


# Chapter 2

---

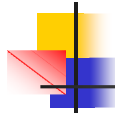
## Data Communications Concepts



# Topics

---

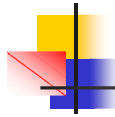
- Data:
  - Encoding
  - Digitization
  - Transmission
  - Modulation
  - Compression
- Packetization
- Encapsulation
- Multiplexing
- Switching
  - Circuit
  - Packet
- Datagram Delivery
- Error
  - Detection
  - Correction
- Flow Control



# Data Digitization

## Character encoding

- Process of transforming humanly readable characters into machine readable code.
  - Characters are transformed into a series of ones and zeroes.
  - Commonly used standards are
    - ASCII
    - EBCDIC
    - UNICODE



## 7-Bit ASCII Code table

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

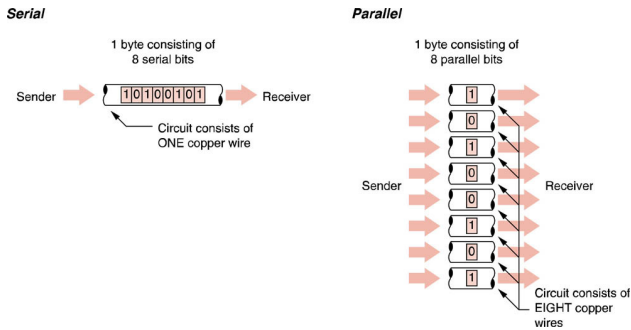
• Ascii encoding is used on most computers today

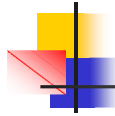
# EBCDIC Code Table

MSB	Bit 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1						
	Bit 1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1	1	1						
	Bit 2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1						
	Bit 3	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1						
LSB		Bit 7	Bit 6	Bit 5	Bit 4																				
		0	0	0	0	NUL	DLE	DS	SP	&	-														
		1	0	0	0	SOH	DC1	SOS				a	j	A	J										
		0	1	0	0	STX	DC2	FS	SYN				b	k	s	B	K	S	2						
		1	1	0	0	ETX	DC3				c	l	t	C	L	T									
		0	0	1	0	PF	RES	BYP	PN				d	m	u	D	M	U	4						
		1	0	1	0	HT	NL	LF	RS				e	n	v	E	N	V	5						
		0	1	1	0	LC	BS	EOB	UC				f	o	w	F	O	W	6						
		1	1	1	0	DEL	IL	PRE	EOT				g	p	x	G	P	X	7						
		0	0	0	1	CAN						h	q	y	H	Q	Y	8							
		1	0	0	1	EM						\	i	r	z	I	R	Z	9						
		0	1	0	1	SMM	CC	SM	>> 1 :																
		1	1	0	1	VT				. \$ . #															
		0	0	1	1	FF	IFS	DC4	<	*	%	@													
		1	0	1	1	CR	IGS	ENQ	NAK	(	)														
		0	1	1	1	SO	IRS	ACK	+	:	>	=													
		1	1	1	1	SI	IUS	BEL	SUB		-	?													

- EBCDIC is used on IBM Mainframes

# Serial vs. Parallel Data Transmission





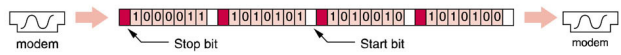
## Serial vs. parallel Data Transmission

Transmission Characteristic	Serial	Parallel
<b>Transmission Description</b>	One bit after another, one at a time	All bits in a single character transmitted simultaneously
<b>Comparative Speed</b>	Slower	Faster
<b>Distance Limitation</b>	Farther	Shorter
<b>Application</b>	Between two computers, from computers to external devices, local and wide area networks	Within a computer along the computer's busses, between a drive controller and a hard drive
<b>Cable Description</b>	All bits travel down a single wire, one bit at a time	Each bit travels down its own wire simultaneously with other bits.



## Synchronous vs. Asynchronous transmission

### Asynchronous transmission



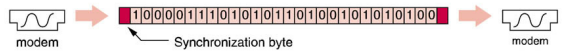
#### Characteristics:

Data is sent one byte at a time  
 Each byte has a start and 1, 1.5, or 2 stop bits  
 Synchronization is reestablished for each byte  
 Time between bytes is unsynchronized and of random length

#### Efficiency (1,000 byte transmission)

Control / overhead bits: 1 start and stop bits per byte  
 2 control bits per byte x 1,000 bytes = 2,000 control bits  
 7,000 data bits / 9,000 total bits = 77.7% efficient

### Synchronous transmission

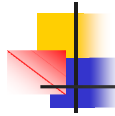


#### Characteristics:

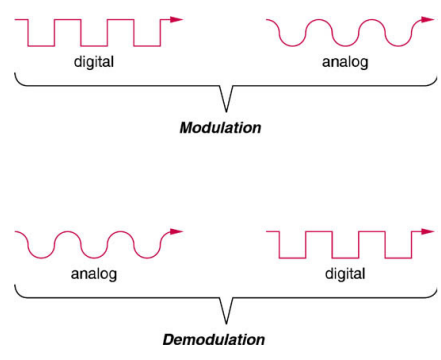
Data is sent as a block of uninterrupted bytes  
 Synchronization bytes precede and follow the data block  
 Synchronization is maintained whether data is actually being sent and detected or not  
 Modems remain synchronized during idle time

#### Efficiency (1,000 byte transmission)

Control / overhead bits: 48 total control bits per block using HDLC  
 48 control bits per block x 1 block = 48 control bits  
 7,000 data bits / 7,048 total bits = 99.3% efficient



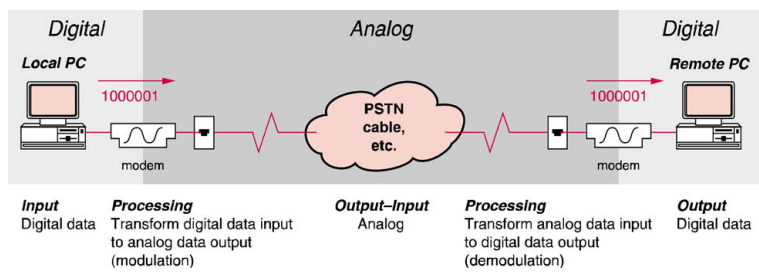
## Modulation vs. demodulation



- This process is done by a mo(dulator)dem(odulator)



## Modem based communication channels

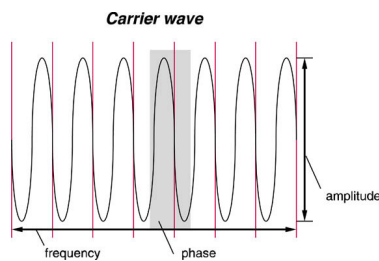


- The dial-up modem allows connections through the phone network

## Half vs. full duplex

- Almost all data communications sessions are bi-directional.
- Bi-directional traffic may be full or half duplex.
- In a full duplex communications environment both devices can transmit at the same time.
- In a half duplex environment at any given point in time, you can only listen *or* talk.

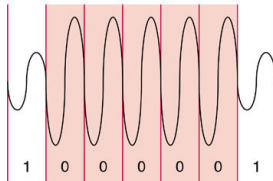
## Carrier Wave



- There are three properties of a wave that can be modulated or altered:
  - Amplitude
  - Frequency
  - Phase

# Amplitude Modulation

*Amplitude modulation*  
(Frequency, phase constant)

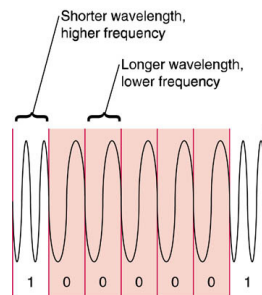


ASCII - 7: letter A

- Each vertical lines separates opportunities to identify a 1 or 0 from another.
- These timed opportunities are known as signaling events.
- The proper name for one signaling event is a **baud**

# Frequency Modulation

*Frequency Modulation*  
(Amplitude, phase constant)

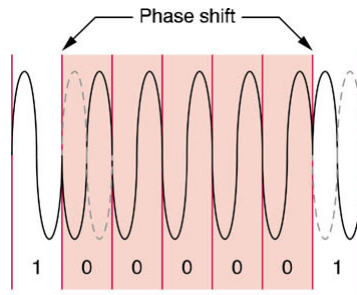


ASCII-7: letter A

- **frequency shift keying or FSK**



# Phase Modulation

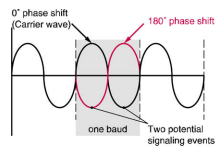


ASCII-7: letter A

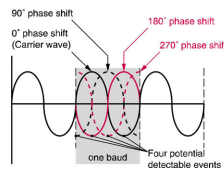
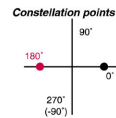
- phase shift keying or PSK



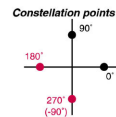
# Detecting Phase Shifting



Phase shift	Interpreted bit pattern
0°	0
180°	1



Phase shift	Interpreted bit pattern
0°	00
90°	01
180°	10
270°	11

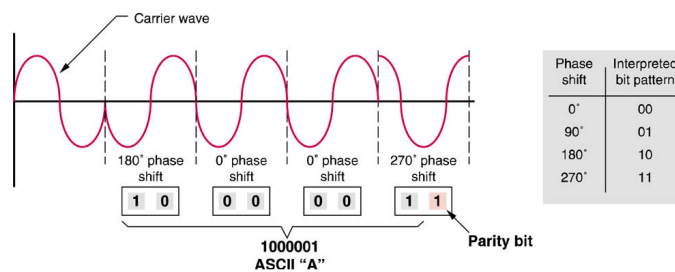


- Quadrature Phase Shift Keying

## Increasing transmission efficiency

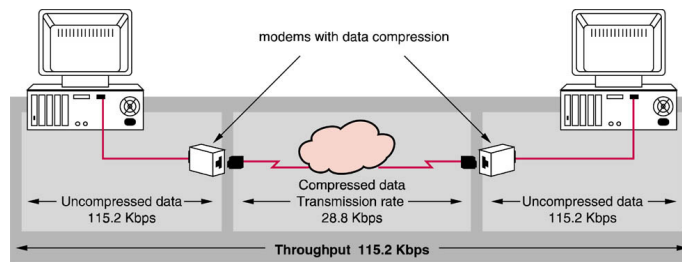
- Two ways in which a given modem can transmit data faster:
  1. Increase the signaling events per second, or baud rate.
  2. Find a way for the modem to interpret more than one bit per baud.

## Differential Quadrature Phase Shift keying



- This technique improves transmission rate by increasing the number of events per baud

## Data Compression



- Data compression techniques improve throughput.

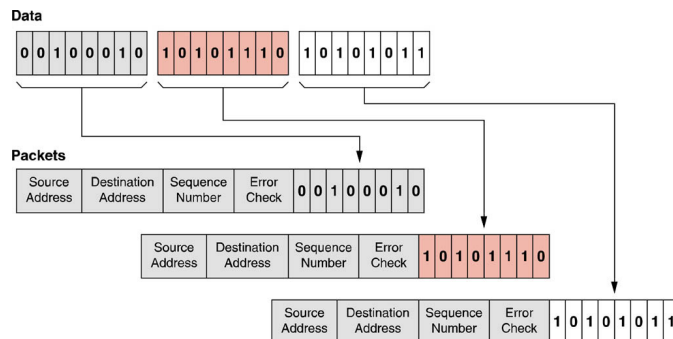
## Data compression

- The sending device replaces strings of repeating character patterns with a special code that represents the pattern.
- The code is significantly smaller than the pattern it represents.
- This results in the amount of data sent between the sending device and the receiving device to increase.

## Packetization

- The process of dividing the data stream flowing between devices into structured blocks known as packets.
- A **packet** is a group of bits organized in a pre-determined, structured manner consisting of a piece of the data stream to which management information is added.

## Packetization



- This data stream is divided into 3 packets
- Note the addition of header information to the data portion



## Packetization

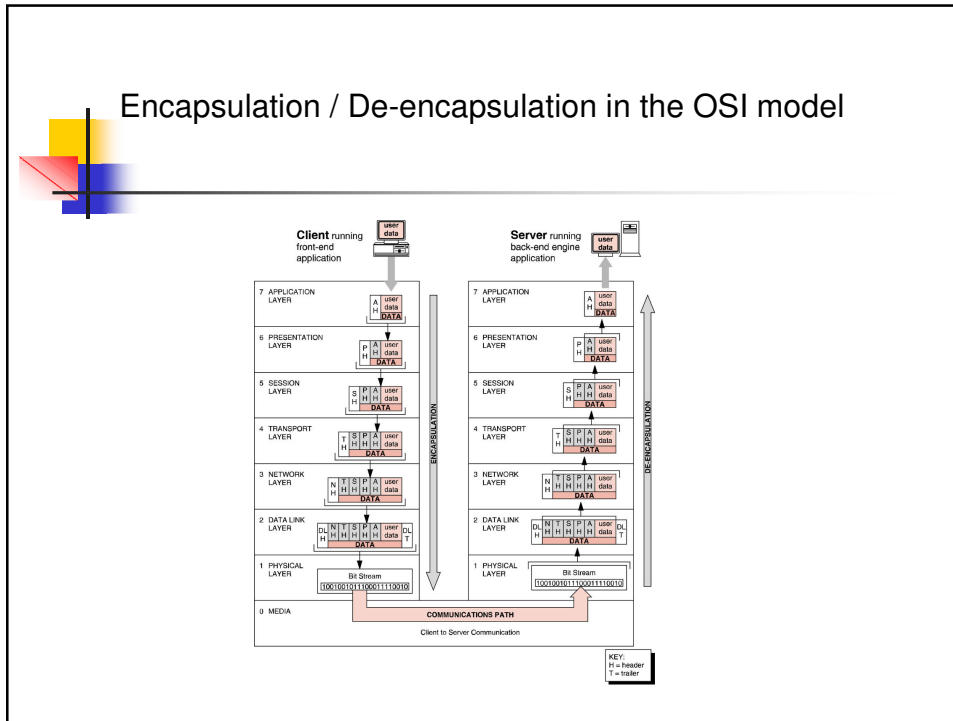
- The predetermined structure of a packet is critical.
- Through the use of standards, devices know the number of bits in each section; the header, data portion and trailer.



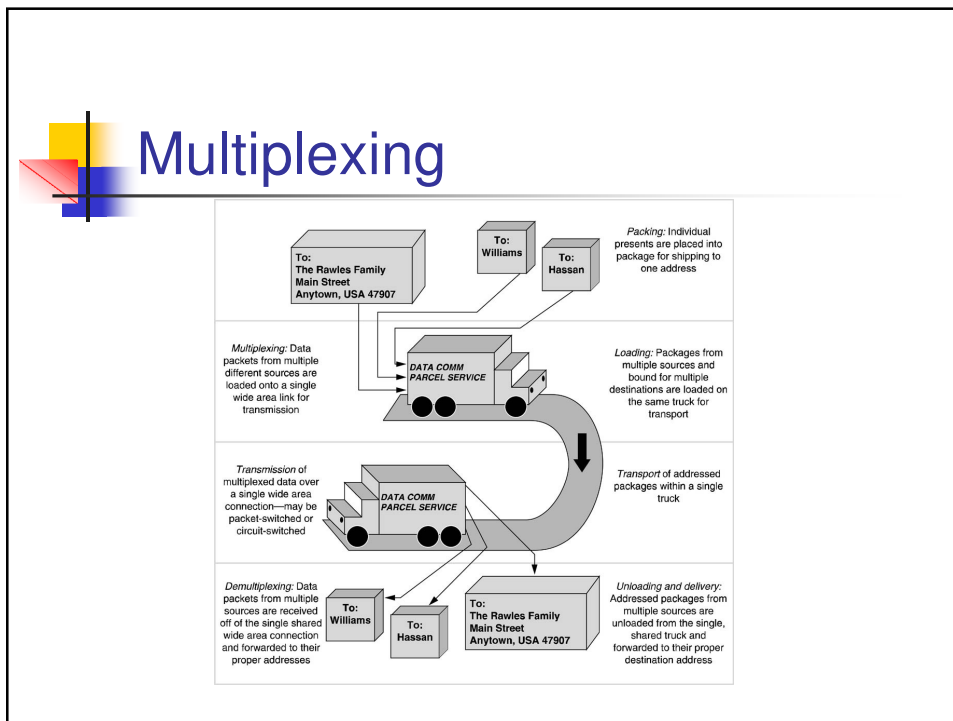
## Encapsulation / De-encapsulation

- In a layered protocol, each layer adds a header according to the layer's syntax.
- The sending device adds this information in a process of encapsulation
- The receiving device reverses the this process (de-encapsulation)

## Encapsulation / De-encapsulation in the OSI model

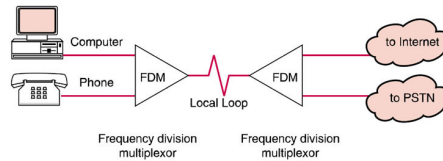


## Multiplexing

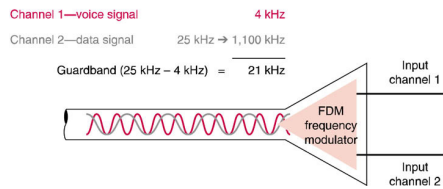


# Frequency Division Multiplexing

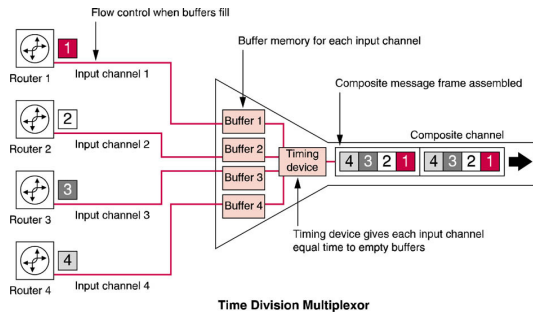
## Overall Configuration



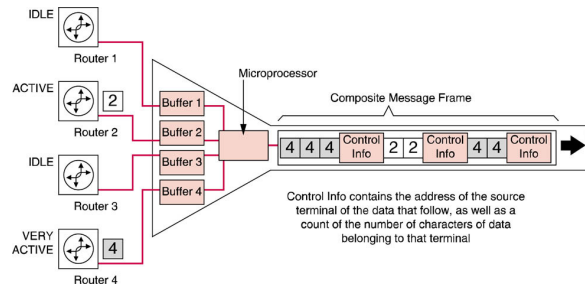
## Inside the Local Loop



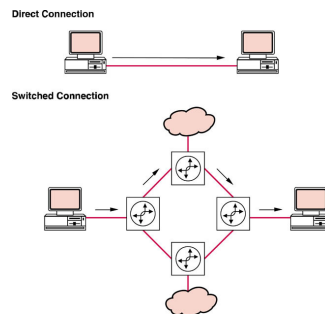
# Time Division Multiplexing



# Statistical Time Division Multiplexing



# Switching



- Switching allows temporary connections to be established, maintained and terminated between sources and destinations



## Circuit Switching

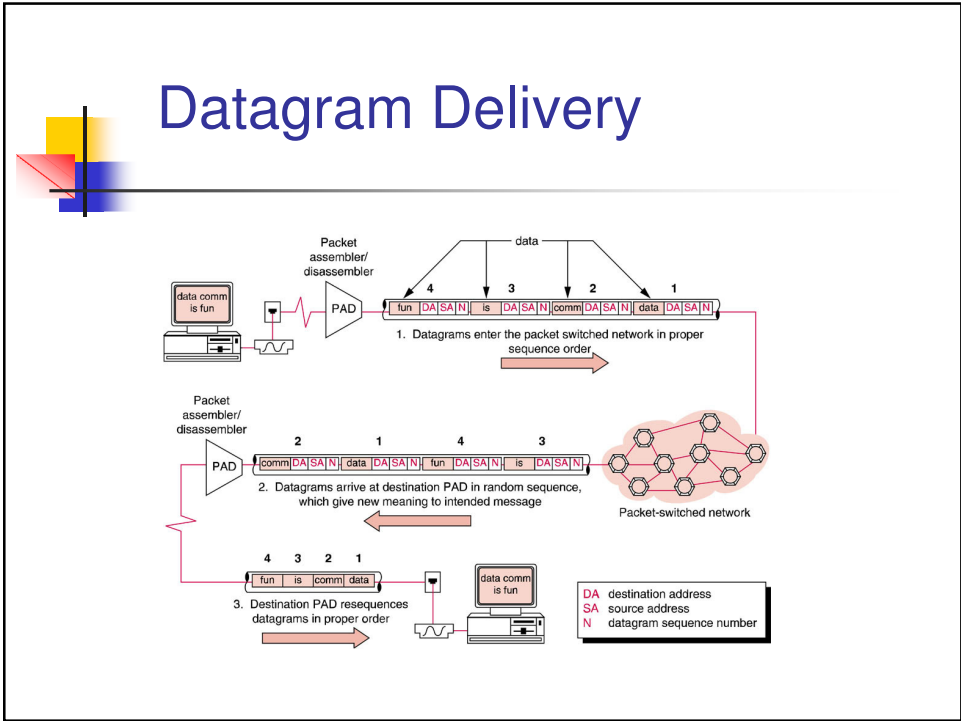
- The work to create a signal path is done up front; a switch fabric creates a direct path between the source and the destination.
- Communication takes place just as if the temporary circuit were a permanent direct connection:
- The switched dedicated circuit makes it appear to the user of the circuit as if a wire has been run directly between the communicating devices.



## Packet switching

- In a **packet switched network**, packets of data travel one at a time from the message source to the message destination.
  - The physical path taken by one packet may be different than that taken by other packets in the data stream.
  - The path is unknown to the end user.
- A series of **packet switches** pass packets among themselves as they travel from source to destination

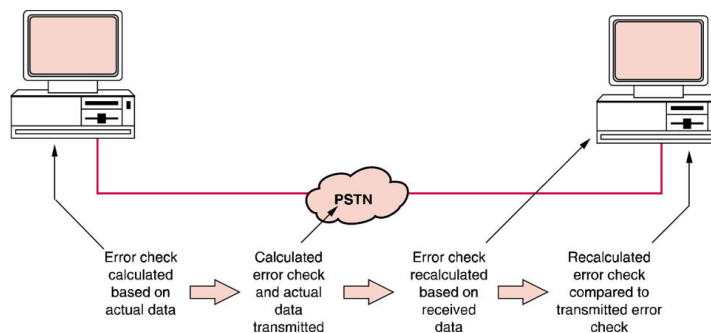
# Datagram Delivery



# Connectionless vs. Connection-Oriented Networks

	Overhead	Greatest Strength	Call Set-up	Addressing	Also Known as...	Error Correction	Flow Control
Connectionless	Less	Ability to dynamically reroute data	None	Global	Datagram unreliable	Left to end-user devices	Left to end-user devices
Connection-oriented	More	Reliability	Yes	Local logical channel number	Reliable Virtual circuit	By virtual circuit	By virtual circuit

## The error detection process



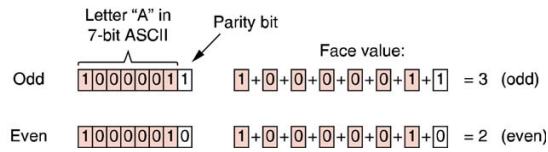
## Error detection process

- The transmitting and receiving devices agree on how the error check is to be calculated
- The transmitting device calculates and transmits the error check along with the transmitted data
- The receiving device re-calculates the error check based on the received data and compares its newly calculated error check to the error check received with the data
- If the two error checks match, everything is fine. If they do not match, an error has occurred



## Parity Checking

### Letter "A" with odd and even parity

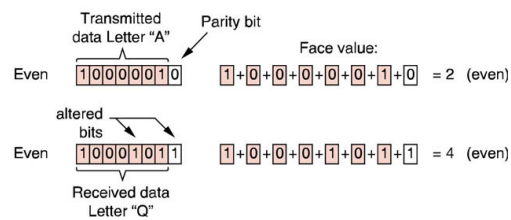


- Simple parity checking

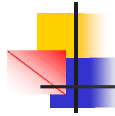


## Parity Checking

### Multiple bit errors/even parity



- Parity checks can miss multiple bit errors

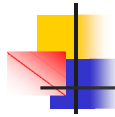


## Alternatives to parity

	Data bits							Parity bit
	1	2	3	4	5	6	7	VRC
A	1	0	0	0	0	0	1	0
B	0	1	1	0	0	0	0	0
C	1	1	0	0	0	0	1	1
D	0	0	1	0	0	0	0	0
E	1	0	1	1	0	0	0	1
F	0	1	1	0	0	0	1	1
G	1	1	1	0	0	0	1	0
Block check character	LRC	0	0	0	0	0	1	1

*This example is based on EVEN parity*

- LRC improves parity checking at the cost of extra data transmitted



## Error Control Techniques

- Error Prevention
- Error Detection



## Error detection techniques

---

- •Parity (VRC)
- •Longitudinal Redundancy Checks (LRC)
- •Checksums
- •Cyclic Redundancy Checks (CRC)



## Parity

---

- **Parity**, also known as a **(Vertical Redundancy Check or VRC)**, is the simplest error detection technique.
- Parity works by adding an error check bit to each character.



## Longitudinal Redundancy Checks

---

- **Longitudinal Redundancy Checks (LRC)** seek to overcome the weakness of simple, bit-oriented one directional parity checking..
- LRC adds a second dimension to parity.



## Checksums

---

- **Checksums** are also block-oriented error detection characters added to a block of data characters.
- a checksum is calculated by adding the decimal face values of all of the characters sent in a given data block and sending only the least significant byte of that sum.
- The receiving modem generates its own checksum and compares the locally calculated checksum with the transmitted checksum



## Error Correction

---

- The receiving device detects the error and requests a re-transmission
- The sending device then retransmits the the portion that contained the error.



## Flow Control

---

- To prevent buffer overflows the receiving device sends a signal to the sending device
- The flow control software constantly monitors the amount of free space available in buffer memory and tells the sending device to stop sending data when there is insufficient storage space.
- When the buffer once again has room the sending device is told to resume transmitting



Questions?

---