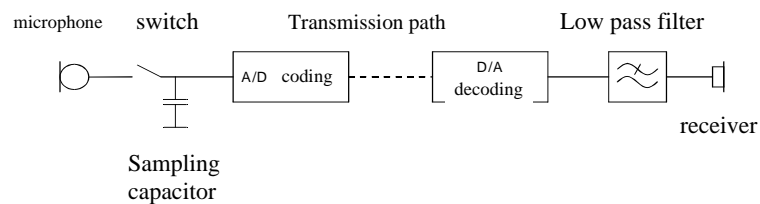## *Requirements for the Voice path and the Switching Fabric*

✔ **The Switching Fabric must understand the bits, the timeslots and the frames in the same way as the transmission systems that carry the bits**
  › **The Fabric and the transmission systems must be synchronized**

✔ **Voice must be coded efficiently** (what is efficient changes over time)

✔ **CRC -multi-frame must enable transparent transmission (= any octet values can be sent over the network freely)**

✔ **An exchange must supervise voice connections:**
  › **calls shall/should not be offered to faulty connections**
  › **calls must sometimes be cleared from faulty connections**
  › **detected faulty connections must be reported to far end if possible**

Sampling

✔ **Nyquist theorem**
  › **If an analogue signal with limited spectrum is sampled regularly with a frequency of at least twice as high as the highest frequency component, the samples carry all the information in the original signal. The original signal can be reconstructed using a low pass filter.**

✔ **In voice transmission, the spectrum carried is specified to be 300 - 3400 Hz, resulting in a minimum sampling rate of 6,8 kHz.**

✔ **In practice, since the width of the transmission channel in an analogue system is 4kHz, in a digital system a sampling rate of 8 kHz (8000 samples/s) is used.**

Digital voice transmission

✔ **The voice path includes a microphone, A/D-converter, D/A-converter and a loudspeaker.**

✔ **In practice, the analogue signal needs to be filtered before the conversion**

microphone   switch   Transmission path   Low pass filter

A/D   coding   D/A decoding

receiver

Sampling capacitor

Pulse Code Modulation - PCM

✔ **In PCM, analogue voice is digitized and thus it can be carried by digital transmission systems and switched in digital switching fabrics.**

✔ **PCM was invented in 1937 but the first real implementations became possible only using transistor technology during 1960's. This is also one of the origins of Nokia Electronics (1968) and Nokia Telecommunications.**

✔ **PCM conversion has four steps:**
  › **filtering**
  › **sampling**
  › **quantizing**
  › **coding**

Sampling of the analogue signal

- ✔ **Sampling of the analogue signal is done with a frequency of 8 kHz, I.e. with an inter-sample interval of 125 µs.**
- ✔ **The result is a PAM -signal**

125µs
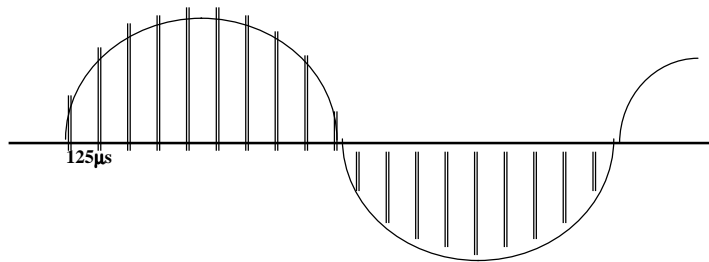
Pulse Amplitude Modulation PAM

- ✔ **Sampling produces a time discrete PAM signal reflecting the amplitude of the analogue signal.**
- ✔ **PAM-signal is quatized producing PCM-code.**
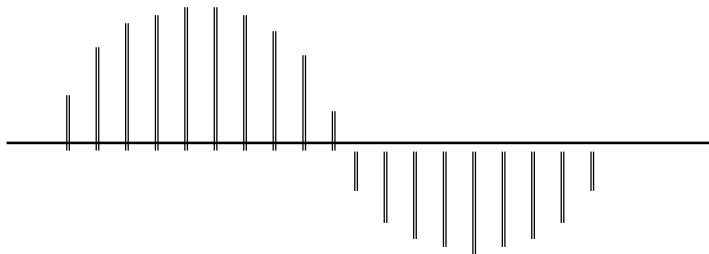
Quantizing results in approximation of the samples

✔ **Real valued amplitude figures are replaced by discrete integer values.**

✔ **Quantizing should result in values that appear in the signal with equal probability.**
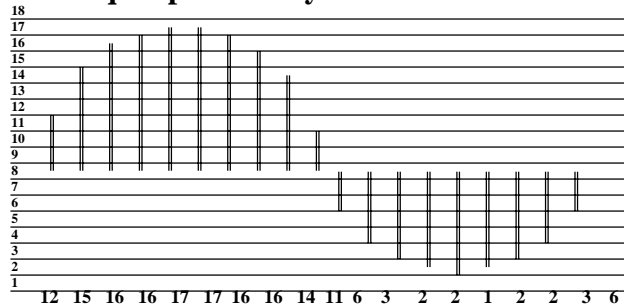
Quantizing distortion

✔ **Quantizing produces distortion, that is called quantizing distortion.**

✔ **Quantizing distortion is made by the replacement of real values by their integer approximates and at maximum can reach ½ quantizing interval.**

✔ **In linear quantizing the signal to distortion ratio is**

$$S/D = 6n + 1,8 \text{ dB} \qquad n = \text{word length}$$

12

11   ↕ Quantizing error

10

9
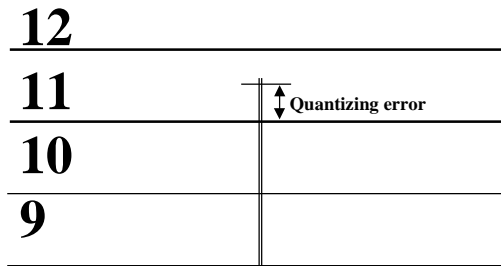
Linear vs. non-linear

✔ **The result of quantizing should use signal values with equal probability.**

✔ **This results in minimization of distortion, because a larger number of discrete signal values falls into the most typical analogue signal value area.**

✔ **In a voice signal, small analogue values appear with higher probability than larger values.**

--> non-linear quantizing

Non-linearity

✔ **Non-linear conversion can be implemented in two ways:**
  › using non-linear quantizing
  › using compression before linear quatizing is applied

✔ **Non-linear quantizing can be implemented e.g. using a network of resistors, compression requires a non-linear amplifier.**

✔ **Irrespective of the method of implementation, the non-linear quantizing follows a conversion function giving the mapping of analogue signal values to integers.**
  › In Europe (ETSI) A-function
  › In USA (ANSI) μ-function

PCM-coding and quantizing

✔ **Accoding to ETSI specification, voice coding uses 8 bits per sample.**
  › **bit-1 gives the polarity of the signal**
  › **bits 2-4 give the segment of the non-linear quantizing**
  › **bits 5-8 give the value of the discrete signal inside the segment**

✔ **Non-linearity follows the so called A -law**

$$\left| \frac{A|x|}{1+\ln(A)} \right|, 0 \le |x| \le \frac{1}{A}$$

$$\left| \frac{1+\ln|Ax|}{1+\ln(A)} \right|, \frac{1}{A} \le |x| \le 1$$

The value of A is 87,6.

Quantizing according to the A-law

Page 7
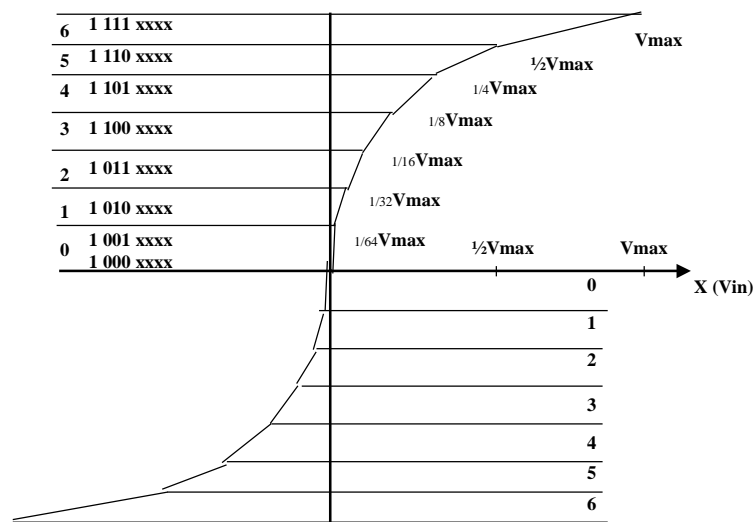
Quantizing inside a Segment

✔ **In a segment quantizing is linear**

1 101 1111

1 101 0000
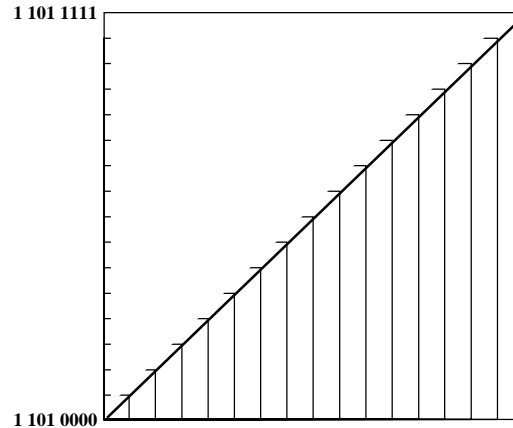
Linear vs non-linear quantizing

✔ **Linear and non-linear quantizing can be compared using the gain in signal resolution by non-linearity.**

✔ **Non-linear quantizing emphasizes small signal values, for which a gain in resolution of 24 dB is achieved.**

$$G_{dB} = 20 \log V_{in}/V_{comp}$$

PCM-hierarchy

✔ **PCM-hierarchy is created by overlapping time division multiplexed signal connections bit by bit. Bits become shorter.**

✔ **The basic speed in the hierarchy is the bitrate of a single voice channel**

$$S=8000Hz* 8bit = 64kbit/s$$

✔ **The following voice channel groups are difined**
  › **30 voice channels**
  › **120 voice channels**
  › **480 voice channels**
  › **1920 voice channels**

PCM 30 (E1)

✔ **The most common information switching and transmission format in the telecommunication network is PCM 30.**

✔ **PCM 30 contains:**
  › **1 synchronization and management channel**
  › **1 signaling channel**
  › **30 voice channel**

✔ **A channel is a time slot in the PCM-frame (125μs), created by TD multiplexing.**

✔ **PCM 30 system carries 32 time slots, each 64kbit/s. This gives a total bit rate of 2048kbit/s.**

PCM 30 frame

✔ **PCM 30 -frame contains 32 time slots**
  › **time slot 0 is dedicated for synchronization and management information**
  › **Time slot 16 is assigned for signaling information (CAS)**
  › **Time slots 1-15 and 17-32 are voice or user information channels**

✔ **Even and odd frame structures differ**
  › **In even numbered frames time slot 0 carries the frame alignment signal (C0 01 10 11). C is the CRC-bit (cyclic redundancy check) for ensuring the frame alignment recovery in case someone is sending X0 01 10 11 on a user information channel.**
  › **Time slot 0 in odd frames carries alarm information. To avoid wrong frame alignment, the second bit in tsl 0 is set to the constant value of 1.**

The use of PCM time slots in the Finnish CCS#7 network

Voice or user information channels

2 - 31

CCS#7 signaling channel   1

PCM-alarms, frame alignment   0
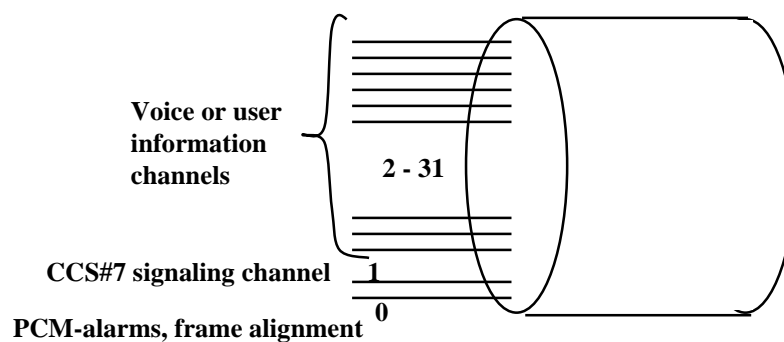
Nowadays, tsl 16 is used for voice!

Even numbered PCM 30 -frame

1 multi-frame = 16 frames

K0	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	

1 frame = 32 time slots (even frame)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27	T28	T29	T30	T31

KL — Voice channels 1 - 15 — MA — Voice channels 16 - 30

Frame alignment
time slot T0

B1	B2	B3	B4	B5	B6	B7	B8
C	0	0	1	1	0	1	1

7 bits for alignment
in even frames

CRC -bit

Signaling
time slot T16

B1	B2	B3	B4	B5	B6	B7	B8
0	0	0	0	1	A	1	1

Multi-frame
alignment
in frame 0

Multi-frame
alarm

Voice channel 26
time slot T27

B1	B2	B3	B4	B5	B6	B7	B8

Voice Sample
amplitude value

polarity

Applies only to K0, other
even numbered, look at the previous slide

PCM-frame structure (odd frame)

1 multi-frame = 16 frames

K0	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	

1 frame = 32 time slots (odd frame)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27	T28	T29	T30	T31

KL — Voice channels 1 - 15 — MA — Voice channels 16 - 30

Frame alignment
time slot T0

B1	B2	B3	B4	B5	B6	B7	B8
C	1	A	D	D	D	D	D

Data bits for mgt

CRC -bit

Far end alarm

Signaling
time slot T16

B1	B2	B3	B4	B5	B6	B7	B8
a	b	c	d	a	b	c	d

Channel 1
signaling
bits

Channel 16
signaling
bits
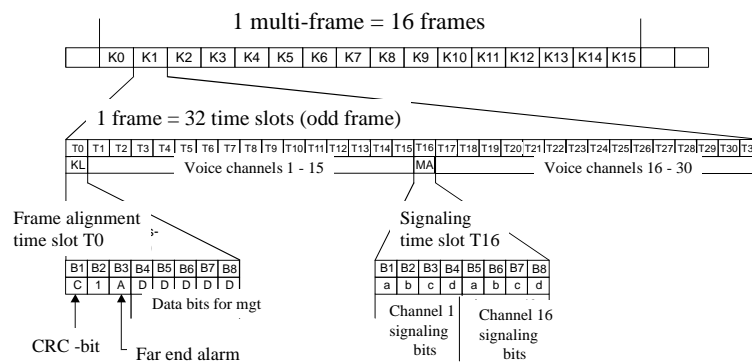
CRC-4 calculation ensures, that the frame alignement function can not lock into a user signal of (x0011011)

Tsl-0/bit x

Frame nr   t0/1   t0/2

		Frame nr		t0/1	t0/2	Tsl-0/bit x
CRC-4 multi-frame	I-half	0	C1	0	0	1 1 0 1 1
		1	0	1	A	
		2	C2	0	0	
		3	0	1	A	
		4	C3	0	0	
		5	1	1	A	
		6	C4	0	0	
		7	0	1	A	
	II-half	8	C1	0	0	
		9	1	1	A	
		10	C2	0	0	
		11	1	1	A	
		12	C3	0	0	
		13	E	1	A	
		14	C4	0	0	
		15	E	1	A	

C1…C4 - CRC4 -bits
E - CRC4-error bits
001011 - CRC4 -multi-frame
   alignment
A - far-end alarm (t0-
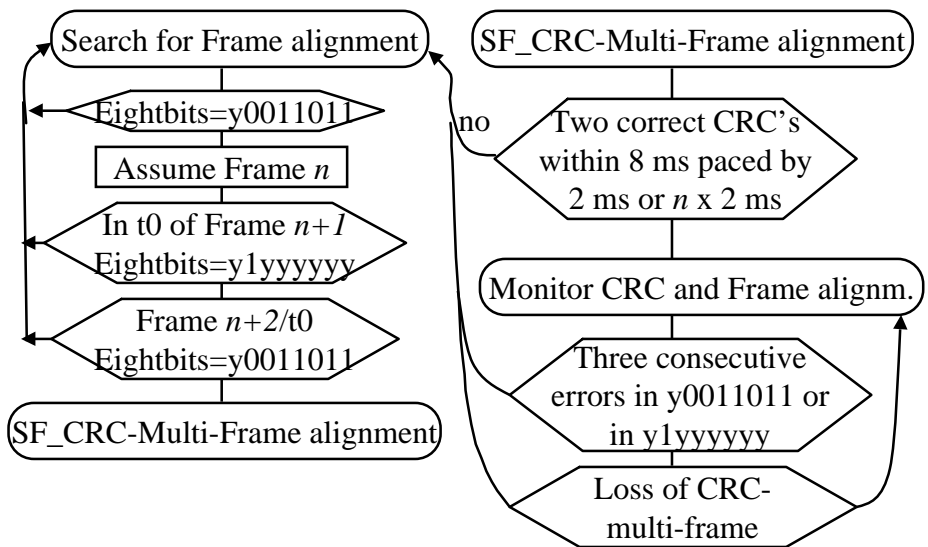   frame alignment lost)

E1 Frame alignement algorithm/ G.706

Search for Frame alignment

Eightbits=y0011011

Assume Frame n

In t0 of Frame $n+1$
Eightbits=y1yyyyyy

Frame $n+2$/t0
Eightbits=y0011011

SF_CRC-Multi-Frame alignment

SF_CRC-Multi-Frame alignment

no

Two correct CRC's within 8 ms paced by 2 ms or n x 2 ms

Monitor CRC and Frame alignm.

Three consecutive errors in y0011011 or in y1yyyyyy

Loss of CRC-multi-frame

Page 12

Higher levels in the hierarchy

✔ **PCM multiples**
  › **PCM 30 (E1) 2,048Mbit/s**
  › **PCM 120 (E2) 8,448 Mbit/s**
  › **PCM 480 (E3) 34,368 Mbit/s**
  › **PCM 1920 (E4) 139,264 Mbit/s**

✔ **Multiples are formed by multiplexing frames from four lower level connections into new higher order frames. Management overhead info is added into the higher order frame.**
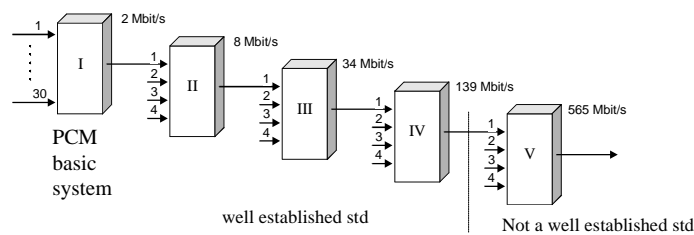
Line code in the PCM system

✔ **PCM-system uses bipolar transmission.**

✔ **Binary one is transmitted only during 50% of the cycle time.**
  › **There is no direct current that would need to be eliminated**
  › **Power spectrum is concentrated around ½ bit rate**

✔ **Two alternative line codes are used in the PCM system:**
  › **AMI - Alternate Mark Inversion**
  › **HDB3 - High Density Binary 3**

Alternate Mark Inversion - AMI

✔ **In the AMI-code**
  › **Binary one changes polarity at each occurrence**
  › **Binary zero is no signal on the line**

✔ **Weakness is in loosing bit sync in case of long series of zeroes.**

1 1 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 1
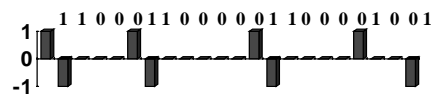
High Density Binary 3 - HDB3

✔ **In HDB3 code**
  › **Binary one changes polarity as in the AMI-code**
  › **Binary zero**
    • **First zero is replaced by "one" if in the previous group, an inversion was used**
    • **Second and third zeroes are no signal**
    • **The fourth consecutive zero raises an inversion - I.e. violation of the code, I.e. a pulse with the same polarity as the previous one or "one" is transmitted**

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

Inversion

HDB3 korvaussäännöt

Nrof B pulses after previous violation	Polarity of last B pulse	Line code	Representation of Line code	
				B - bipolar pulse
Odd	Negative (-)	000-	000V	
Odd	Positive (+)	000+	000V	
Even	Negative (-)	+00+	B00V	
Even	Positive (+)	-00-	B00V	

```
      1| 1| 0| 1| 0| 0| 0| 0| 1| 1| 0| 0| 0| 0| 0| 0| 0| 0| 1| 0|
  +V   _   _      _                        _   _              _
HDB3 0 |_|_|_____| |B 0 0 V |_____| |_____  or inverse
  -        |_|_0__0__0|V| |_|      |   |B|      |V|
   V
```

HBD3 - high density bipolar 3
V - violation
B - balance

HDBN -receiver initializes by detecting a violation and starts decoding groups of N+1 -bits

A tentative algorithm

HDB3 -receiver:

Initialization
- Interpret 0=0, +/- = 1, count pulses (odd, even)
- Until an AMI code violation is found in the signal = two consecutive + or - pulses,
- =>interpret odd& previous 000V = 0000, even& prev. B00V =0000
  =>next bit starts a groups of 4 bits

- count the nrof pulses
  - 0 --> interpret as 0
  - + --> interpret as 1   (increment nrof-pulses)
  - - --> interpret as 1   (increment nrof-pulses)
- nrof pulses = odd & 000V
  interpret = 0000
- nrof-pulses=even & B00V
  interpret = 0000