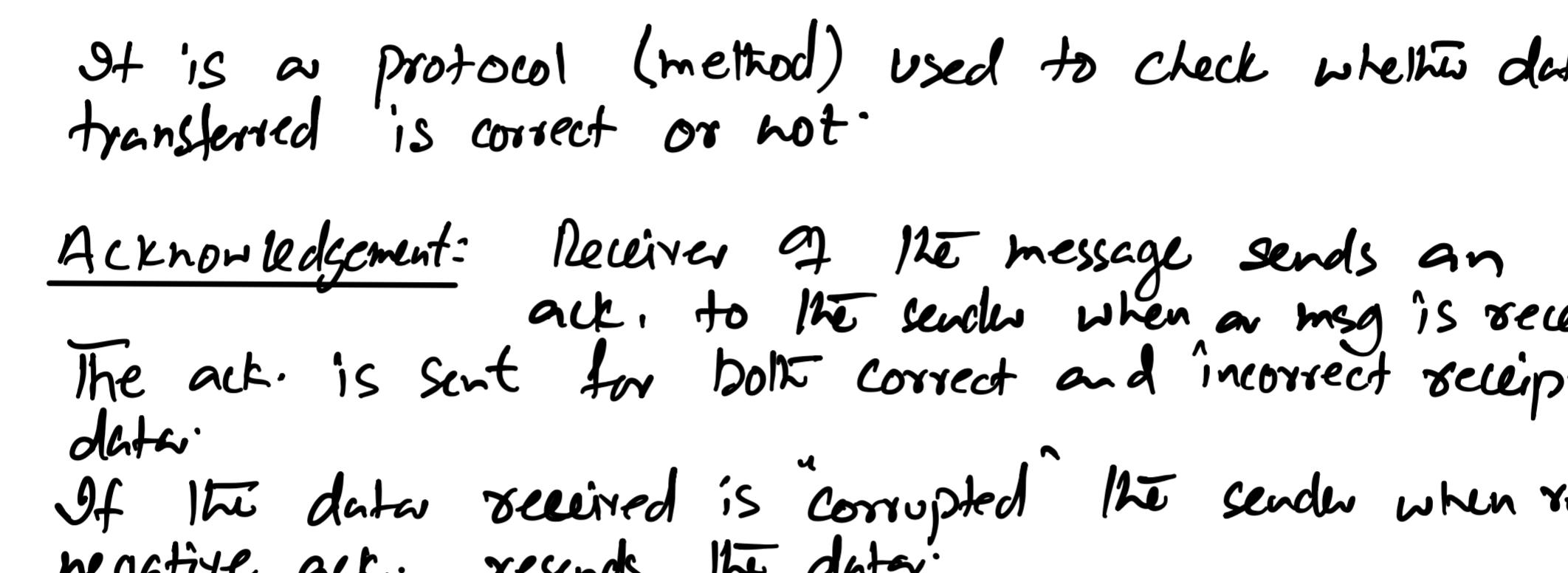


Verification is checking of data against the source. If data is transferred manually to computer then a double entry check or visual check is used. If data is transferred digitally then ARQ, parity, checksum and echo back is used.

Double Entry: Complete data is entered twice and computer compares both entries to find if there was any discrepancy while data was being entered.

Visual Check: Once the data is entered, it is checked visually either on monitor or printout is matched with the source form.

### Automatic Repeat Request (ARQ):



Protocol: Roles set forth between devices before actual data transfer occur. E.g: http, www, ARQ etc.

ARQ: It is a protocol (method) used to check whether data transferred is correct or not.

Acknowledgment: Receiver of the message sends an ack. to the sender when a msg is received. The ack. is sent for both correct and incorrect receipt of data.

If the data received is "corrupted" the sender when receives negative ack., resends the data.

If the data received is "correct data" then next due data is sent.

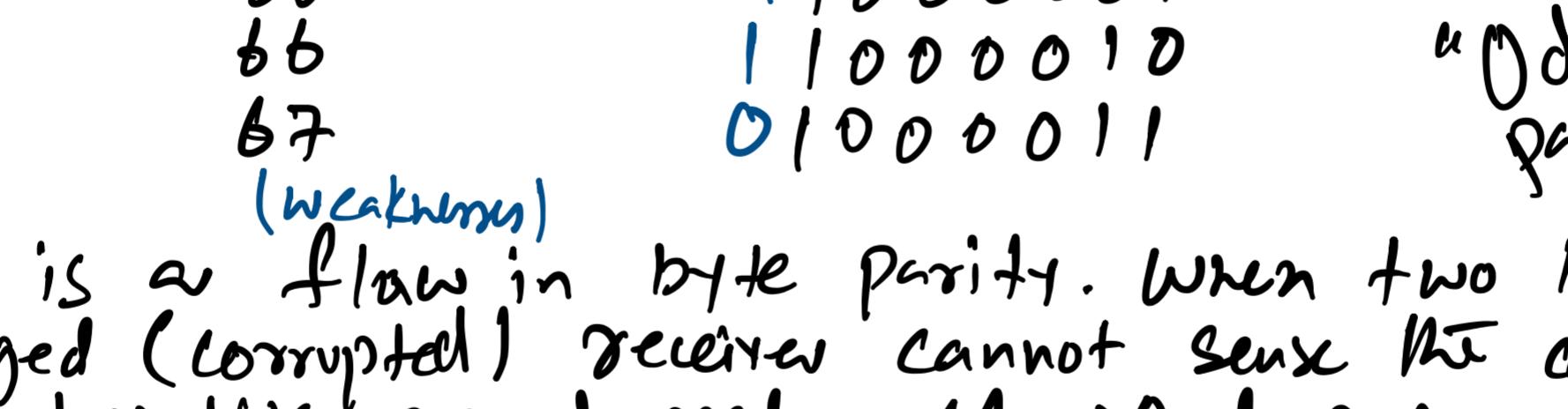
Timeout: When two devices start data transfer, they set an agreed time (as part of protocol) for the receipt of ack.

If ack. is not received for that predefined/preset time, timeout occurs and sender resends the data.

This may go-on until ack finally arrives at sender's end.

Parity Check: It is a method used to identify data corruption upon reception of data/msg.

It is mainly used with text data in close proximity.



There are two parity mechanisms at work:

1. Byte parity.
2. Block parity.

They both use either "even" or "odd" parity types.

Example: Parity is the count of "1's"

CHAR	ASCII DECIMAL	ASCII BINARY	"Even Parity"
A	65	01000001	
B	66	01000010	
C	67	11000011	

A      65      11000001  
B      66      01000010      "Odd parity"  
C      67      11000011

(weakness)

There is a flaw in byte parity. When two bits are changed (corrupted) receiver cannot sense the corruption.

- If two bits are dropped ( $1 \rightarrow 0$  drop)

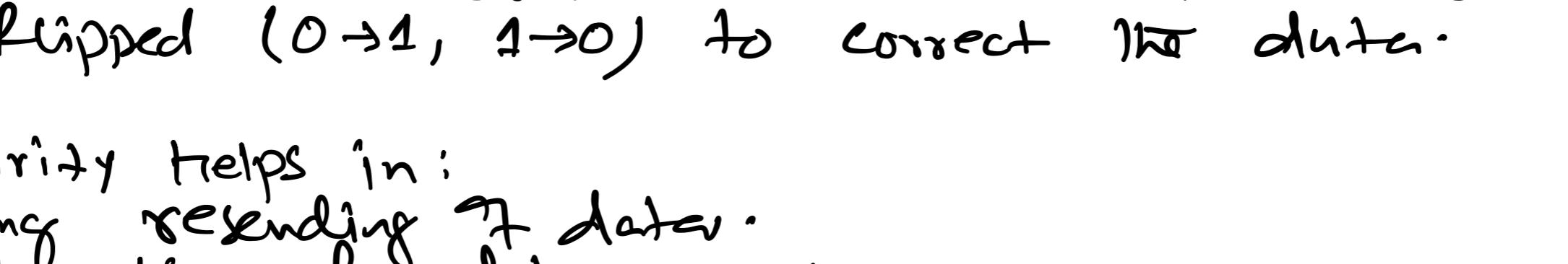
- If two bits are gained ( $0 \rightarrow 1$  gain)

- If two bits exchange position (transposition)

01000011      01000001      two bits are dropped.

11000011      11000101      two bits exchange their positions.

To overcome this weakness "Block Parity" is used. In block parity whole bunch of bytes with byte & column parity is sent.



↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

01000001      01001010      11000011      11000000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

</