



# Lezione 4: Livello di Collegamento I Sottolivello Data-Link

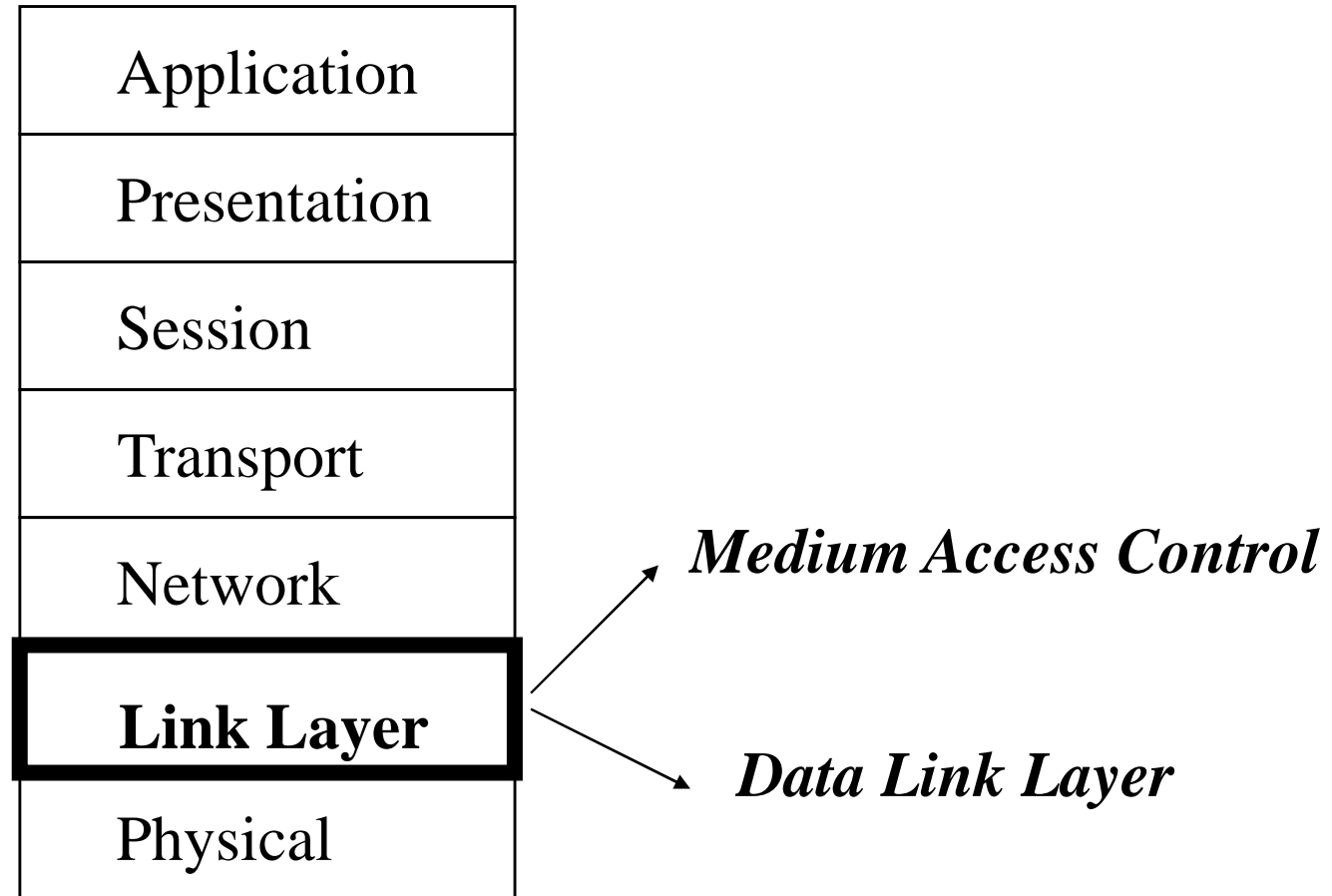


## Sommario della Lezione

- ❑ Descrizione generale livello di Collegamento
- ❑ Principi di progettazione del sottolivello data-link del livello di collegamento
  - Servizi forniti al livello rete
  - Framing
  - Controllo degli errori
  - Controllo del flusso
- ❑ Rilevazione e correzione degli errori
  - Codici di correzione degli errori
  - Codici di rilevazione degli errori



## Dove si trova il livello di Collegamento?



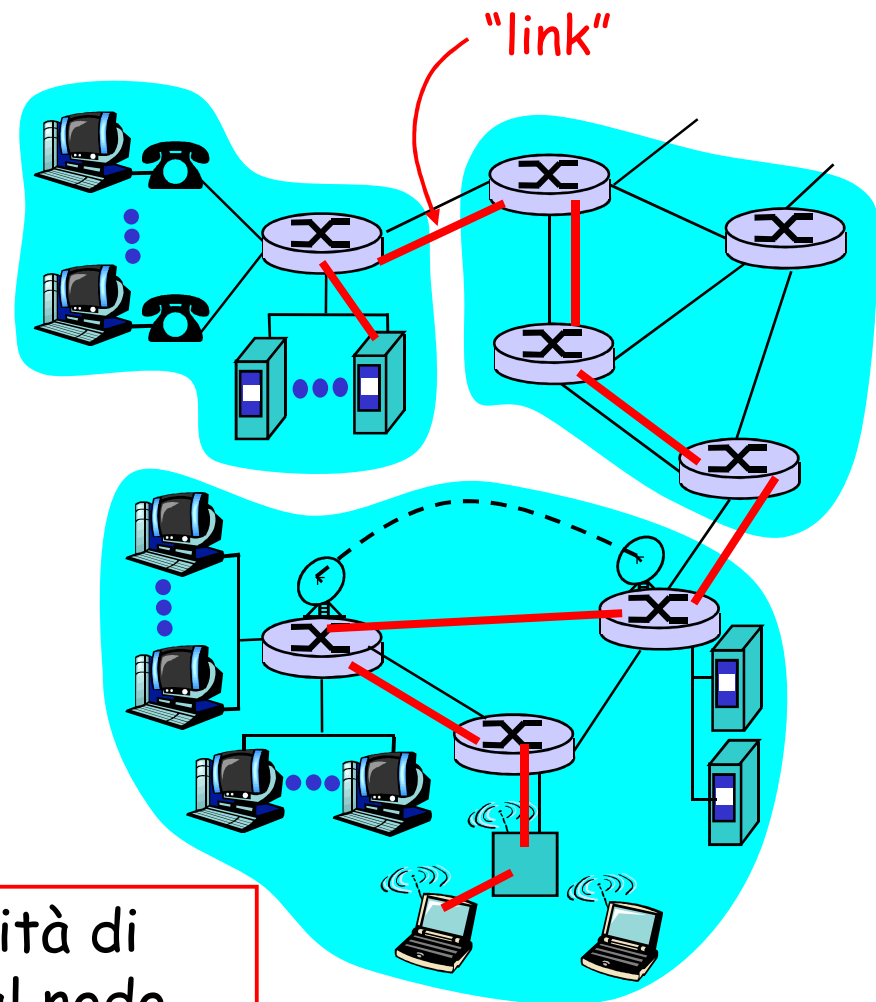
Cavo

---

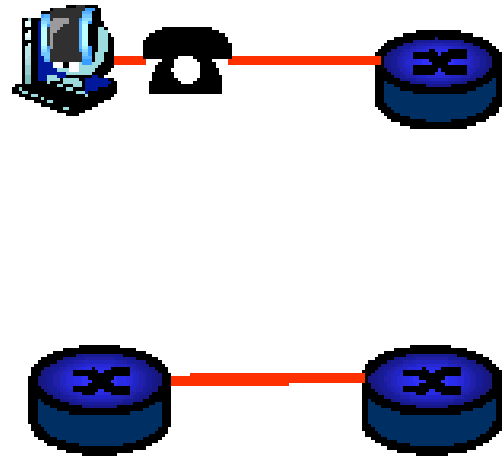
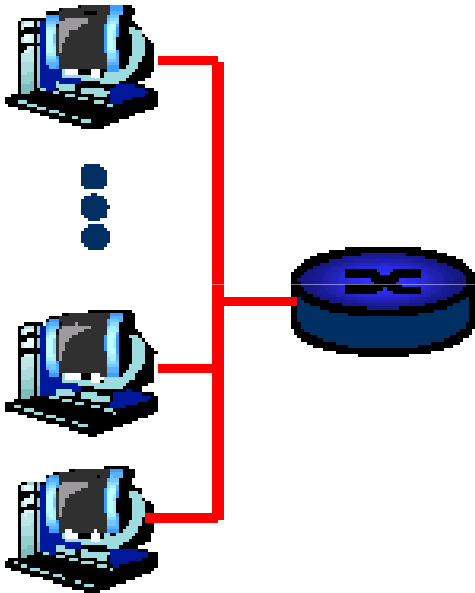
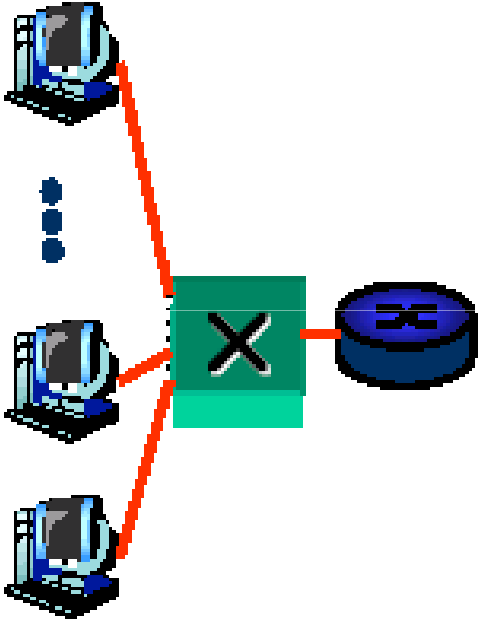
# Terminologia

- ❑ hosts e routers sono **nodi**  
(lo sono anche bridges e switches)
- ❑ canali di comunicazione che connettono nodi adiacenti lungo un percorso sono **links**:
  - wired links
  - wireless links
  - LANs
- ❑ PDU è un **frame**, incapsula datagrammi

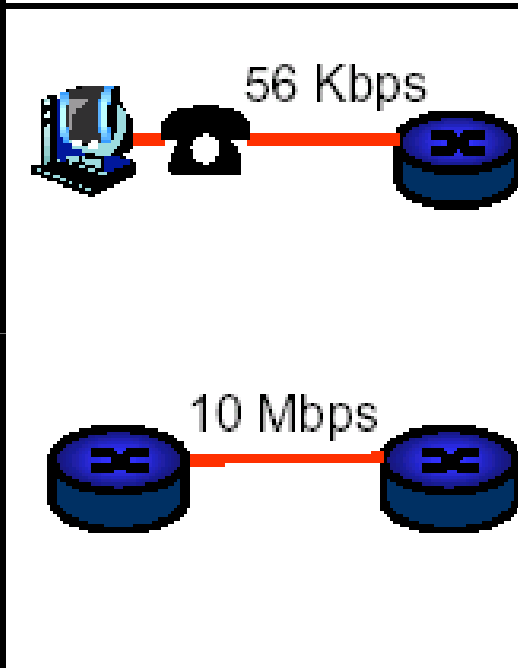
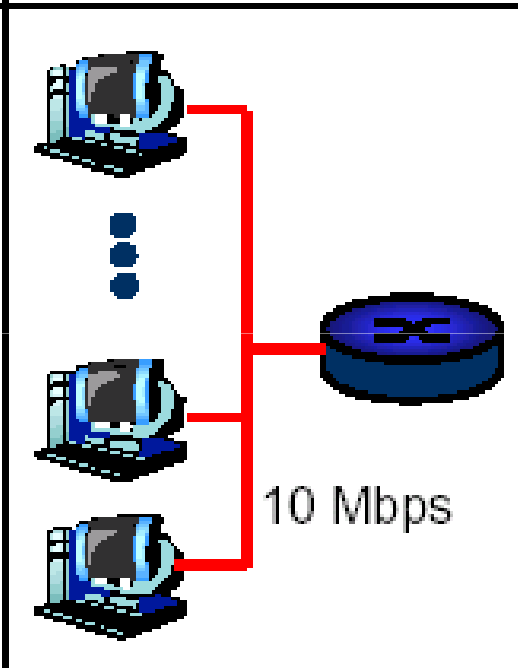
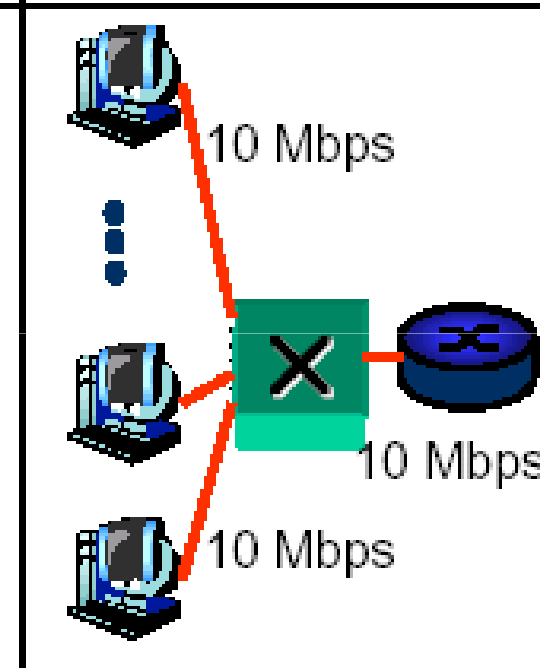
Il livello **data-link** ha la responsabilità di trasferire datagrammi da un nodo al nodo adiacente su un link



# Tassonomia dei collegamenti

Point to point	Broadcast	Switched
		
Single wire; e.g., phone line, PPP, SLIP.	Shared wire or medium; e.g., Ethernet, air (radio waves).	e.g., switched Ethernet, ATM.

# Velocità dei collegamenti

Point to point	Broadcast	Switched
		
Dedicated. Half duplex or full duplex	Shared, resource contention.	Similar to point to point.



## Contesto

- ❑ I datagrammi vengono trasferiti da differenti protocolli di collegamento tramite differenti links:
    - ad es.: Ethernet sul primo link, frame relay sui link intermedi, 802.11 sull'ultimo link
  - ❑ Ogni protocollo di collegamento fornisce servizi differenti
- ### Analogia nei trasporti
- ❑ Viaggio da Cosenza a Leeds
    - limousine: Cosenza-Lamezia
    - aereo: Lamezia-Londra
    - treno: Londra-Leeds
  - ❑ turista = **datagramma**
  - ❑ Segmento di trasporto = **communication link**
  - ❑ Tipologia di trasporto = **protocollo link layer**
  - ❑ Agente di viaggio = **routing algorithm**



## Funzioni del livello di collegamento

- ❑ Fornisce servizi al livello rete
- ❑ Realizza due funzioni principali:
  - **Livello Data link**
    - Framing
    - Controllo Errori
    - Controllo Flusso
  - **Livello Medium Access Control**
    - Nel caso di mezzo condiviso fornisce i mezzi per condividere in maniera ottimale le risorse





## Modalità di trasferimento dati

- ❑ **Il livello di collegamento si occupa del trasferimento dei dati su un link seriale**
  - **modo di trasmissione asincrono o sincrono**
  - **protocollo orientato al carattere o al bit**
  - **servizio connectionless o connection-oriented**
    - **connectionless: trame errate scartate e ritrasmissioni a carico dei livelli superiori (LAN, ISDN dove BER è basso)**



# Protocolli di livello data-link

- ❑ **classificazione**
  - **orientati al carattere**
  - **orientati al bit**
- ❑ **colloquio**
  - **half-duplex**
  - **full-duplex**
- ❑ **relazioni tra le stazioni**
  - **master-slave**
  - **peer-to-peer**



## Classificazione Protocolli

- ❑ **orientati al carattere (ormai in disuso)**
  - **informazione numerica interpretata a gruppi di 8 bit sulla base dei caratteri di un alfabeto (es. ASCII)**
- ❑ **orientati al bit**
  - **informazione senza interpretazione diretta**
  - **trasferimento trasparente dei bit**



## Utilizzo Protocolli

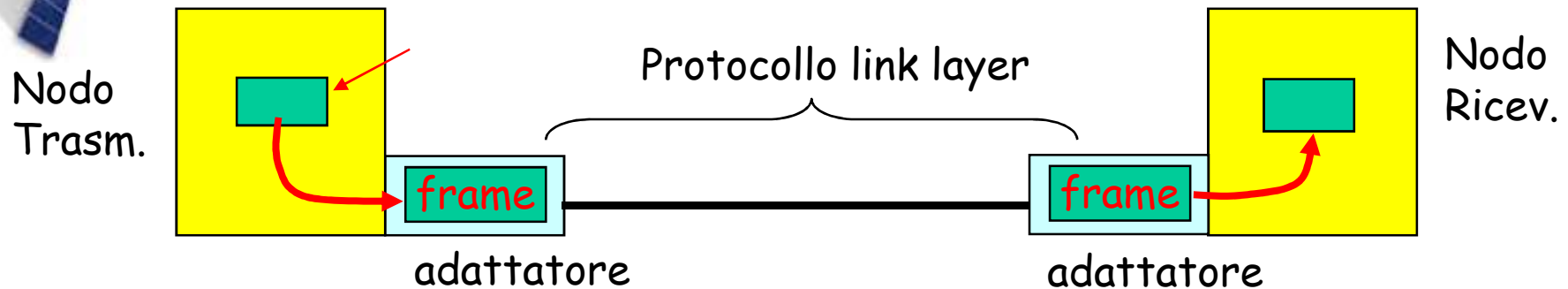
- ❑ **Il tipo di protocollo usato è funzione della separazione tra i terminali e del bit rate della linea**
  - **su linee a basso bit rate, come quelle usate coi modem, si preferisce un protocollo orientato al carattere (es. Kermit, X-Modem simplex, BSC half-duplex) che usa lo stop&wait**
  - **su linee lunghe e ad alta velocità si preferiscono i protocolli orientati al bit (es. HDLC full-duplex) che usano protocolli di controllo a finestra**



## Implementazione Protocolli

- ❑ Il protocollo del livello di link è in gran parte implementato in un “adattatore”
  - **es. scheda PCMCIA, scheda Ethernet**
  - **tipicamente contiene: RAM, chip DSP, interfaccia per il bus dell’host e un’interfaccia del link**
  - **gli adattatori sono anche conosciuti col nome di schede di interfaccia di rete (Network Interface Card, NIC)**

# Adattatori



- ❑ Il link layer è implementato nell' "adattatore"
  - Scheda Ethernet, PCMCIA, 802.11
- ❑ Lato trasmittente:
  - incapsula datagrammi in un frame
  - aggiunge bits di controllo di errore, controllo di flusso, tc.
- ❑ Lato ricevente
  - controlla errori, flusso, etc
  - estrae datagrammi e li passa al nodo ricevente
- ❑ L'adattatore è semi-autonomo



# Il sottolivello Data-Link



## Funzioni elementari di un protocollo del sottolivello data-link

- ❑ **Delimitazione e identificazione delle trame (Framing)**
- ❑ **Rivelazione degli errori trasmissivi**
- ❑ **Recupero del corretto trasferimento delle trame in caso di errore**
- ❑ **Controllo di flusso**
- ❑ **Gestione della connessione data-link (instaurazione, abbattimento e reinizializzazione)**





# Framing



## Framing

- ❑ **Nella trasmissione dati i bit trasmessi dal livello fisico sono organizzati in gruppi logici: *trame***
- ❑ **Le trame sono separate con metodi opportuni e identificate tramite un *header* dal livello di linea**
  - trame fisiche (es. TDM) e trame dati
- ❑ **Perché dividere il flusso in trame:**
  - controllo d'errore (checksum)
  - indirizzamento
  - numerazione trame, ecc.



## Delimitazione delle trame

- ❑ **la delimitazione delle trame consente al ricevitore di riconoscerne l'inizio e la fine senza ambiguità**
- ❑ **I delimitatori di trama possono essere:**
  - **sequenze di bit (flag) o caratteri speciali inseriti a inizio e fine trama**
  - **violazione del codice di linea usato a livello fisico (es. manchester)**
  - **temporizzazioni ricevute su canali di servizio**
  - **conteggio dei caratteri o dei bit**



## ...nei protocolli orientati al carattere

### **Binary Synchronous Communications (BSC)**

- **vecchio protocollo a caratteri di IBM**
- **uso di caratteri speciali per delimitare trame (SYN) e all'interno header (SOH) e dati (STX, ETX)**





## ...nei protocolli orientati al bit

- ❑ **uso del carattere DLE (Data Link Escape) prima di ogni carattere di controllo: ha significato solo la sequenza DLE-carattere di controllo**
- ❑ **i bit del flusso possono codificare casualmente anche il DLE: tecnica del character stuffing**



## Uso di flag

- ❑ **Uso di flag: sequenze di 8 bit**
- ❑ **Esempio: HDLC**
  - **sequenza di flag all'inizio e alla fine di una trama**  
***0 1 1 1 1 1 1 0***
  - **come impedire una casuale presenza della sequenza di flag nei dati**

# Bit Stuffing

informazione

da trasmettere 111100011111110001001010111111

inserimento bit di stuffing

111100011111011000100101011111011

trama

01111110 111100011111011000100101011111011 01111110

↑  
flag

↑  
flag

ricezione

~~01111110~~ 111100011111011000100101011111011 ~~01111110~~

riconoscimento  
flag d'inizio

eliminazione di un bit  
dopo 5 uno consecutivi

riconoscimento  
flag di fine



# Controllo e recupero di errori





## Rilevazione di errori

- ❑ **In ricezione è possibile che venga riconosciuta una sequenza di bit diversa da quella trasmessa (bit errati)**

**11101010011011101010001111010**

**1110001001101110101010101111010**

- ❑ **cause:**

- **rumore termico (mezzi trasmissivi, apparati di ricezione e trasmissione)**
- **interferenza da altre trasmissioni sullo stesso mezzo**
- **disturbi elettromagnetici**
- **perdite di sincronismo**
- **etc.**



## Rilevazione di errori

- ❑ **I livelli della pila possono introdurre dei rimedi agli errori collaborando ad offrire un servizio con basso tasso di errore:**
  - Trasporto: ritrasmissione
  - Collegamento: ritrasmissione
  - Fisico: correzione



## Tecniche per la protezione dagli errori di trasmissione

- ❑ **FEC (forward error correction)**
- ❑ **ARQ (automatic retransmission request)**



## Tecniche di controllo (1)

- ❑ **Correzione (FEC - forward error correction)**
  - **codifica di canale**
  - **aggiunta di ridondanza in trasmissione**
  - **uso della ridondanza in ricezione per rimediare ai bit errati (se in numero contenuto)**
  - **teoria dei codici**
  - **esempio: codice a ripetizione (N volte lo stesso bit); correzione di  $N/2 - 1$  errori**



## Tecniche di controllo (2)

- ❑ **Ritrasmissione (ARQ - Automatic Repeat reQuest)**
  - aggiunta di ridondanza in trasmissione per ogni unità informativa (FCS - Frame Check Sequence)
  - uso della ridondanza in ricezione per **rivelare** la presenza di errori e non per correggerli
  - uso di messaggi di servizio per la richiesta di ritrasmissioni o la conferma di corretta ricezione
    - Richiedono collegamenti half o full duplex



## Tecniche di controllo (3)

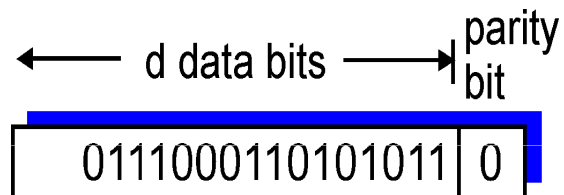
- ❑ **Ritrasmissione (ARQ - Automatic Repeat reQuest)**
- ❑ **La ridondanza richiesta per rivelare gli errori è molto più contenuta di quella richiesta per la correzione**
  - un'aggiunta di 16-32 bit ai pacchetti è di solito sufficiente
- ❑ **Il più semplice codice a rivelazione di errore è costituito dal bit di parità: alla fine del pacchetto viene aggiunto un bit pari a 1 se il numero di 1 nel pacchetto è dispari e pari a 0 se il numero di 1 è pari**



# Controllo di parità (1)

## Parità sul singolo bit:

**Rileva errori singoli sul bit**



**Schema di parità pari:**  
numero totale di '1' è pari.

**Schema di parità dispari:**  
numero totale di '1's è dispari



# Controllo di parità bidimensionale

## Bit di parità bidimensionale:

Rileva e corregge errori sul singolo bit

				row parity →
	$d_{1,1}$	$\dots$	$d_{1,j}$	$d_{1,j+1}$
	$d_{2,1}$	$\dots$	$d_{2,j}$	$d_{2,j+1}$
	$\dots$	$\dots$	$\dots$	$\dots$
	$d_{i,1}$	$\dots$	$d_{i,j}$	$d_{i,j+1}$
column parity ↓	$d_{i+1,1}$	$\dots$	$d_{i+1,j}$	$d_{i+1,j+1}$

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

*no errors*

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

parity  
error

*correctable  
single bit error*





# Internet Checksum

**Obiettivo:** rileva "errori" (ad es., flipped bits) nel segmento trasmesso

- ❑ Trasmittente:
- ❑ tratta segmenti di 16-bit interi
- ❑ checksum: somma il contenuto dei segmenti e fa il complemento ad 1 della somma
- ❑ Il trasmittente mette il valore della checksum nel campo checksum dell'UDP

## **Ricevitore:**

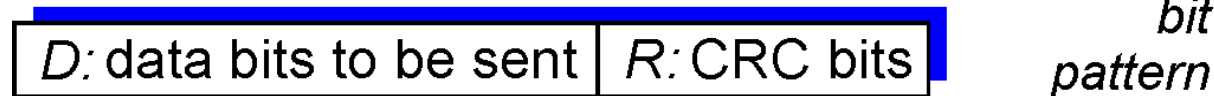
- ❑ Calcola la checksum del segmento ricevuto
- ❑ Considera se la checksum calcolata è uguale al valore del campo checksum:
  - NO – rilevato errore
  - SI – nessuno errore rilevato.



## Checksum: Cyclic Redundancy Check

- ❑ controlla data bits,  $D$ , come numero binario
- ❑ sceglie un pattern di  $r+1$  bits,  $G$
- ❑ obiettivo: sceglie  $r$  CRC bits,  $R$ , tali che
  - $\langle D, R \rangle$  esattamente divisibile per  $G$  (modulo 2)
  - il ricevitore conosce  $G$ , divide  $\langle D, R \rangle$  per  $G$ . Se il resto non è zero: rilevato errore!
  - Può rilevare tutti i burst di errori a meno di  $r+1$  bits
- ❑ molto usato nella pratica (ATM, HDCL)

← d bits → ← r bits →



$$D * 2^r \text{ XOR } R$$

*mathematical formula*

## Esempio CRC

- Vogliamo:
  - $D \cdot 2^r \text{ XOR } R = nG$
- *equivalentemente:*
  - $D \cdot 2^r = nG \text{ XOR } R$
- *equivalentemente:*
- se noi dividiamo  $D \cdot 2^r$  per  $G$ , vogliamo avere il resto  $R$

$$R = \text{resto} \quad \left[ \begin{array}{c} D \cdot 2^r \\ G \end{array} \right]$$

