Università degli Studi di Roma "Tor Vergata"
Laurea in Informatica

Sistemi Operativi e Reti
(modulo Reti)
a.a. 2024/2025

Livello di collegamento (parte1)

dr. Manuel Fiorelli

manuel.fiorelli@uniroma2.it

https://art.uniroma2.it/fiorelli

Basate sulle slide del libro di testo:
https://gaia.cs.umass.edu/kurose_ross/ppt.php

Livello di collegamento e LAN: obiettivi

- Comprendere i principi alla base dei servizi del livello di collegamento:
  - rilevazione e correzione degli errori
  - condivisione di un canale broadcast: access multiplo
  - indirizzamento a livello di collegamento
  - reti locali: Ethernet, VLAN, reti dei data center
- istanziazione e implementazione di varie tecnologie a livello di collegamento

Livello di collegamento e LAN: tabella di marcia

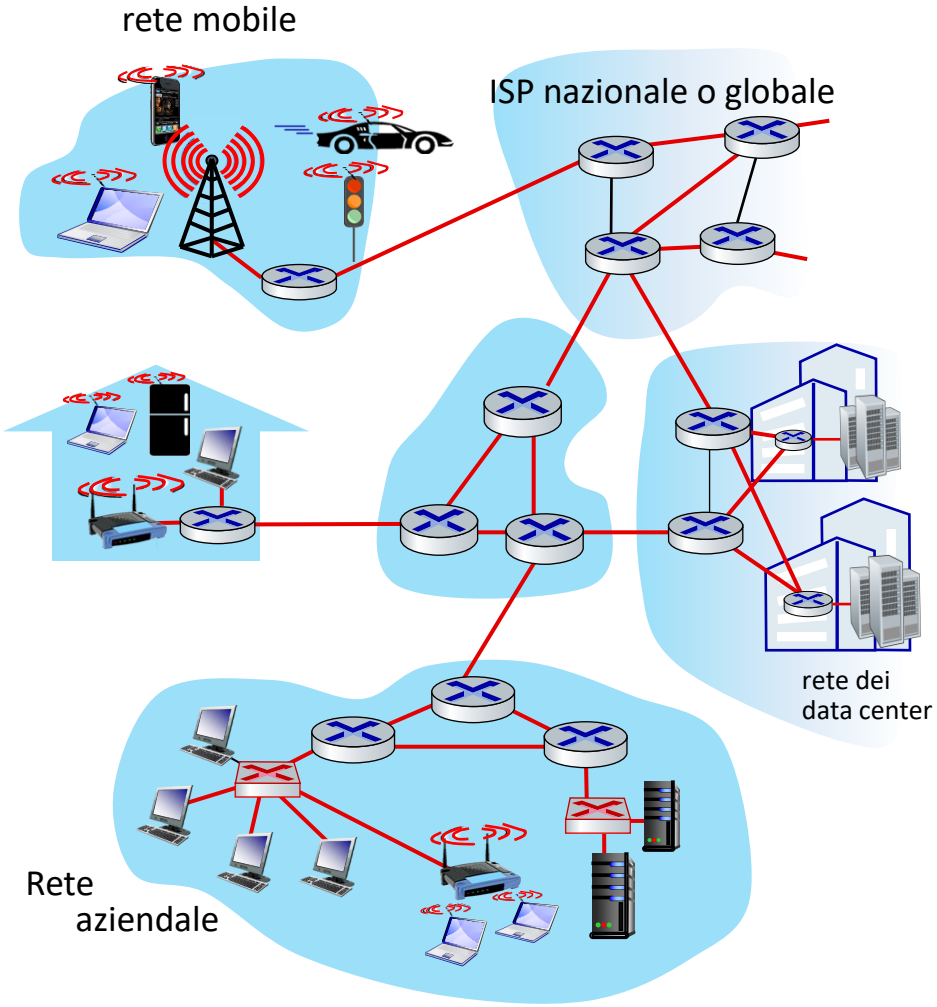
- introduzione
- rilevazione e correzione degli errori
- protocolli di acceso multiplo
- LAN
  - indirizzamento, ARP
  - Ethernet
  - switch
  - VLAN
- canali virtuali: MPLS
- Reti dei data center

- un giorno nella vita di una richiesta web

Livello di collegamento: introduzione

terminologia:

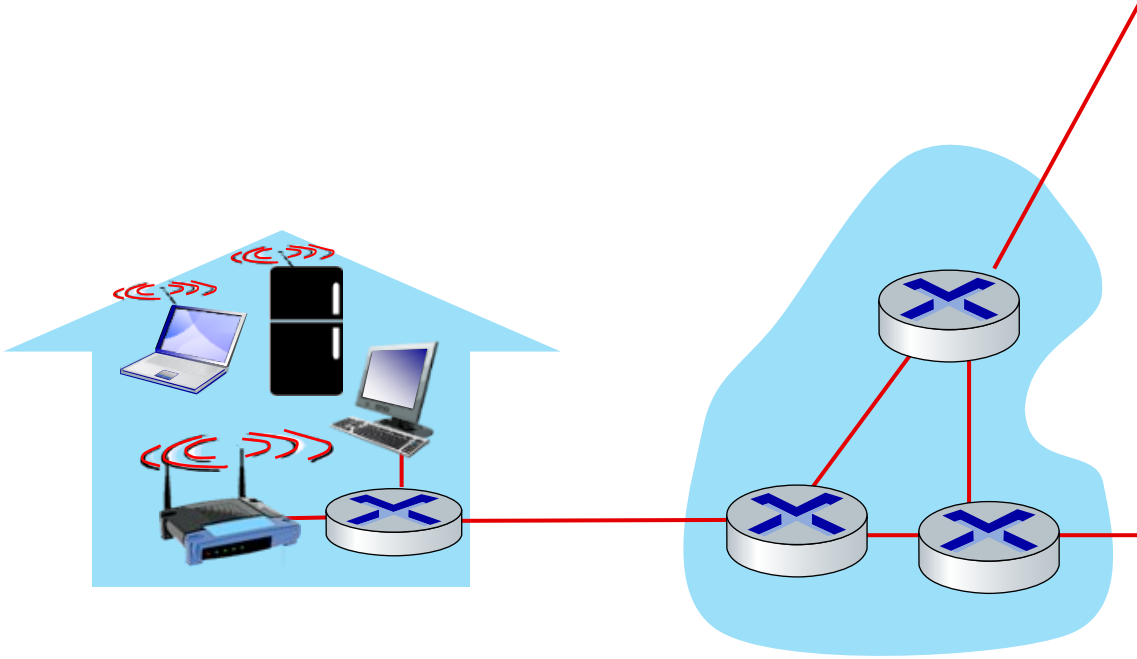
- host, router, switch, etc..: nodi
- canali di comunicazione che collegano nodi adiacenti lungo il percorso di comunicazione: collegamenti (*link*)
  - cablati, wireless
  - LAN
- pacchetto di livello 2: *frame*, incapsula datagrammi



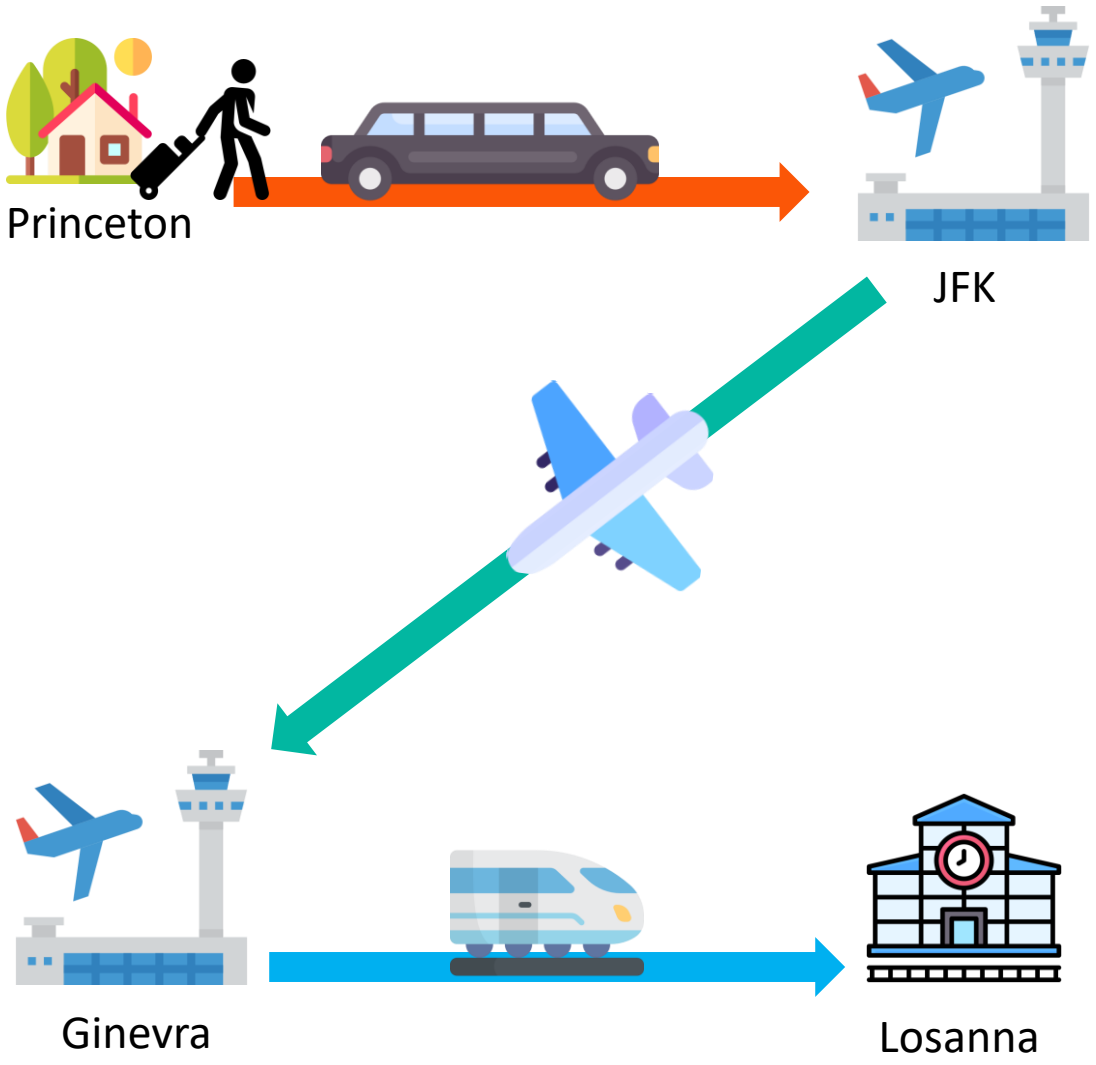
Il livello di collegamento ha la responsabilità di trasferire i datagrammi da un nodo a quello fisicamente adiacente lungo un collegamento

Livello di collegamento: contesto

- datagramma trasferito da protocolli di collegamenti differenti su collegamenti differenti:
  - es., WiFi sul primo collegamento, Ethernet sul collegamento successivo
- ciascun protocollo di collegamento fornisce servizi differenti
  - es. può o meno fornire il trasferimento di dati affidabile sul collegamento



Analogia con i trasporti

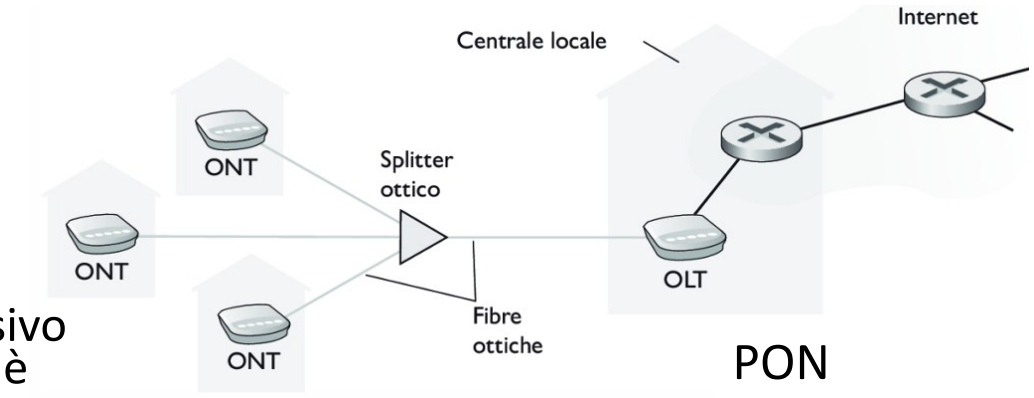


analogia con i trasporti

- viaggio da Princeton a Losanna
  - limo: da Princeton a JFK
  - aereo: da JFK a Ginevra
  - treno: da Geneva a Lausanne
- turista = datagramma
- segmento di trasporto = collegamento
- modalità di trasporto = protocollo a livello di collegamento
- agenzia di viaggi = algoritmo di instradamento

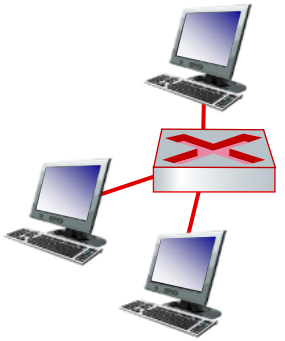
Livello di collegamento: servizi

- framing:
  - incapsula i datagrammi in frame, aggiungendo una intestazione e un trailer
- accesso al collegamento:
  - un protocollo che controlla l'accesso al mezzo trasmissivo (MAC, medium access control) se il mezzo trasmissivo è condiviso (e ci sono più mittenti)
  - indirizzi "MAC" nell'intestazione dei frame per identificare la sorgente e la destinazione (diversi dagli indirizzi IP!)
- half-duplex e full-duplex:
  - con half duplex, i nodi ad entrambi gli estremi del collegamento possono trasmettere, ma non contemporaneamente
- consegna affidabile tra nodi adiacenti
  - già sappiamo come farlo!
  - usato raramente con canali con basso tasso di errore
  - collegamenti wireless: tassi di errore elevati
    - correggere l'errore localmente anziché costringere il livello di trasporto o l'applicazione a ritrasmissioni dalla sorgente alla destinazione?



PON

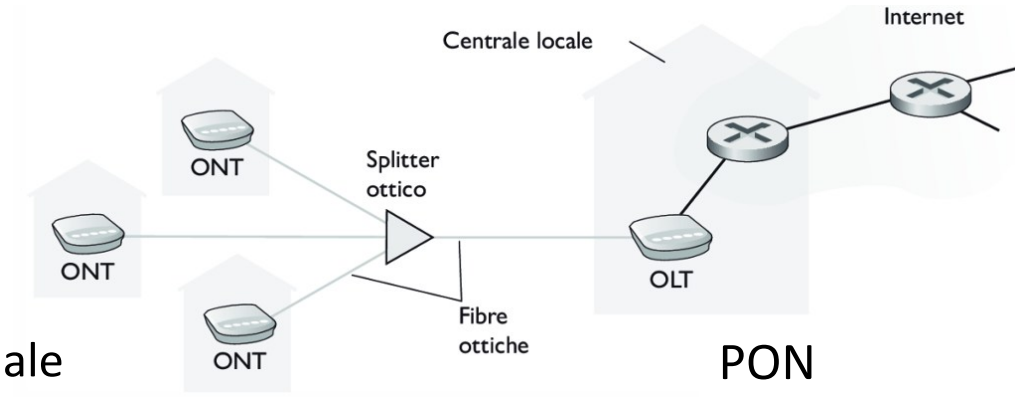

rete cellulare


LAN Ethernet


WiFi

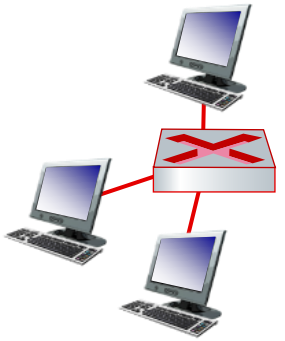
Livello di collegamento: servizi (continua)

- controllo di flusso:
  - velocità tra nodi trasmittente e ricevente adiacenti
- rilevazione e correzione degli errori:
  - gli errori sui bit sono causati da attenuazione del segnale e da rumore
  - il nodo ricevente rileva gli errori. Due approcci per la correzione:
    - ARQ (*automatic repeat request*): basato su ritrasmissioni
    - *forward error correction* (FEC, correzione degli errori in avanti): il ricevente identifica *e corregge* gli errori sui bit senza ritrasmissioni (evitando un'attesa di circa un RTT, utile per applicazioni in tempo reale o nel caso di RTT molto elevato [si pensi alla comunicazione nello spazio profondo] e al di là delle reti anche nelle applicazioni di storage).



PON

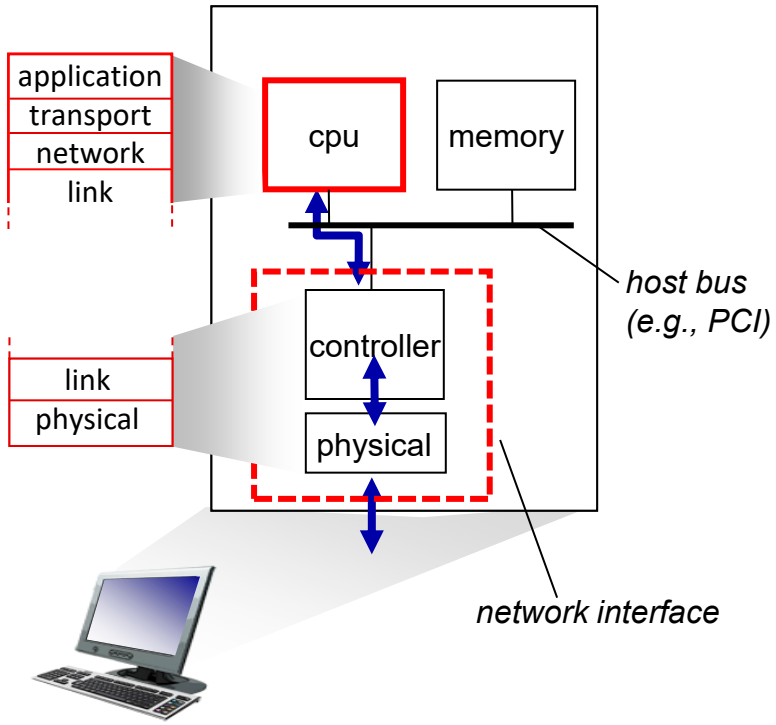

rete cellulare


LAN Ethernet

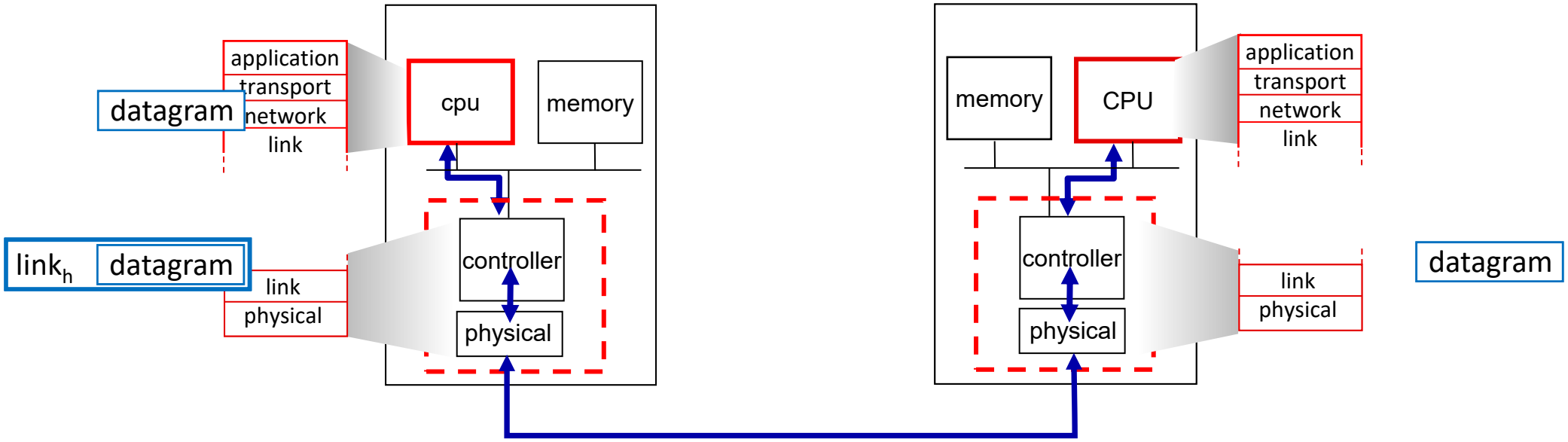

WiFi

Implementazione del livello di collegamento negli host

- in ogni singolo host
- il livello di collegamento implementato (principalmente) dall'adattatore di rete (*network adapter*) o scheda di rete (*network interface card*, NIC)
  - implementa il livello di collegamento e quello fisico
  - si collega al *bus* di sistema
- combinazione di hardware, software e firmware



Adattatore di rete negli host



Lato mittente, il controllore:

- incapsula il datagramma in un frame
- aggiunge bit di controllo degli errori, implementa il trasferimento di dati affidabile, il controllo del flusso, etc.

Lato ricevente, il controllore:

- verifica la presenza di errori e si occupa del trasferimento dati affidabile, del controllo di flusso, etc.
- estrae il datagramma e lo passa al livello superiore (quest'ultimo passato gestito in realtà dal software)

La CPU esegue la parte software del livello di collegamento, relativa a: interazione con l'adattatore di rete (come dispositivo di IO), assemblaggio delle informazioni di indirizzamento (lato mittente), gestione di condizioni di errore e il passaggio del datagramma fino al livello di rete (lato ricevente).

Livello di collegamento e LAN: tabella di marcia

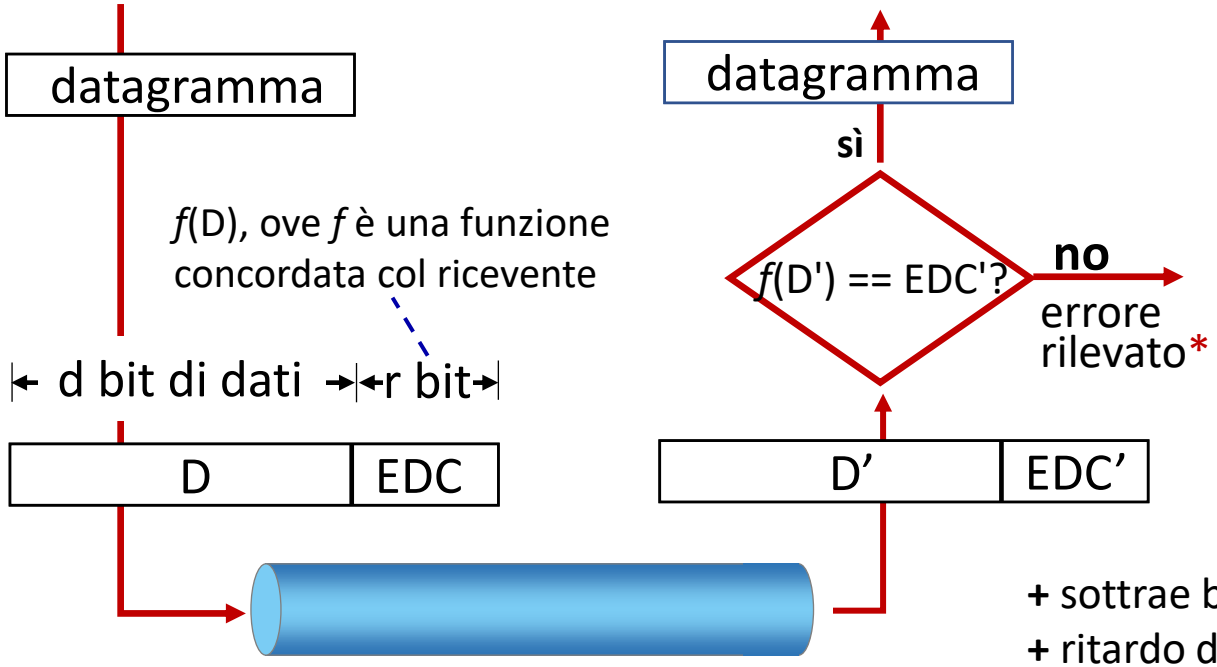
- introduzione
- rilevazione e correzione degli errori
- protocolli di acceso multiplo
- LAN
  - indirizzamento, ARP
  - Ethernet
  - switch
  - VLAN
- canali virtuali: MPLS
- Reti dei data center

- un giorno nella vita di una richiesta web

Rilevazione degli errori

EDC: error detection and correction

D: dati protetti dal controllo d'errore, può includere i campi di intestazione



*è possibile rilevare errori sia nei dati sia nei bit EDC

Rilevazione non affidabile al 100%!

- **errori non rilevati**
- occorre ridurne la probabilità:
  - più bit di EDC
  - calcoli più complessi

overhead

+ sottrae banda ai dati utili
+ ritardo di elaborazione (mitigato se implementato in hardware)

rispetto al livello di trasporto, il livello di collegamento utilizza tecniche più complesse implementate in hardware

Considerazioni generali (1)

Assumendo errori casuali (non introdotti ad arte) e supponendo che la funzione di calcolo dell'EDC offra una sufficiente capacità di diffusione, possiamo ***in prima approssimazione*** stimare la probabilità condizionata che un errore — una volta verificatosi — non venga rilevato in 2^{-r} (assumendo cioè che i valori di EDC siano distribuiti in maniera uniforme).

Questa è solo una *rule of thumb* che ci aiuta a capire perché è utile incrementare il numero *r* di bit EDC. Tuttavia, non descrive le prestazioni di uno specifico EDC soprattutto se si considerano assunzioni più realistiche sulla distribuzione degli errori.

Considerazioni generali (2)

Per valutare la robustezza dei codici di rilevamento degli errori (EDC), è infatti importante tenere presenti i seguenti aspetti:

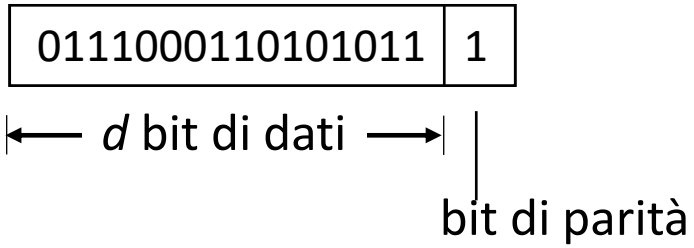
- la probabilità di errore su un bit è $<<$ 1 (molto minore di): è assai più probabile che un bit sia ricevuto correttamente piuttosto che essere alterato
- gli errori non sono indipendenti ma spesso si manifestano in raffiche (burst)
- certi pattern di errore potrebbero essere rilevati con certezza mentre altri potrebbero passare sicuramente inosservati

Per questo, nel seguito, analizzeremo il comportamento degli EDC rispetto a diversi pattern di errore.

Controllo di parità

Singolo bit di parità (parity bit):

- Rileva un numero *dispari* di errori



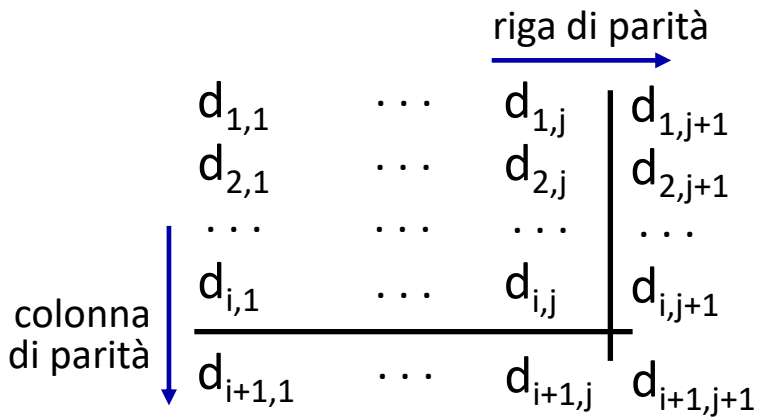
Parità pari/dispari: imposta il bit di parità in modo che ci sia un numero pari/dispari di 1

Il ricevente:

- calcola la parità dei *d* bit ricevuti
- lo confronta con il bit di parità ricevuto – se differente, allora è stato rilevato un errore

Parità bidimensionale:

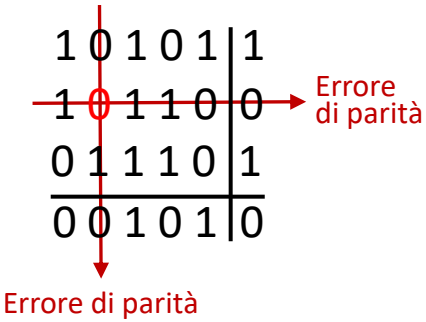
- rileva tutte le combinazioni di al più 3 errori
- rilevazione e *correzione* di errori singoli



Senza errori:

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

Errore su un singolo bit rilevato e correggibile:



* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Controllo di parità

Senza errori:

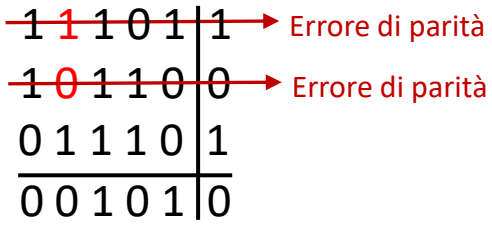
1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

Due errori sulla stessa riga, rilevati su colonne differenti

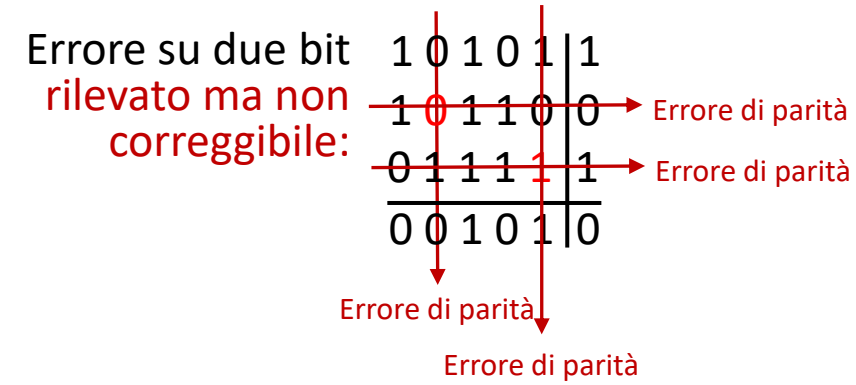
Errore su due bit rilevato ma non correggibile:

Due errori sulla stessa colonna, rilevati su righe differenti

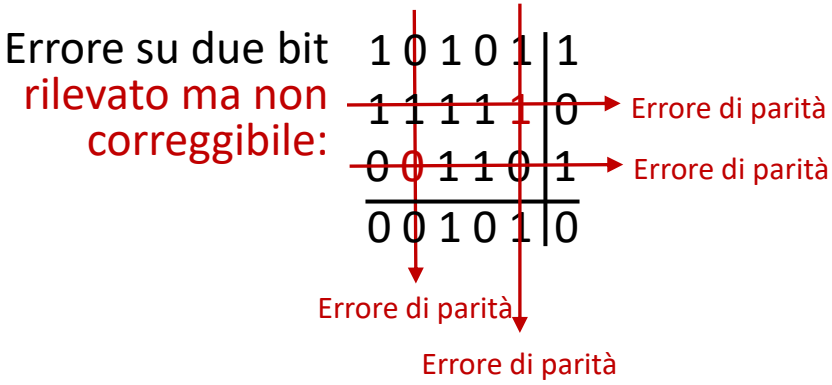
Errore su due bit rilevato ma non correggibile:



Errore su due bit rilevato ma non correggibile:



Errore su due bit rilevato ma non correggibile:



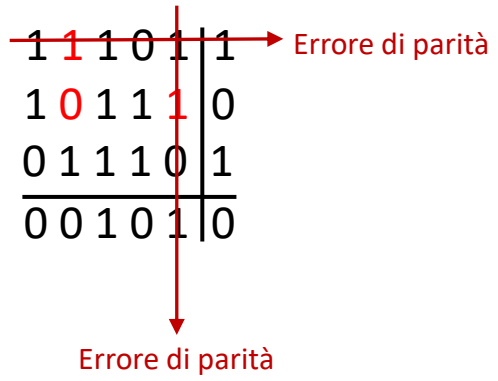
Si noti come i due pattern di errore qui sopra sono differenti, ma ciononostante abbiamo prodotto gli stessi errori di parità! L'ambiguità tra i due casi ci impedisce di correggere l'errore.

Controllo di parità

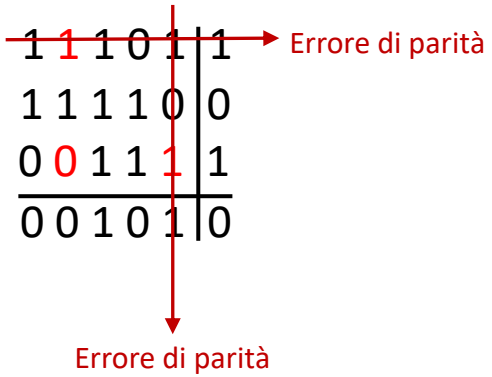
Senza errori:

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

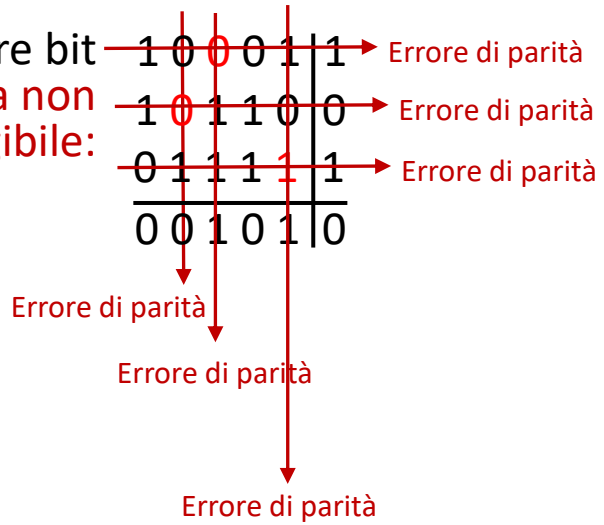
Errore su tre bit rilevato ma non correggibile:



Errore su tre bit rilevato ma non correggibile:



Errore su tre bit rilevato ma non correggibile:



Controllo di parità

Errore su quattro bit non rilevato

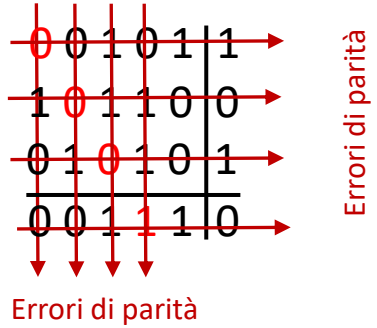
1	0	1	0	1	1
1	0	1	1	1	0
0	0	1	1	1	1
0	0	1	0	1	0

Senza errori:

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

Ma non significa che non si possano rilevare *alcuni* errori su quattro bit

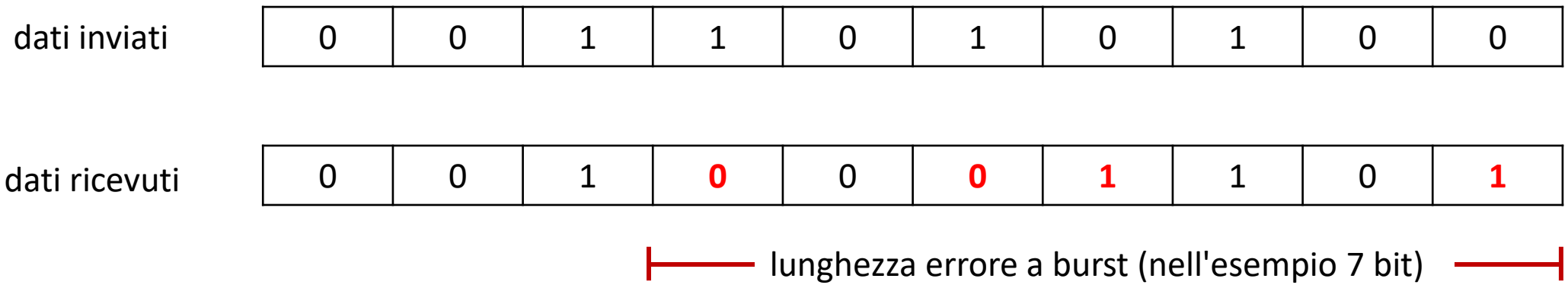
Errore su quattro bit rilevato ma non correggibile:



Controllo di parità

- adatto quando entrambe le seguenti condizioni sono vere:
  - la probabilità di errori nei bit è bassa
  - gli errori sono indipendenti

  Sotto queste ipotesi, la probabilità di errori multipli è molto bassa: pertanto è improbabile che si verifichi un numero pari di errori non rilevati dal bit di parità.
- nella realtà, però, gli errori tendono a verificarsi in *burst*
  - la probabilità che errori a burst non siano rilevati da un singolo bit di parità può avvicinarsi al 50%



Checksum Internet (ripasso, si veda sezione 3.3)

Obiettivo: rilevare gli "errori" (*bit alternati*) nel segmento trasmesso

mittente:

- tratta il contenuto del segmento come una sequenza di interi a 16 bit (inclusi i campi dell'intestazione UDP e gli indirizzi IP)
- checksum: complemento a 1 della somma (in complemento a 1) della sequenza di interi a 16 bit
- pone il valore del checksum nel campo checksum del segmento UDP

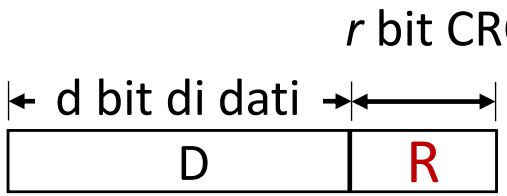
ricevente:

- calcola la somma in complemento a 1 allo stesso modo del mittente, includendo però il checksum ricevuto
- il risultato è costituito da tutti bit 1 (-0 nell'aritmetica del complemento a 1)?
  - Sì - nessun errore rilevato
  - No - errore rilevato
- oppure, si esegue il complemento a 1 finale: si è calcolato il checksum di tutti i dati ricevuti (incluso il checksum stesso) e si verifica che sia formata da soli 0

Codici di controllo a ridondanza ciclica (Cyclic Redundancy Check, CRC)

- codifica di rilevamento degli errori più potente
- D: *dati* da trasmettere (d bit)
- G: sequenza di (r + 1) bit concordata, detta *generatore* (definito nello standard CRC)

il bit più significativo deve essere 1; nei generatori di uso pratico, lo è anche quello meno significativo



bit da inviare

$<D,R> = D * 2^r$ XOR R

formula per questi bit

mittente: calcola *r* bit CRC, R, tali che <D,R> si *divisibile esattamente* da G (mod 2)

- il ricevente conosce G, divide <D,R> per G. Se il resto è diverso da zero: errore rilevato!
- può rilevare tutti gli errori a burst di lunghezza inferiore a r+1 bit (ovvero, errori di non più di r bit consecutivi)
- la frazione dei burst più lunghi che può rilevare è approssimativamente $1 - 2^{-r}$
- largamente usato in pratica (Ethernet, 802.11 WiFi)

Codici di controllo a ridondanza ciclica (Cyclic Redundancy Check, CRC)

tutti i calcoli di CRC sono eseguiti in aritmetica modulo 2 senza riporti nelle addizioni e prestiti nelle sottrazioni

- addizione e sottrazione sono la stessa operazione, corrispondente all'or esclusivo (*exclusive or*, XOR) bit a bit

```
1011 +        1011 -
1101 =        1101 =
------        ------
0110          0110
```

1011 XOR 1101 = 1011 $\oplus$ 1101 = 0110

- la moltiplicazione e la divisione sono calcolate come al solito, usando queste definizioni di addizione e sottrazione

```
1011 x
 101 =
------
1011 +
   0 +
1011  =
------
100111
```

Questa "strana" aritmetica corrisponde al vedere le sequenze di bit come i coefficienti (modulo 2) di polinomi

Codici di controllo a ridondanza ciclica (Cyclic Redundancy Check, CRC)

assumendo che il primo bit sia 1, allora il *grado* del polinomio è uguale al numero di bit - 1

1011 x → $x^3 + x + 1$

101 = → $x^2 + 1$

1011 + → $x^3 + x + 1$

0 +

1011 = → $(x^3+x+1)\cdot x^2 = x^5 + x^3 + x^2$

100111 → $x^5 + x^2 + x + 1$

$= (x^5 + x^3 + x^2) + (x^3 + x + 1)$

$= x^5 + (1+1)\,x^3 + x^2 + x + 1$

$= x^5 + (1+1)\,x^3 + x^2 + x + 1$

$= x^5 + x^2 + x + 1$

modulo 2

Codici di controllo a ridondanza ciclica: esempio

Il mittente vuole calcolare R tale che:

$D \cdot 2^r \ XOR \ R = nG$

... o equivalentemente (XOR R in entrambi i lati):

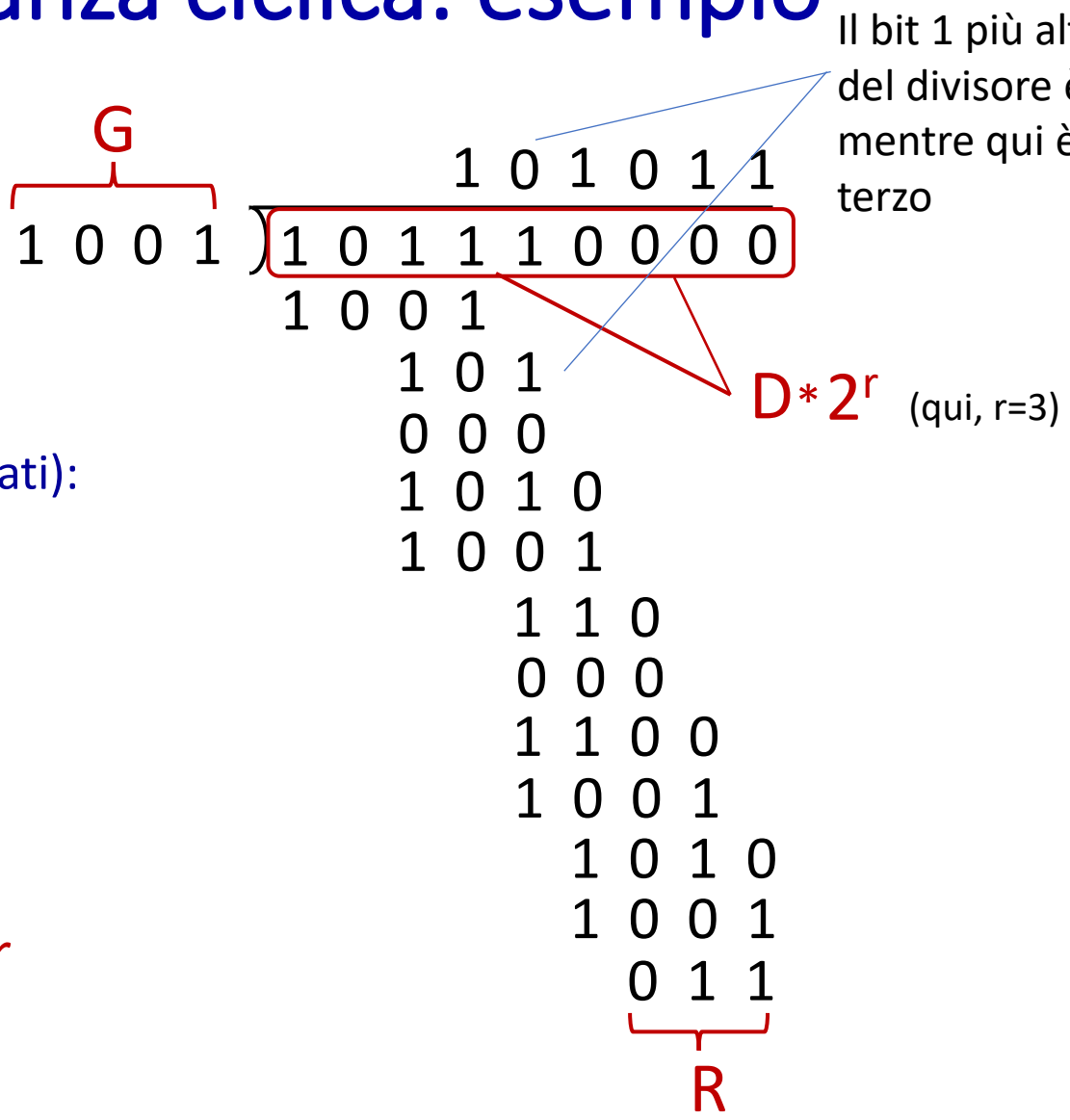
$D \cdot 2^r = nG \ XOR \ R$

... che ci dice che:

se dividiamo $D \cdot 2^r$ per G il resto è precisamente R

$$R = \text{resto}\left[\frac{D \cdot 2^r}{G}\right]$$

algoritmo per calcolare R



* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

CRC: polinomio di errore e scelta del generatore

il mittente invia $T = D\ 2^r$ XOR R

il ricevente riceve T' = T XOR E = T + E (usando l'addizione CRC)

dove E è una sequenza di bit (ovvero un polinomio), i cui bit a 1 indicano dove si è verificato un errore.

Siccome T è divisibile per G, T' sarà divisibile per G se e solo se E è divisibile per G.

Quindi, *G deve essere scelto in modo tale che NON divida i polinomi di errore*.

In pratica, si cerca di definire G in modo che rilevi diversi tipi di errore.

CRC: polinomio di errore e scelta del generatore

Se G ha un numero pari di bit a 1, allora è in grado di rilevare qualsiasi numero dispari di errori.

Perché?

- $E(x)$ il polinomio che rappresenta l'errore (con un numero **dispari** di bit a 1),
- $G(x)$ il polinomio generatore (con un numero **pari** di bit a 1).

Supponiamo per assurdo che l'errore passi inosservato, cioè che

$E(x) = Q(x) \cdot G(x)$ per qualche polinomio $Q(x)$.

Valutiamo entrambi i lati in $x = 1$

- poiché $E(x)$ contiene un numero dispari di bit a 1, si ha $E(1) = 1$
- poiché $G(x)$ contiene un numero pari di bit a 1, si ha $G(1) = 0$.

Ne segue

$E(1) = Q(1) \cdot G(1) \Longrightarrow 1 = Q(1) \cdot 0$, che è una contraddizione.

Quindi $E(x)$ **non** è divisibile per $G(x)$ e pertanto l'errore sarà sicuramente rilevato

Nota: il caso più semplice è $G(x) = x + 1$ (in bit "1"), che corrisponde al classico controllo di parità.

CRC: polinomio di errore e scelta del generatore

Se G ha almeno due bit a 1, esso è in grado di rilevare qualunque errore singolo.

Perché?

Un errore singolo è espresso dal polinomio x^k(cioè $1\overbrace{0000\cdots0}^{k}$)che è divisibile solo dai polinomi x^i per $i \leq k$, essendo $x^k = x^{k-i} \cdot x^i$.