Transmission Principles

Serialization, Bit synchronization, Framing, Error Checking
Physical Aspects of Transmission, Modem

Agenda

- **Introduction**
- **Bit synchronization**
  - asynchronous
  - synchronous
- **Frame synchronization**
  - framing
  - byte stuffing
  - bit stuffing
- **Frame protection**
- **Physical aspects**

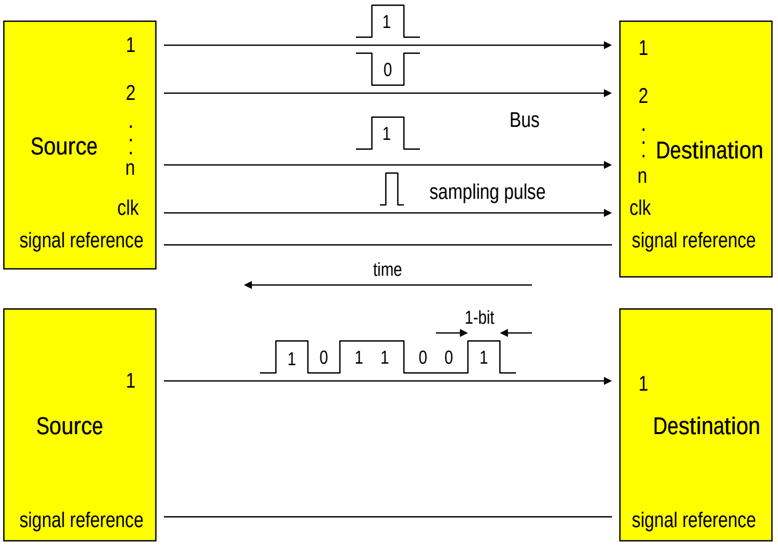
Representation of Information

- **information is stored, processed and exchanged by computer systems in binary form**
  - bit (binary digit)
  - values 0 or 1
- **these values are physically represented**
  - electrical transmission systems (using copper e.g. coax-, twisted-pair cables)
    - voltage level
    - current level
  - optical transmission systems (using fiber e.g. multi-mode, single-mode fiber)
    - light on / off

Transmission of Information

- **within a computer system**
  - parallel transfer mode
  - a data word (8-bit, 16-bit, …) is transferred at the same time using several parallel lines called “Bus”
    - data-bus for transferring data words
    - address-bus for addressing memory location
    - control-bus for signaling direction of transfer (read/write), clock (clk.), interrupt, …
- **between computer systems**
  - bit-serial transmission
  - bits are transferred bit by bit using a single line
  - basic transmission technique used in data communication networks

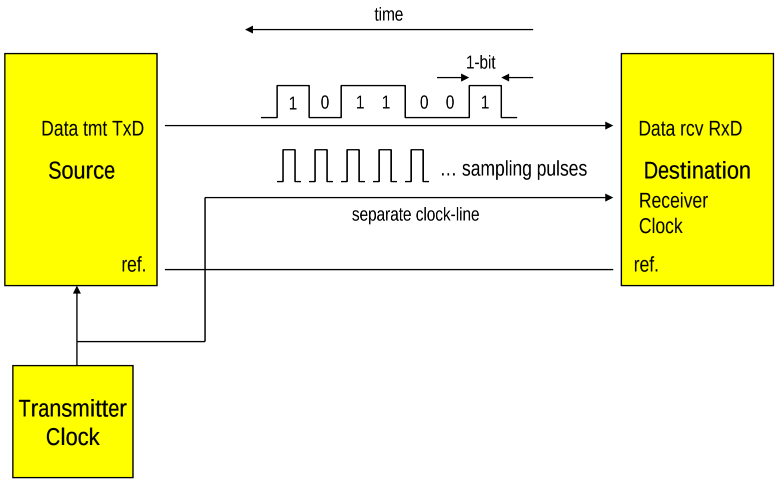
Parallel versus Serial



Serial Transmission

- **what does <u>serial</u> transmission mean?**
  - bits are transmitted on one physical line a single bit at a time using a constant time interval (bit-cell) for each bit
  - the receiver of a serial transmission line must sample bits at the right time in order to interpret the bit pattern correctly
  - receiver clock must be synchronized to transmitter clock
  - one way is to use a separate clock line as it is done by parallel transmission technique
  - in case of WAN a separate clock line is not acceptable for reasons of cost
  - therefore so called bit (clock) synchronization techniques are used

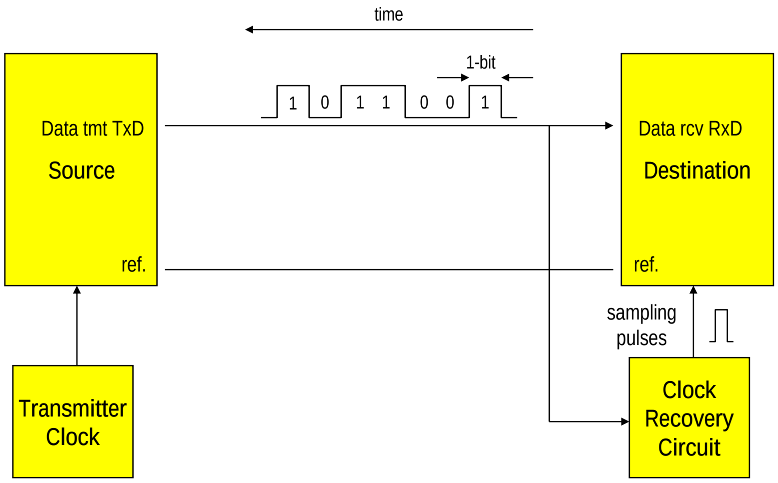
Separate Clock-Line ?



Bit Synchronization

- **clock synchronization of receiver clock for serial transmission is called**
  - bit synchronization
- **bit synchronization principle**
  - signal changes are used by the receiver for clock recovery
  - recovered clock generate pulses which are used to sample the bit stream to decide if 0 or 1
  - sampling should occur in the center of bit-cell
    - because signal attenuation, bandwidth limitation, delay distortion will modify signal form
- **depending on duration of bit synchronization we can differentiate between**
  - asynchronous and synchronous transmission method

Clock (Bit) Synchronization



time

1-bit

1 0 1 1 0 0 1

Data tmt TxD

Source

ref.

Data rcv RxD

Destination

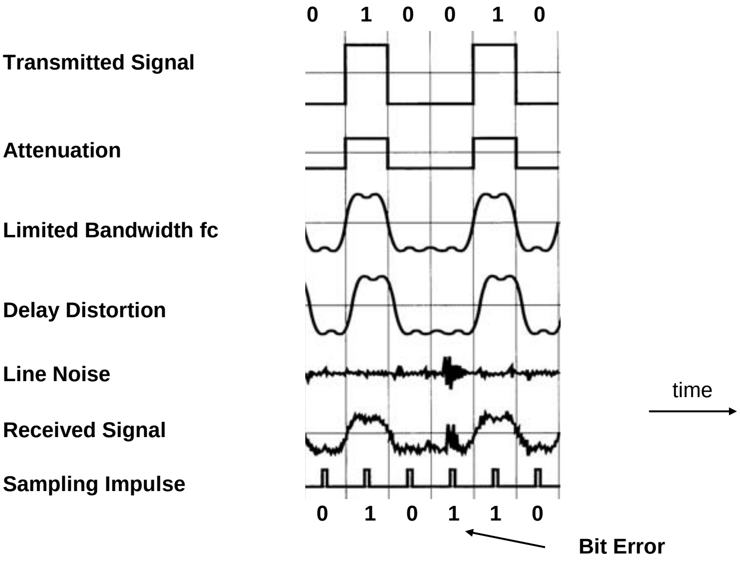
ref.

sampling pulses

Transmitter Clock

Clock Recovery Circuit

What Happens to a Signal?



0 1 0 0 1 0

Transmitted Signal

Attenuation

Limited Bandwidth fc

Delay Distortion

Line Noise

Received Signal

Sampling Impulse

0 1 0 1 1 0

Bit Error

time

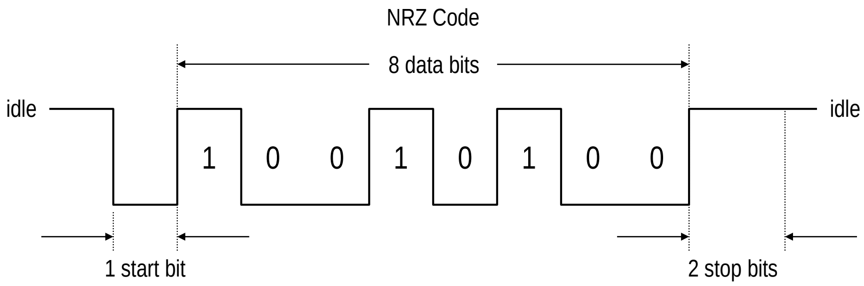
Agenda

- **Introduction**
- **Bit synchronization**
  - asynchronous
  - synchronous
- **Frame synchronization**
  - framing
  - byte stuffing
  - bit stuffing
- **Frame protection**
- **Physical aspects**

Asynchronous Transmission

- **bit synchronization lasts only for the time needed to transmit one data word**
- **data words could be sent independently and are synchronized independently from each other**
- **technique of start/stop bit is used**
  - start bit
    - indicated by a binary change from 1 to 0
    - synchronizes the following 8-bit data word by over sampling
  - stop bit(s)
    - one or two bits being binary 1
    - makes sure that every following start bit is recognized correctly regardless of the transmitted data

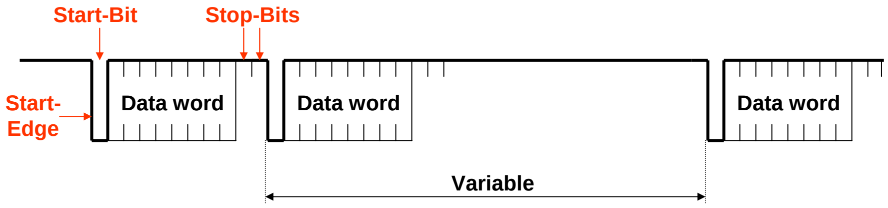
Data Word Framing by Start / Stop Bits



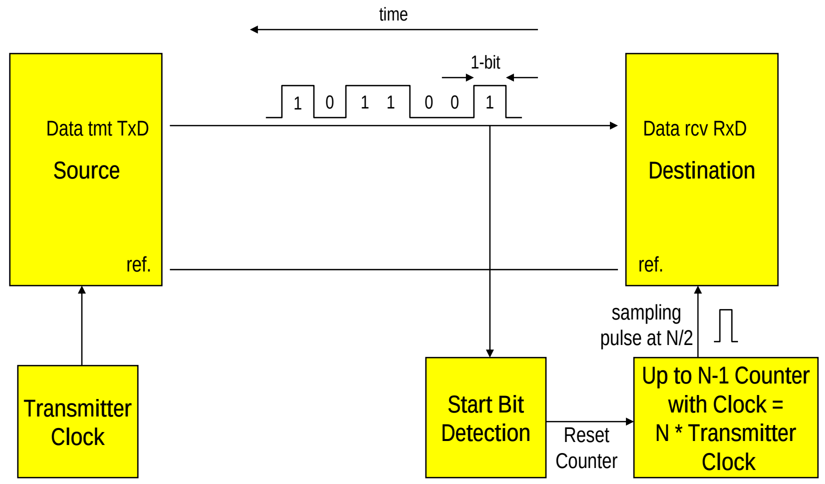
- NRZ (non return to zero) describes the encoding of bits where level 1 refers to logical 1 and level 0 refers to logical 0
- Idle .... no data is transmitted, no change of signal level

Asynchronous Transmission

- **Independent clocks at transmitter and receiver**
  - Nearly same frequency
- **Only phase is synchronized**
  - Using Start-bits and Stop-bits
  - Variable intervals between data words
  - Synchronism only during transmission of a data word
- **Inefficient**
  - 8 bits data need additional 3 bits for bit synchronization!



Bit Synchronization Circuit Asynchronous



time

1-bit

1 0 1 1 0 0 1

Data tmt TxD

Source

ref.

Transmitter Clock

Data rcv RxD

Destination

ref.

sampling pulse at N/2

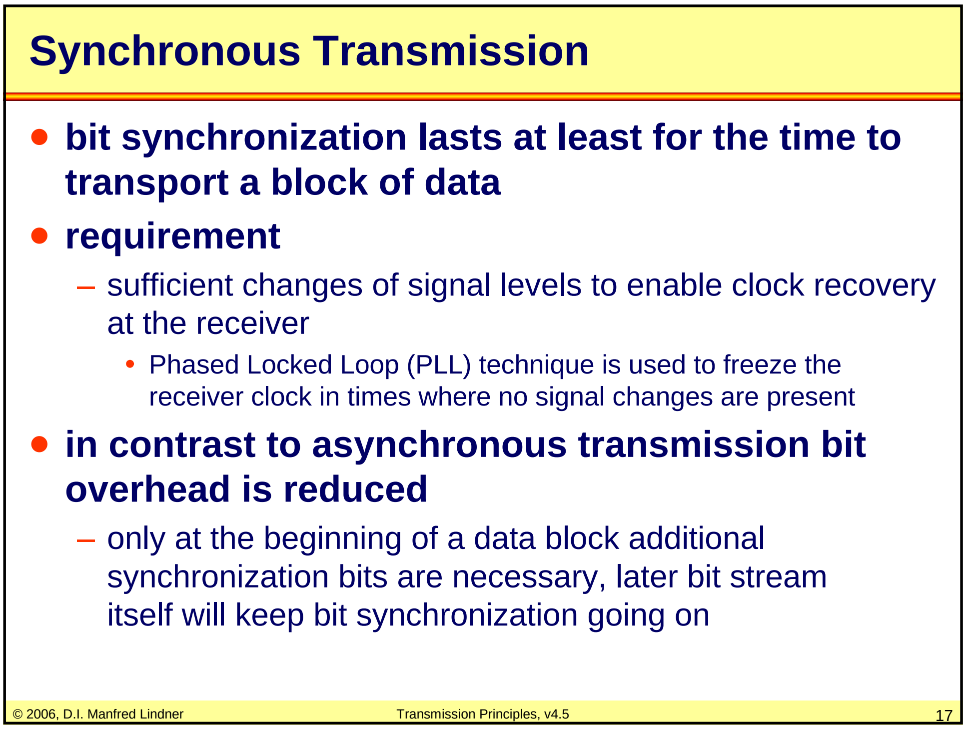
Start Bit Detection

Reset Counter

Up to N-1 Counter with Clock = N * Transmitter Clock

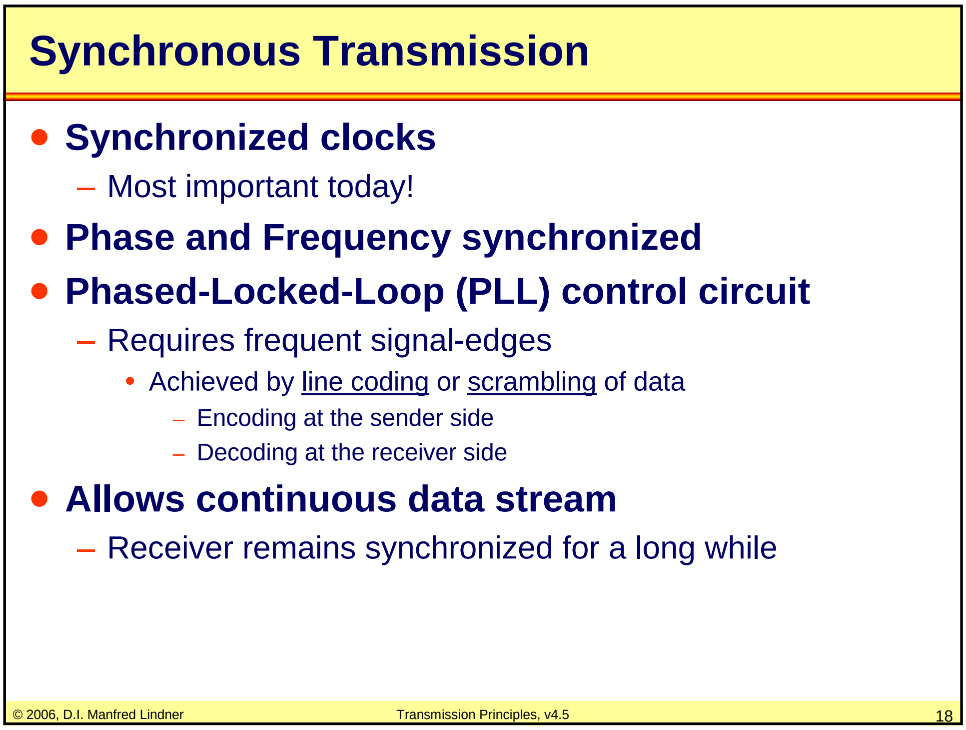
Agenda

- **Introduction**
- **Bit synchronization**
  - asynchronous
  - synchronous
- **Frame synchronization**
  - framing
  - byte stuffing
  - bit stuffing
- **Frame protection**
- **Physical aspects**



Synchronous Transmission

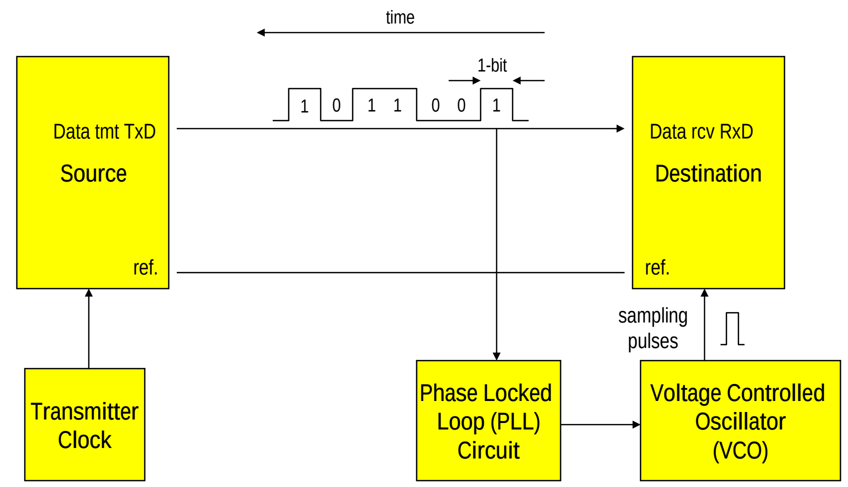
- **bit synchronization lasts at least for the time to transport a block of data**
- **requirement**
  - sufficient changes of signal levels to enable clock recovery at the receiver
    - Phased Locked Loop (PLL) technique is used to freeze the receiver clock in times where no signal changes are present
- **in contrast to asynchronous transmission bit overhead is reduced**
  - only at the beginning of a data block additional synchronization bits are necessary, later bit stream itself will keep bit synchronization going on



Synchronous Transmission

- **Synchronized clocks**
  - Most important today!
- **Phase and Frequency synchronized**
- **Phased-Locked-Loop (PLL) control circuit**
  - Requires frequent signal-edges
    - Achieved by line coding or scrambling of data
      - Encoding at the sender side
      - Decoding at the receiver side
- **Allows continuous data stream**
  - Receiver remains synchronized for a long while

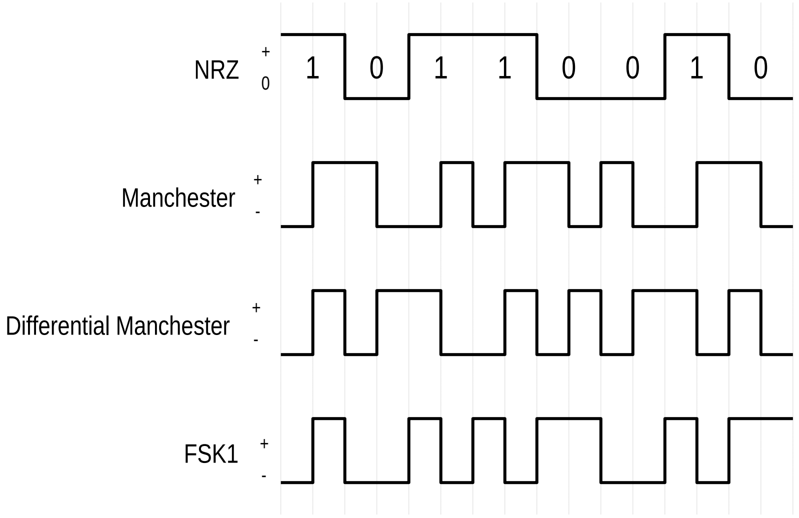
Bit Synchronization Circuit Synchronous



Synchronous Transmission

- **bit synchronization depends on sufficient signal changes within the bit stream**
  - for long series of 0s or 1s simple NRZ encoding is not able to provide this changes
- **two methods are used to guarantee signal changes**
  - encoding of bits that every bit contains a signal change
    - Manchester-code (Biphase code), Differential-Manchester-code, Frequency Shift Keying (FSK)-code, commonly used in a LANs
  - encoding of bits in such a way that there are enough signal changes in the bit stream
    - NRZI (with bitstuffing), RZ and AMI (with scrambler)
    - HDB3 (with code violations), commonly used in a WANs

Manchester-Codes, FSK-Codes



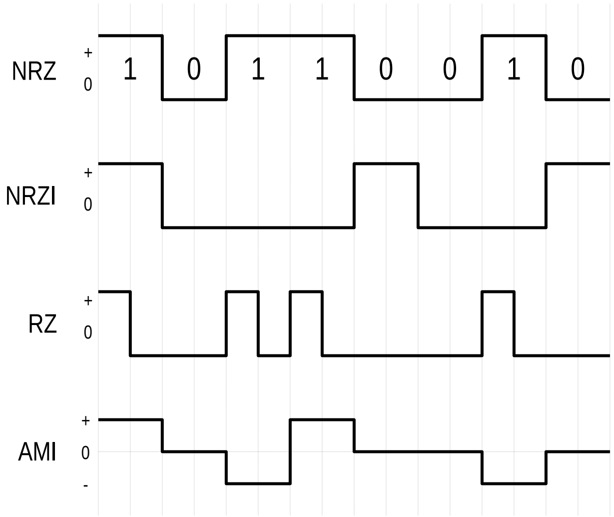
Encoding Rules For Manchester

- **Manchester**
  - bit is divided into two half-bits
  - first half-bit is the complement of the data bit, second half-bit is identical to data bit
  - change of signal level occurs in the center of each bit
    - change from 1 to 0 describes a logical 0
    - change from 0 to 1 describes a logical 1
- **differential Manchester**
  - logical 0 is defined by a signal change at the beginning and at the center of the bit
  - change of signal only at the center identifies a logical 1
  - no signal change at the center of a bit can be used for code violation (J and K symbols)

Encoding Rules for FSK

- **FSK1**
  - logical 1 is defined by a signal change at the beginning and at the center of bit
  - change of signal level only at the beginning of a bit identifies a logical 0
- **FSK0**
  - vice versa to FSK1
- **principle characteristics of Manchester and FSK codes**
  - bandwidth requirement is twice of NRZ
  - they have no or constant dc (direct current) component

NRZI, RZ, AMI Code



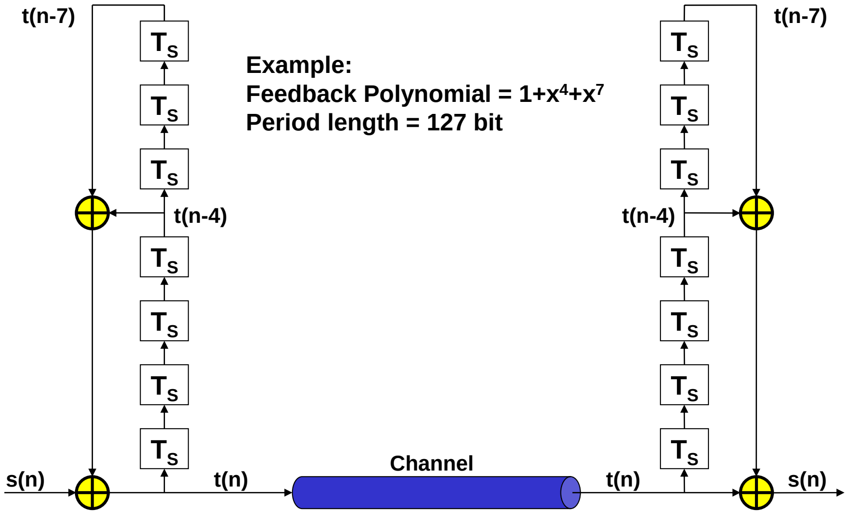
Encoding Rules for NRZI, RZ

- **NRZI (Non Return to Zero Inverted)**
  - logical 0 is defined by change of signal level at beginning of bit, logical 1 does not produce any change of signal
  - bit stuffing prevents large numbers of 1´s in bit stream
  - bandwidth requirements are identical to NRZ
  - has a dc component
- **RZ (Return to Zero)**
  - positive impulse (half bit length) describes a logical 1, logical 0 does not trigger any signal change
  - scrambler prevents large numbers of 0´s in bit stream
  - bandwidth requirements are twice of NRZ
  - has a dc component

Encoding Rules for AMI

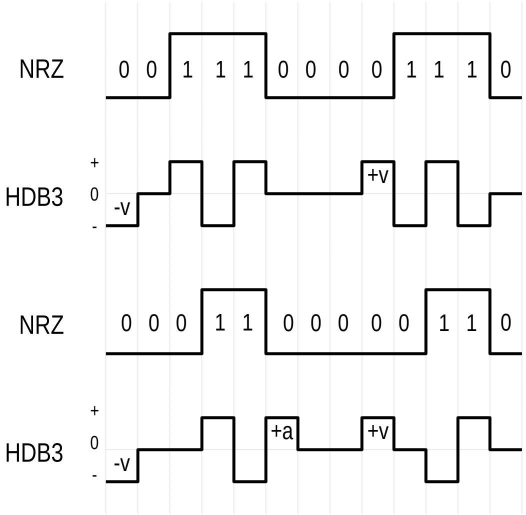
- **AMI (Alternate Mark Inversion)**
  - three level encoding (+, 0, -)
  - pulses (length = 1 bit) with changing polarity describe logical 1´s, no pulse characterizes a logical 0
  - scrambler prevents large numbers of 0´s in bit stream
  - bandwidth requirements are identical to NRZ
  - has no or constant dc component
- **NRZI, AMI used in WAN´s**

How Does a Scrambler Circuit Look Like?



Example:
Feedback Polynomial = $1+x^4+x^7$
Period length = 127 bit

HDB3 (High Density Bipolar 3) Code



Encoding Rules for HDB3

- **logical 1´s are encoded using pulses with alternate polarity, a logical 0 never generates a pulse**
- **exception for sequence of 0´s:**
  - four 0´s are encoded by a special pattern consisting of one or two impulses (A and V-bits)
  - V-bits are code violations, breaking the rule of alternating pulses
  - the following rule avoids DC portion using A- and V-bits
  - bandwidth requirements are identical to NRZ
  - has no or constant dc component

	polarity of last pulse	amount of pulses since last violation	
		odd	even
bit pattern	plus	`0 0 0 +V`	`-A 0 0 -V`
	minus	`0 0 0 -V`	`+A 0 0 +V`

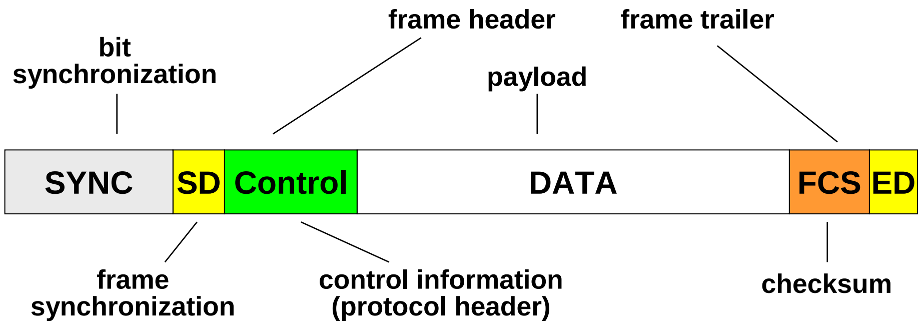
Agenda

- **Introduction**
- **Bit synchronization**
  - asynchronous
  - synchronous
- **Frame synchronization**
  - framing
  - byte stuffing
  - bit stuffing
- **Frame protection**
- **Physical aspects**

Basic Requirements

- **information between systems is exchanged in blocks of data or information frames**
- **the recognition of the beginning and the end of a block is necessary**
  - frame synchronization
- **errors on physical lines may lead to damage of digital information**
  - 0 becomes 1 and vice versa
  - the longer the block the higher the probability for an error
- **methods necessary for error checking**
  - frame protection
  - error detection

Generic Frame Format



SYNC- Sync Pattern
SD - Starting Delimiter
ED - Ending Delimiter
FCS - Frame Check Sequence

SYNC

- **SYNC is a special bit pattern**
  - used for bit synchronization after an idle period
  - can be used as fill pattern during idle times to keep the receiver clock synchronized
  - typically a 0101010...-pattern
  - e.g. 8 Byte preamble in Ethernet frames

SYNC	**SD**	**Control**	**DATA**	**FCS**	**ED**

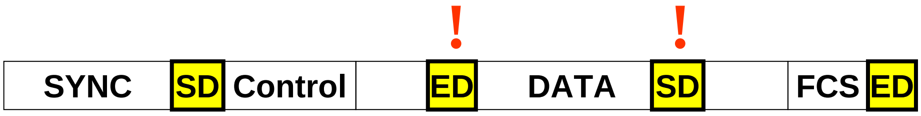
Control Field

- **is used for implementing protocol procedures**
- **contains information such as**
  - frame type, protocol type
    - Data, Ack, Nack, Connect, Disconnect, Reset, etc.
    - IP, IPX, AppleTalk, etc.
  - sequence numbers for identification of frame sequence
    - necessary for error recovery and flow control with connection oriented services
  - address information of source and destination in case of a multipoint line
  - frame length, etc.

SYNC	**SD**	**Control**	**DATA**	**FCS**	**ED**

SYNC, SD and ED

- **SD, ED are special bit patterns to mark the beginning and the end of a block**
  - not allowed inside the frame
- **What, if delimiter symbols occur within frame ?**



! !

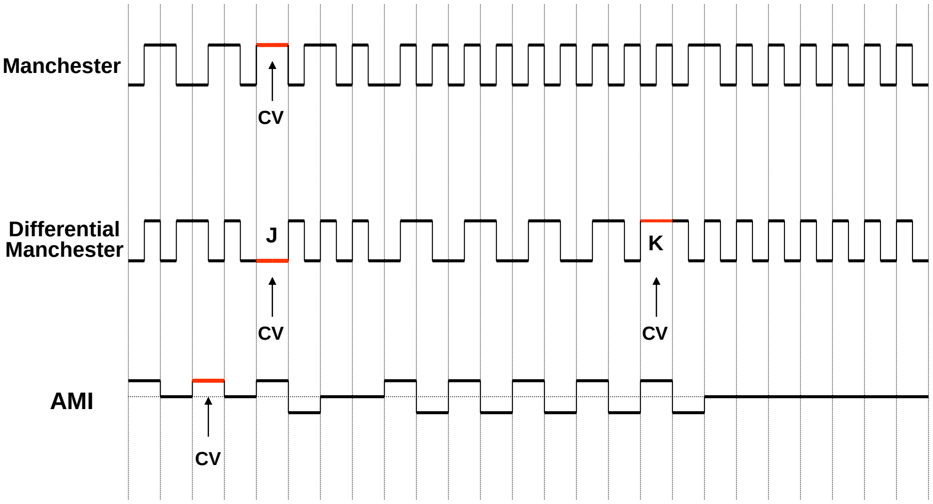
SYNC | SD | Control | ED | DATA | SD | FCS | ED

- **If application of sender must care of avoiding this bit patterns in the data stream**
  - transmission is <u>not data-transparent</u>
  - goal is data transparency

Data Transparency

- **techniques to avoid this bit pattern inside the frame**
  - byte stuffing with character based method
    - e.g. IBM BSC (Binary Synchronous Control) protocol
    - e.g. PPP over asynchronous line
  - bit stuffing with bit oriented method
    - e.g. HDLC (High level Data Link Control)
    - e.g. PPP over synchronous line
  - code violations
    - e.g. Token Ring J,K Symbols of Differential-Manchester-code
  - byte count technique
    - e.g. DDCMP (Digital Data Communications Message Protocol)
  - idle line/sync bits before special SD and idle line as ED
    - e.g. Ethernet

Code Violations

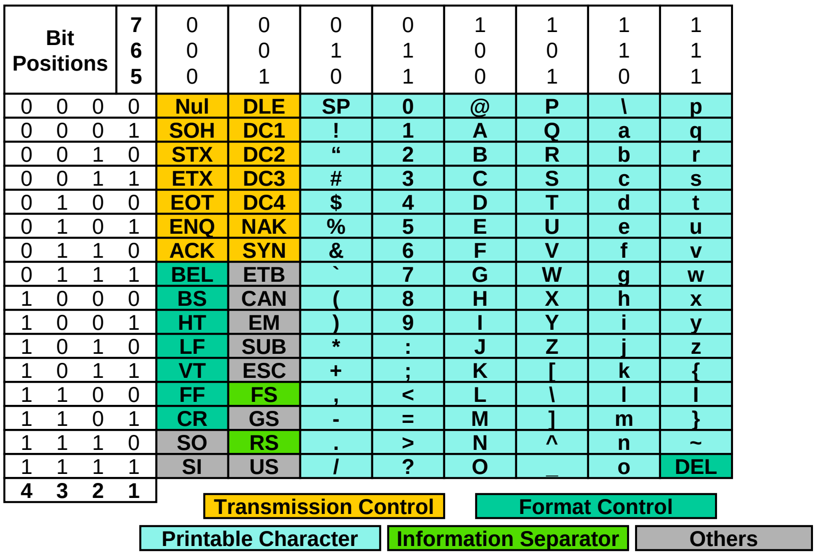


Agenda

- **Introduction**
- **Bit synchronization**
  - asynchronous
  - synchronous
- **Frame synchronization**
  - framing
  - byte stuffing
  - bit stuffing
- **Frame protection**
- **Physical aspects**

Character Based Transmission - ASCII-Code

American Standard Code for Information Interchange



Bit Positions 7 6 5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0	Nul	DLE	SP	0	@	P	\	p
0 0 0 1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	STX	DC2	“	2	B	R	b	r
0 0 1 1	ETX	DC3	#	3	C	S	c	s
0 1 0 0	EOT	DC4	$	4	D	T	d	t
0 1 0 1	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	ACK	SYN	&	6	F	V	f	v
0 1 1 1	BEL	ETB	`	7	G	W	g	w
1 0 0 0	BS	CAN	(	8	H	X	h	x
1 0 0 1	HT	EM	)	9	I	Y	i	y
1 0 1 0	LF	SUB	*	:	J	Z	j	z
1 0 1 1	VT	ESC	+	;	K	[	k	{
1 1 0 0	FF	FS	,	<	L	\	l	l
1 1 0 1	CR	GS	-	=	M	]	m	}
1 1 1 0	SO	RS	.	>	N	^	n	~
1 1 1 1	SI	US	/	?	O	_	o	DEL
4 3 2 1								

Transmission Control

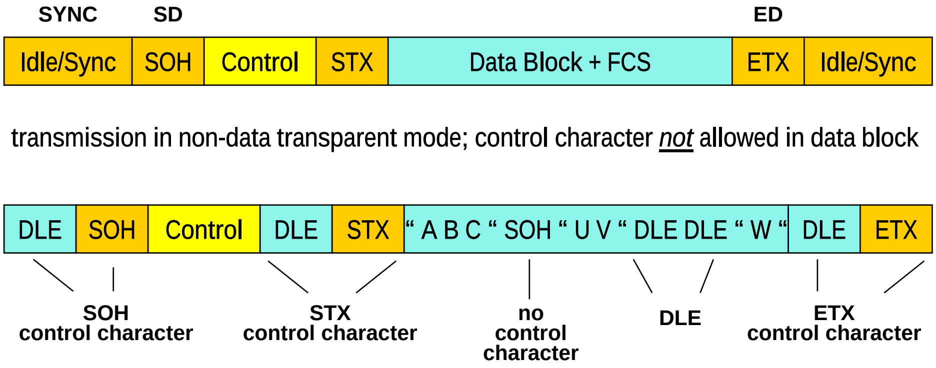
Format Control

Printable Character

Information Separator

Others

Character Based Transmission without and with Byte Stuffing



transmission in non-data transparent mode; control character *not* allowed in data block

transmission in data transparent mode with byte stuffing; control character allowed in data block

byte stuffing: DLE inside data portion will be doubled by sender; receiver deletes this doubled DLE

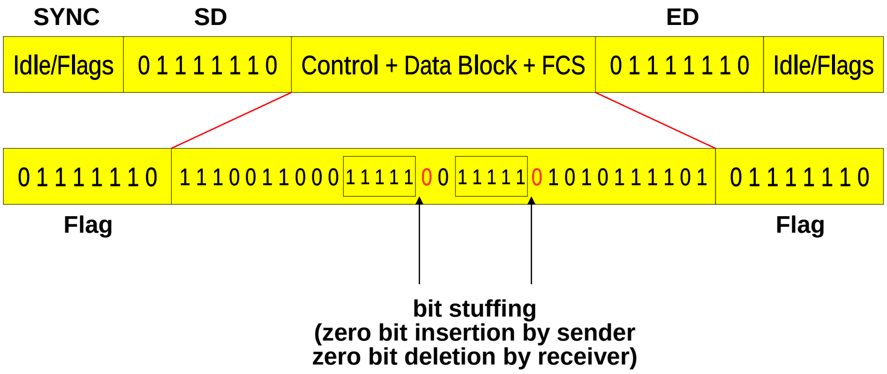
Character Based Transmission Byte Stuffing

- **the following control characters are used (ASCII, EBCDIC)**
  - SOH (Start of Header; ASCII 0x01)
  - STX (Start of Text; ASCII 0x02)
  - ETX (End of Text; ASCII 0x03)
- **not allowed inside the data portion**
  - printable characters don't contain control characters
- **no such restriction with byte stuffing**
  - control characters are only recognized as control characters with “DLE” (Data Link Escape; ASCII 0x10) in front of them
  - if “DLE” is to be transmitted as data, it will be doubled

Agenda

- **Introduction**
- **Bit synchronization**
  - asynchronous
  - synchronous
- **Frame synchronization**
  - framing
  - byte stuffing
  - bit stuffing
- **Frame protection**
- **Physical aspects**

Bit Oriented Transmission
Bit Stuffing



bit stuffing
(zero bit insertion by sender
zero bit deletion by receiver)

Bit Oriented Transmission
Bit Stuffing

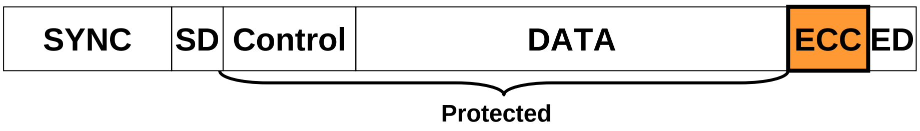
- **SD and ED equals 01111110, called flag**
  - also used for SYNC
  - any bit pattern different to flag will be interpreted as beginning of the frame
- **flag should not occur inside the frame**
  - would indicate the end of the frame
- **bit stuffing avoids the occurrence of the flag within a frame**
  - sender automatically inserts a zero after a sequence of 5 ones
  - receiver automatically deletes inserted zero bits
  - a sequence of 6 ones only occurs at the end of the frame

Agenda

- Introduction
- Bit synchronization
  - asynchronous
  - synchronous
- Frame synchronization
  - framing
  - byte stuffing
  - bit stuffing
- Frame protection
- Physical aspects

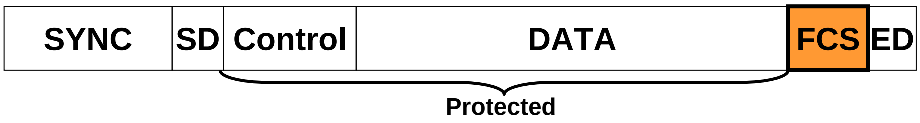
Error Correction versus Error Detection

- Two basic strategies developed by network designers
- 1. Forward Error Control
  - Include enough redundant information with each block of date to enable receiver to correct errors occurred -> error correcting codes (important -> “Hamming Distance”)
  - Required for "extreme" conditions
    - High BER (Bit Error Rate), EMR
    - Long delays, space links
  - Example: Reed-Solomon codes, Hamming-codes



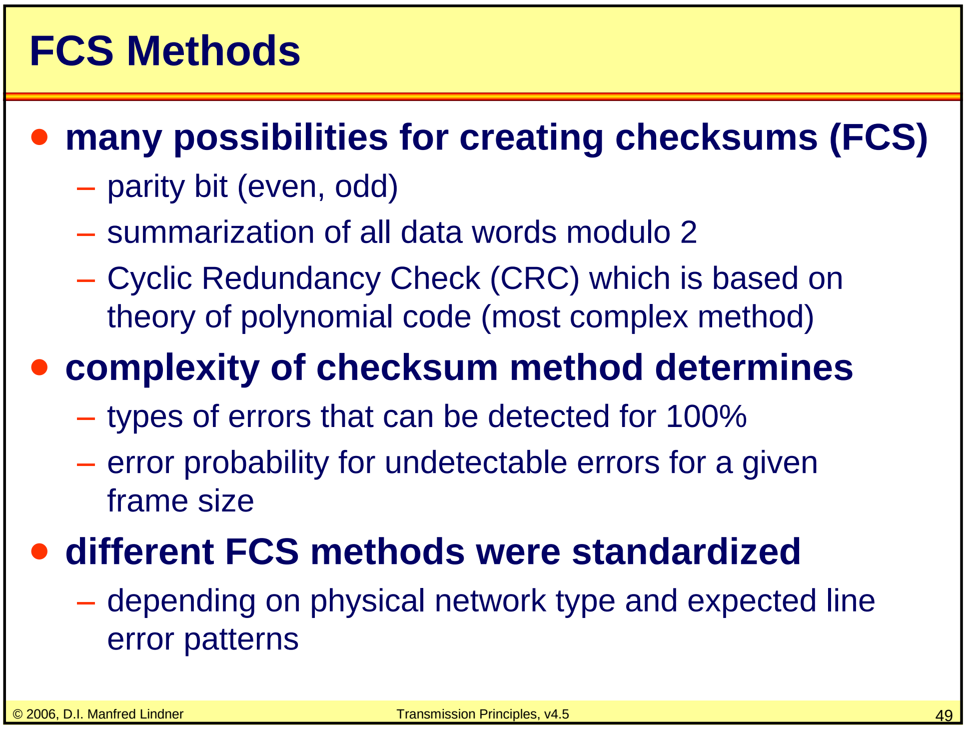
Error Correction versus Error Detection

- **2. Feedback Error Control**
  - Include enough redundant information with each block of date to enable receiver to detect only errors occurred -> error detecting codes -> Frame Check Sequence
  - After error detection a retransmission of frame is initiated through feedback to the sender



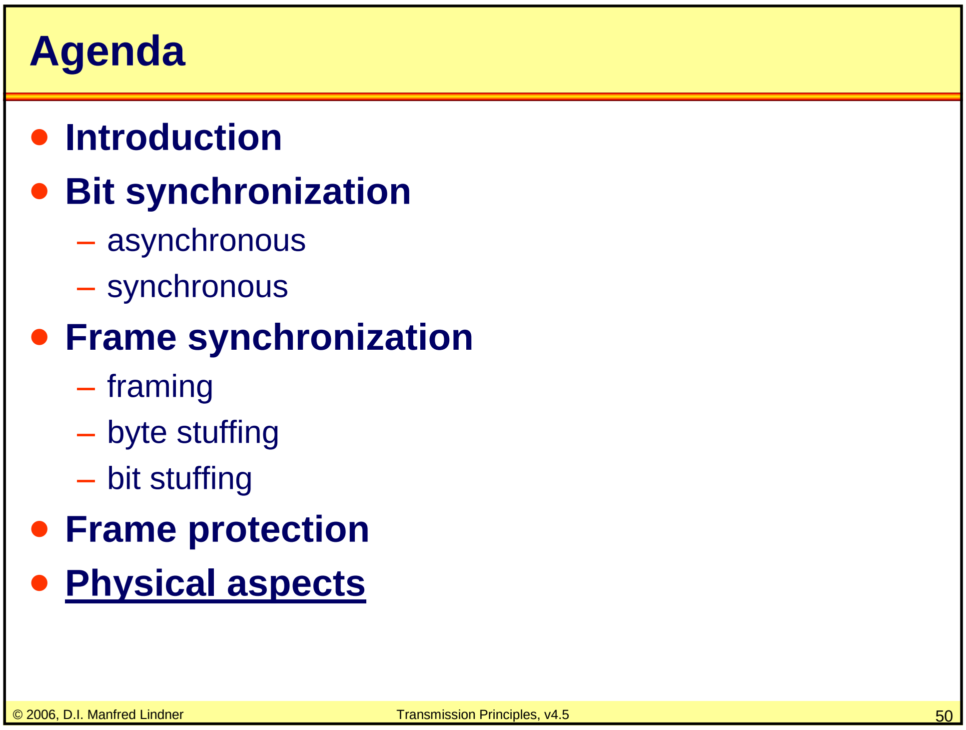
Frame Check Sequence (FCS)

- **sender generates checksum (FCS) using an agreed rule in order to protect the data block**
  - FCS is added at the end of the frame (FCS_tmt)
  - frame protection
- **receiver calculates its own FCS_rcv and compares it with FCS_tmt**
  - error detection
  - FCS_rcv = FCS_tmt ... no error
  - FCS_rcv not equal FCS_tmt ... ERROR
- **detection of an error**
  - error recovery done by retransmission of frame



FCS Methods

- **many possibilities for creating checksums (FCS)**
  - parity bit (even, odd)
  - summarization of all data words modulo 2
  - Cyclic Redundancy Check (CRC) which is based on theory of polynomial code (most complex method)
- **complexity of checksum method determines**
  - types of errors that can be detected for 100%
  - error probability for undetectable errors for a given frame size
- **different FCS methods were standardized**
  - depending on physical network type and expected line error patterns



Agenda

- **Introduction**
- **Bit synchronization**
  - asynchronous
  - synchronous
- **Frame synchronization**
  - framing
  - byte stuffing
  - bit stuffing
- **Frame protection**
- **Physical aspects**

Theoretical Basis for Data Transmission

- **How can a digital signal be represented?**
  - Fourier analysis proves that any periodic function g(t) with period T can be constructed by summing a (infinite in case of rectangle pulses) number of sinus and cosines functions

$$g(t) = (1/2)c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft)$$

  - with f = 1/T and a_n and b_n as amplitudes of the n^{th} harmonics and c as the dc component
  - such a decomposition is called Fourier series

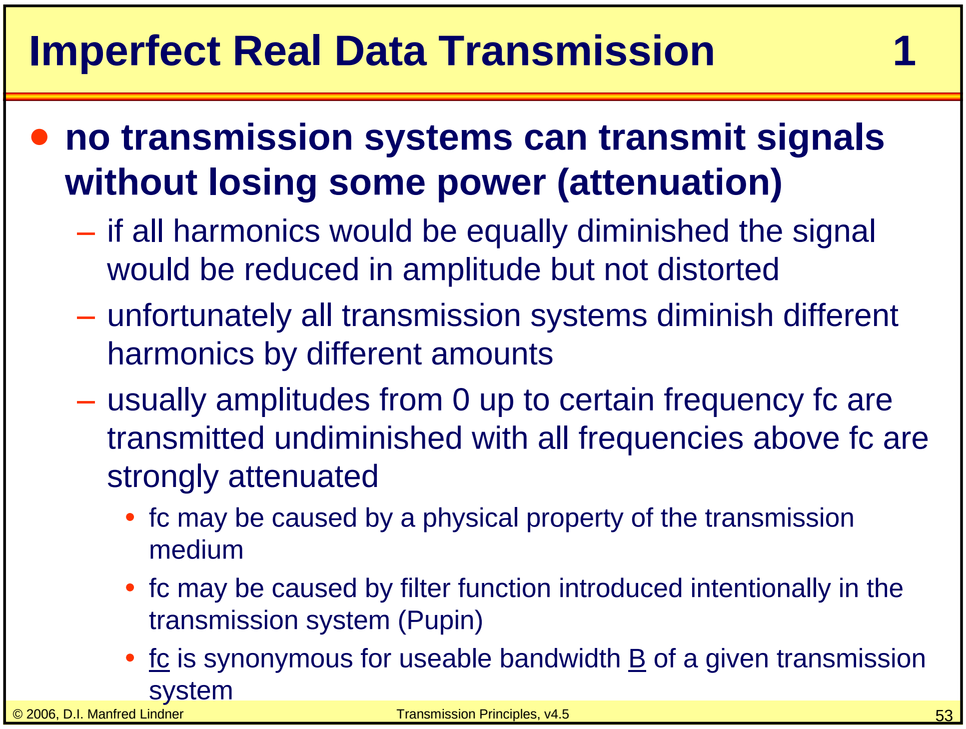
Fourier Coefficients

- **How can the values of *c*, *a_n* and *b_n* be computed?**

$$c = (2/T)\int_0^T g(t)dt$$

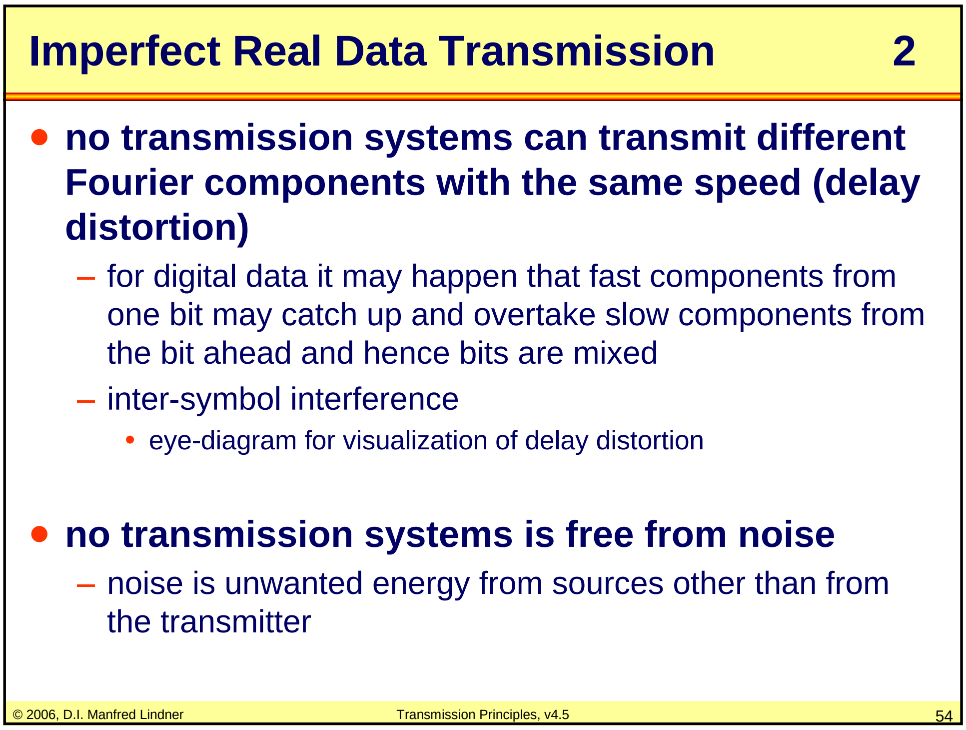
$$a_n = (2/T)\int_0^T g(t)\sin(2\pi nft)dt$$

$$b_n = (2/T)\int_0^T g(t)\cos(2\pi nft)dt$$



Imperfect Real Data Transmission 1

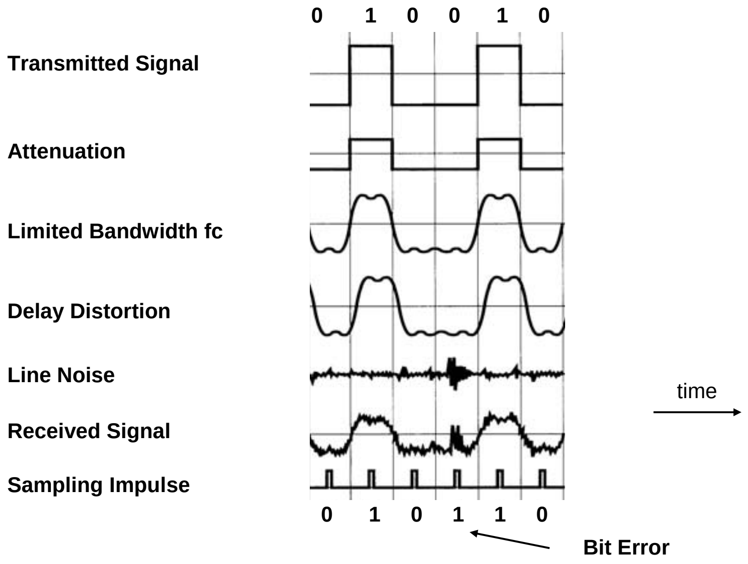
- **no transmission systems can transmit signals without losing some power (attenuation)**
  - if all harmonics would be equally diminished the signal would be reduced in amplitude but not distorted
  - unfortunately all transmission systems diminish different harmonics by different amounts
  - usually amplitudes from 0 up to certain frequency fc are transmitted undiminished with all frequencies above fc are strongly attenuated
    - fc may be caused by a physical property of the transmission medium
    - fc may be caused by filter function introduced intentionally in the transmission system (Pupin)
    - fc is synonymous for useable bandwidth B of a given transmission system



Imperfect Real Data Transmission 2

- **no transmission systems can transmit different Fourier components with the same speed (delay distortion)**
  - for digital data it may happen that fast components from one bit may catch up and overtake slow components from the bit ahead and hence bits are mixed
  - inter-symbol interference
    - eye-diagram for visualization of delay distortion
- **no transmission systems is free from noise**
  - noise is unwanted energy from sources other than from the transmitter

What Happens to a Signal?



Real Data Transmission

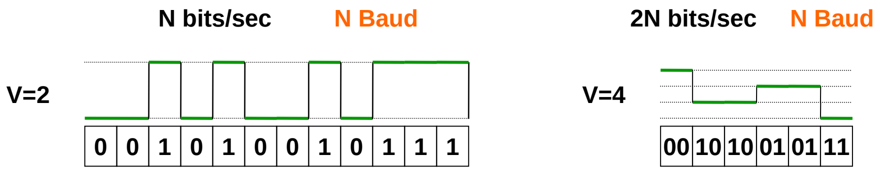
- **in real transmission systems**
  - the original signal will be attenuated, distorted and influenced by noise when traversing the transmission line
- **by increasing the bit rate**
  - bit synchronization even in middle of a bit becomes more and more difficult because of these impairments
  - above a certain rate bit synchronization will be impossible
- **relationship**
  - between bandwidth fc, line length and maximum achievable bit rate on a certain transmission line (system)

Nyquist´s Law

- **How many bits can be transported over a ideal (noiseless) transmission channel ?**
  - Nyquist´s law: $R = 2 * B * \log_2 V$
    - valid for a noiseless channel
    - R ... maximum bit rate (bits/sec)
    - B ... bandwidth range of a bandwidth limited transmission
    - V ... number of signal levels (e.g. 2 for binary transmission)
  - example analogue telephone line
    - B = 3000 Hz (range 400 – 3400 Hz)
    - R = 6000 bits/sec for V = 2
    - R = 18000 bits/sec for V = 8

Bit and Baud

- **the rate of changes of a signal**
  - is called signaling rate R_s and is measured in Baud
- **the rate of bits transported**
  - is called bit rate R and is measured in bits/sec (bps)
  - $R = R_s * \log_2 V$
    - V … number of signal levels
- **$R = R_s$**
  - for binary transmission where V = 2



Shannon´s Law

- **How many bits can be transported over a noisy transmission channel ?**
  - disturbance caused by crosstalk, impulse noise, thermal or white noise
  - Shannon´s law: max R = B * $\log_2$ (1+S/N)
    - S ... signal power, N ... noise power
    - SNR ... Signal to Noise Ratio measured in decibel (db)
    - SNR = 10 * $\log_{10}$ S/N
  - example analogue telephone line
    - B = 3000 Hz
    - SNR = 30 db means 30 = 10 * $\log_{10}$ (S/N) -> S/N = 1000
    - max R = approximately 30000 bits/sec

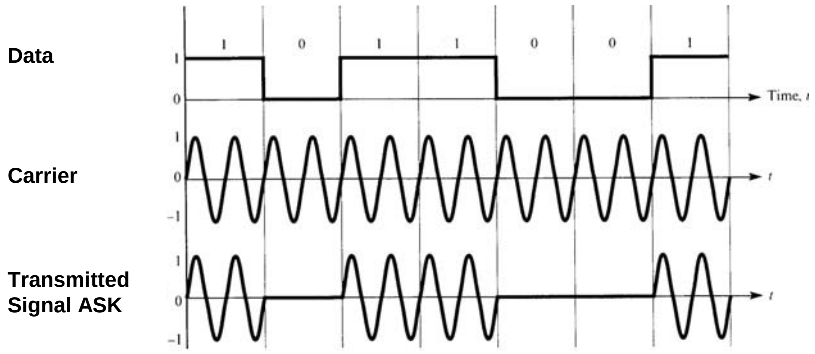
Baseband Mode

- **all the available bandwidth of the serial line is used to derive a single transmission path**
- **signals travel as rectangle pulses**
- **physical property of transmission medium, power of sender, sensitivity of receiver and S/N ratio are the limiting factors for the achievable bit rate**
- **appropriate encoding**
  - to ensure bit synchronization
  - to avoid dc component
  - to keep electromagnetic radiation low

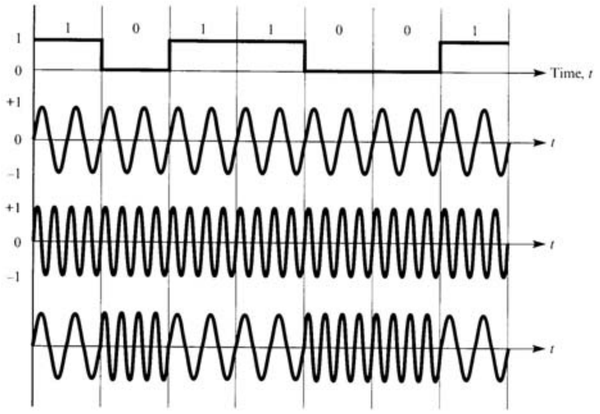
Narrowband Mode

- **bandwidth is intentionally limited and hence binary signals (rectangle pulses) must be adapted before using the line**
- **adaptation is done by modulation**
  - e.g. Modem for transport of data over telephone network
- **several techniques were developed**
  - amplitude modulation (amplitude-shift-keying ASK)
  - frequency modulation (frequency-shift-keying FSK)
  - phase modulation (phase-shift-keying PSK)
  - combination of above like QAM (Quadrature Amplitude Modulation) used in modern high speed modems today
    - 12 phase shifts and two amplitudes are used to represent 16 valid combinations -> 4 bits are transported in a single step
    - used e.g. by V.32 with 9600 bit/sec over 2400 baud line

Amplitude Modulation (AM, ASK)



Frequency Modulation (FM, FSK)



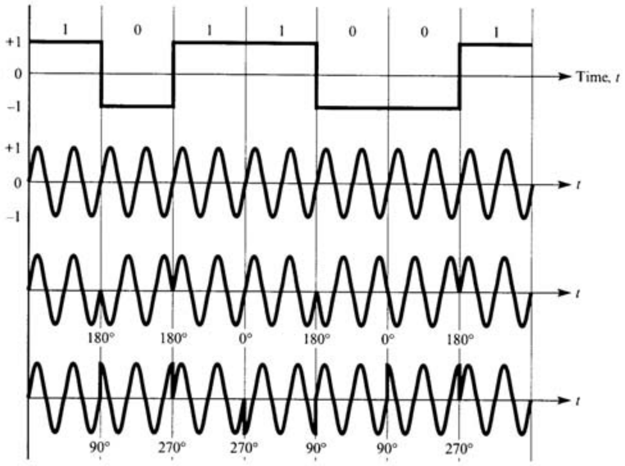
Data

Carrier1

Carrier2

Transmitted Signal FSK

Phase Modulation (PSK)

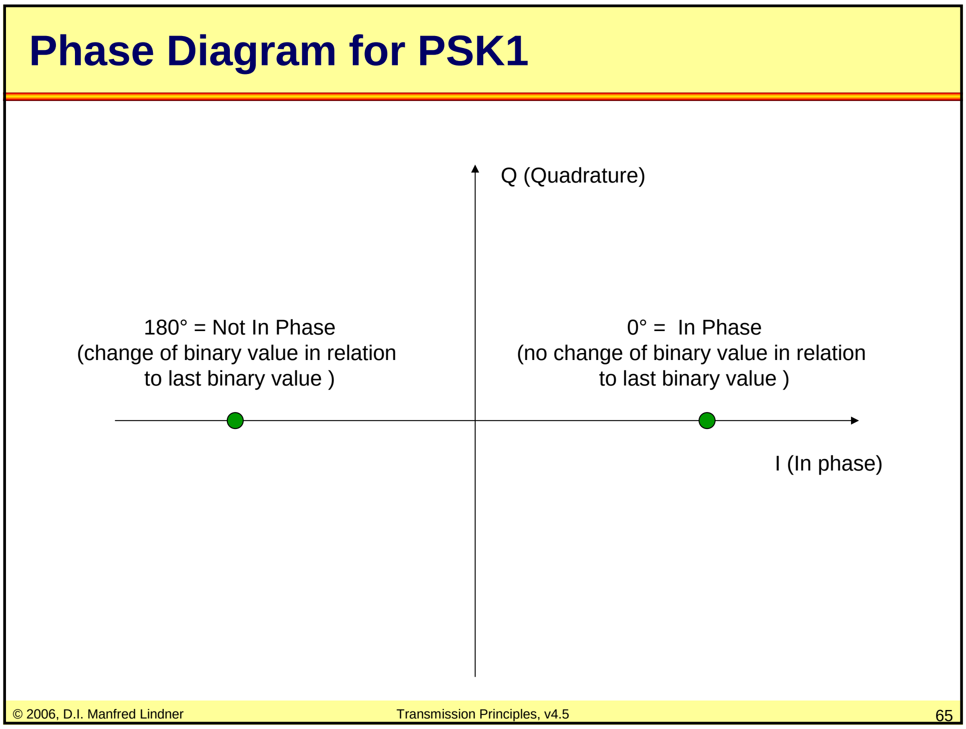


Data

Carrier

Transmitted Signal PSK1

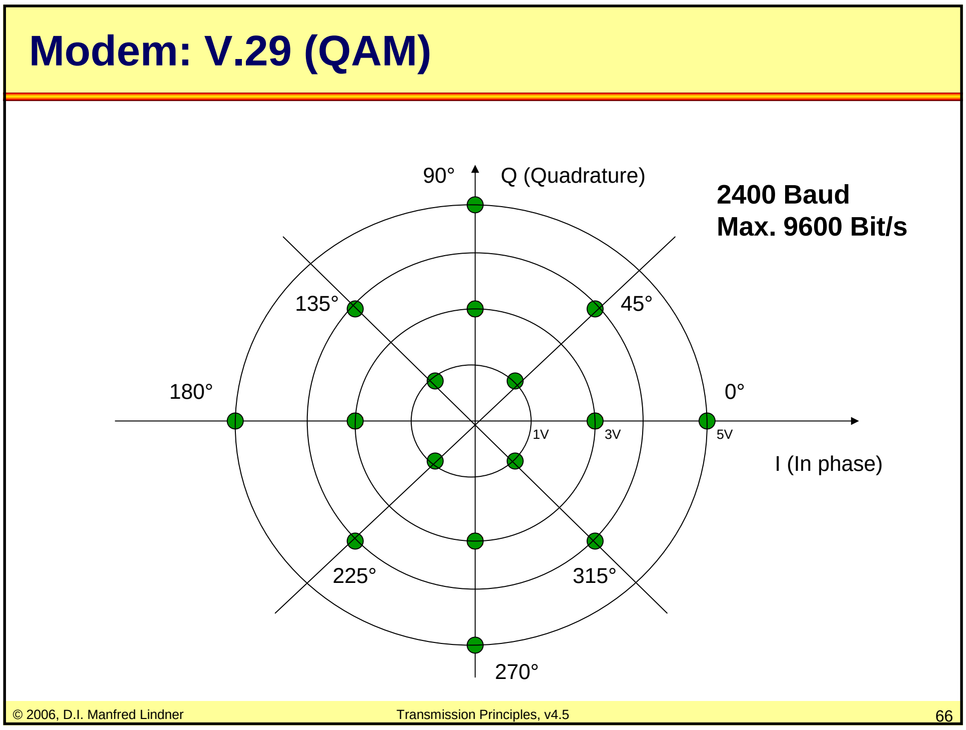
Transmitted Signal PSK2



Phase Diagram for PSK1

180° = Not In Phase
(change of binary value in relation
to last binary value)

0° = In Phase
(no change of binary value in relation
to last binary value)



Modem: V.29 (QAM)

2400 Baud
Max. 9600 Bit/s

Modem

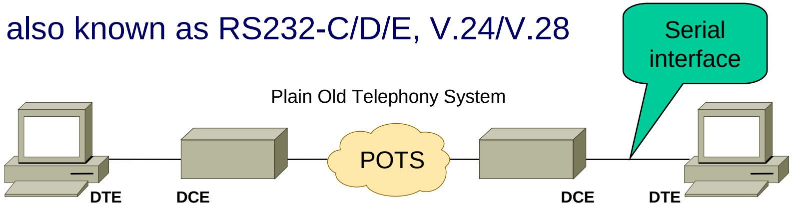
- **Modulator / Demodulator**
  - modem adapts digital (rectangle) signals in order to be transported over analogous telephone network
    - limited bandwidth (200 - 3500 Hz)
  - done by different modulation techniques
    - AM, FM, Phase-Modulation, QAM, Trellis-Code, etc.
  - 1st Wave
    - Frequency Division Protocols, all rates to 2400 bits/s
      - Modems: advanced analog filters
      - Telco: pass audio frequencies of 200 Hz to 2.4 KHz
  - 2nd Wave
    - 1st generation Echo Canceling Protocols, 9600 & 14400 bits/s
      - Modems: low cost DSPs
      - Telco: pass audio frequencies of 200 Hz to 2.4 KHz

Modem (cont.)

- 3rd Wave
  - 2nd gen. Echo Canceling Protocols, rates to 28.8 Kbits/s
    - Modems: higher performing, low cost DSPs
    - Telco: pass audio frequencies of 200 Hz to 2.8 KHz
- 4th Wave
  - extending Echo Canceling Protocols, rates to 33.6 Kbits/s
    - Modems: higher performing, low cost DSPs
    - Telco: pass audio frequencies of 200 Hz to 3.1 KH
- 5th Wave
  - Digital stepping protocols, 34 Kbits/s to 56 Kbits/s
    - Modems: higher performing, low cost DSPs
    - Telco: pass audio frequencies of 200 Hz to 3.1 KHz,
      all digital path to subscriber line,
      64K PCM digital to analog conversion,
      limited loop length, no line conditioners

Modem Control by EIA-232 / V.24

- **EIA-232 / V.24 standard**
  - serial interface definition between a DCE and DTE
    - DTE (Data Terminal Equipment e.g. end system)
    - DCE (Data Circuit Terminating Equipment e.g. modem)
  - for short distance and low speed connectivity
  - specifies a set of physical lines and necessary electrical / mechanical aspects
    - data signals for serial transmission, control signals for modem (DCE) control, unbalanced transmission, connector
  - also known as RS232-C/D/E, V.24/V.28



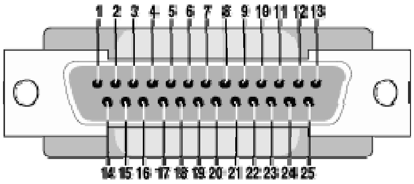
EIA-232 Data and Control Signals

- data signals:
  - transport of serial data bit-stream
  - TxD (Transmit Data) DTE -> DCD
  - RxD (Receive Data) DCE -> DTE
- control signals:
  - control function between modem and end system
  - RTS (Request To Send) DTE -> DCE
    - DTE requests permission to send data to modem
  - CTS (Clear To Send) DCE-> DTE
    - DCE grants permission to send
  - DCD (Data Carrier Detect) DCE -> DTE
    - DCE indicates that it is receiving carrier from remote modem
  - DSR (Data Set Ready) DCE -> DTE
    - DCE indicates that it is operational (the modem is powered on)

EIA-232 Control Signals (cont.)

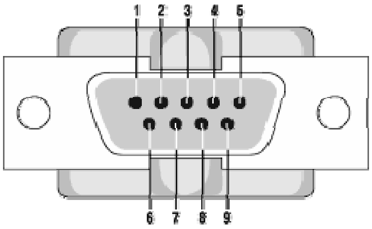
- control signals (cont.)
  - DTR (Data Terminal Ready) DTE -> DCE
    - DTE indicates that it is operational (the end system is powered on)
  - RI (Ring Indicator) DCE -> DTE
    - DCE indicates that the phone is ringing
  - Transmitter Signal Element Timing DCE -> DTE
    - used in synchronous mode to provide clock to the DTE for TxD
  - Receiver Signal Element Timing DCE -> DTE
    - used in synchronous mode to provide clock to the DTE for RxD
  - Transmitter Signal Element Timing Return DTE -> DCE
- EIA-232 specified limits:
  - Length: 15m, 30m
  - Speed: 20kbit/sec, 64kbit/sec / Practice: up to 200kbit/sec

EIA-232 Pinout DB-25



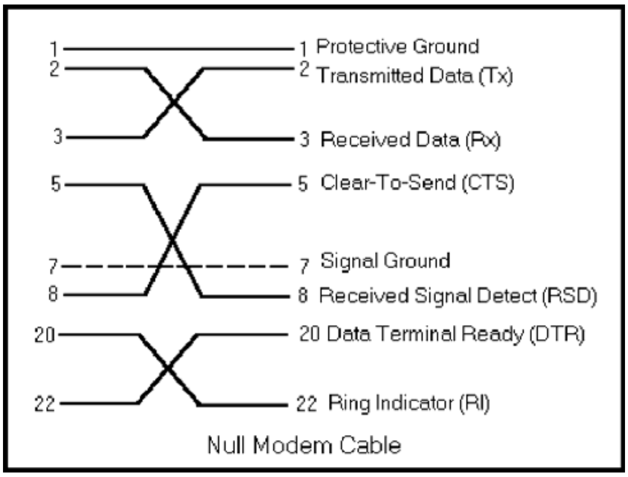
2	**Transmit Data (TxD)**
3	**Receive Data (RxD)**
4	**Request to Send (RTS)**
5	**Clear to Send (CTS)**
6	**Dataset Ready (DSR)**
7	**Signal Ground**
8	**Data Carrier Detect (DCD)**
15	**Transmit Clock**
17	**Receive Clock**
20	**Data Terminal Ready (DTR)**
24	**Auxiliary Clock**

EIA-232 Pinout DE-9



1 Data Carrier Detect (DCD)
2 Transmit Data (TxD)
3 Receive Data (RxD)
4 Data Terminal Ready (DTR)
5 Signal Ground
6 Dataset Ready (DSR)
7 Request to Send (RTS)
8 Clear to Send (CTS)
9 Ring Indicator (RI)

EIA-232 Null Modem Cable



Broadband Mode

- **the available bandwidth of the serial line is divided to derive a number of lower bandwidth transmission paths on one serial line**
- **in analogue systems every path is modulated by a unique carrier**
  - a certain base-frequency which together with the necessary bandwidth range for that channel occupies a certain frequency band of the given transmission system
  - cable television as example
- **in digital systems broadband means sometimes high speed only**
  - e.g. B-ISDN = ATM
  - but no modulation is used to achieve these

Frequency Usage for the Communication Channel