

# Architettura di Internet

Luciano Bononi

[bononi@cs.unibo.it](mailto:bononi@cs.unibo.it)

<http://www.cs.unibo.it/~bononi/>

Figure credits: some of the figures have been taken from the book: Kurose, Ross, Internet e reti di calcolatori

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## The Data Link Layer

### Obiettivo:

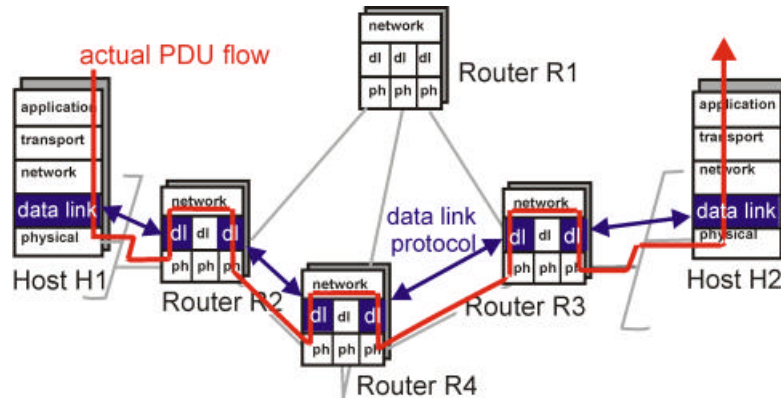
- ✦ Capire i principi dei servizi Data Link:
  - ✦ error detection, correction
  - ✦ Condivisione di broadcast channel: accesso multiplo
  - ✦ Indirizzamento fisico
  - ✦ Trasferimento dati affidabile, controllo di flusso
- ✦ Esempi di tecnologie di livello Data Link alternative

### Sommario

- ✦ Servizi link layer
- ✦ error detection, correction
- ✦ Protocolli di accesso multiplo e reti locali (LAN)
- ✦ Indirizzamento link layer, ARP
- ✦ Tecnologie di livello Link:
  - ✦ Ethernet
  - ✦ hubs, bridges, switches
  - ✦ IEEE 802.11 LANs
  - ✦ PPP

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

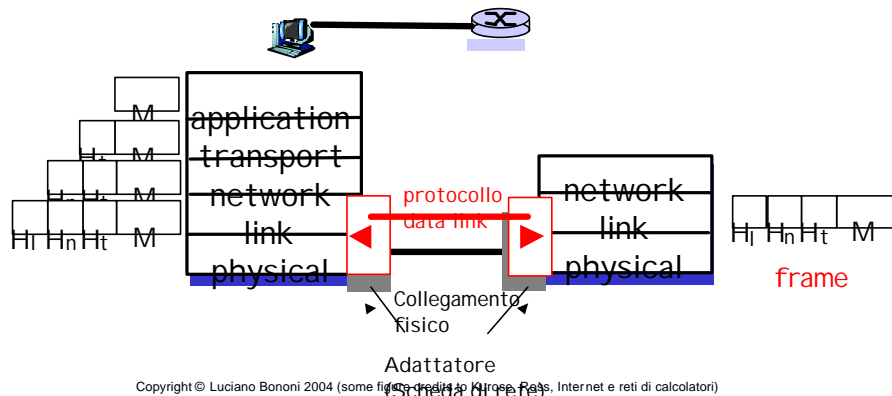
## Link Layer: (livello collegamento)



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Link Layer: definizione del contesto

- ⚡ Due dispositivi *connessi fisicamente*:
  - ⚡ host-router, router-router, host-host
- ⚡ Unità di dati: *frame*



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Servizi del livello collegamento

### ✍ Framing, link access:

- ✍ incapsulare datagram (IP) in frame
  - Aggiungere "la busta": header, trailer
- ✍ Se il mezzo è condiviso si implementa il MAC opportuno
- ✍ "Indirizzo Fisico" (MAC address) usato nei frame headers per identificare mittente e destinatario
  - Diverso dall'indirizzo IP (logico)

### ✍ Consegna affidabile tra due dispositivi direttamente connessi

- ✍ Tecniche usate poco se il collegamento è molto affidabile (fibre ottiche, alcune tecnologie twisted pair)
- ✍ Collegamenti wireless: alta frequenza di errori
  - Q: perchè serve un meccanismo sia a livello collegamento che end-to-end?

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Servizi del livello collegamento

### ✍ Controllo di flusso:

- ✍ Regola la velocità del mittente rispetto alla capacità di ricezione del destinatario

### ✍ Error Detection:

- ✍ Errori causati da attenuazione, rumore.
- ✍ Il ricevente rileva la presenza di errori sui bit:
  - Richiede invio di duplicato, e/o elimina il frame errato

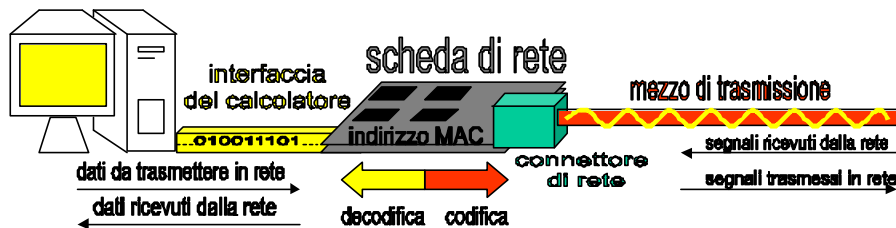
### ✍ Error Correction:

- ✍ Il ricevente identifica *e corregge* errori sul/i bit senza richiedere una ri-trasmissione

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Link Layer: Implementazione

- ✍ Si realizza nell'adattatore (scheda di rete)
  - ✍ e.g., PCMCIA card, scheda Ethernet
  - ✍ Scheda di rete: RAM, chip DSP, interfaccia bus verso il PC e interfaccia di rete verso il mezzo trasmissivo

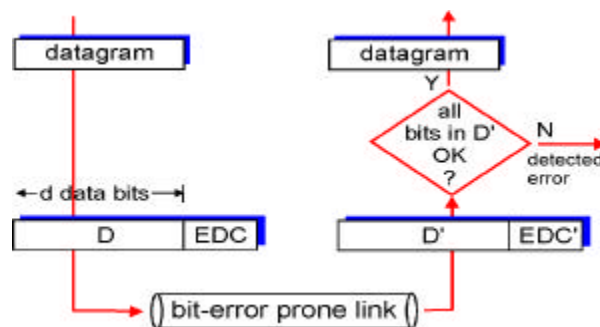


Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Rilevazione di errori (Detection)

EDC= Error Detection and Correction bits (bit ridondanti)  
D = Dati protetti dalla correzione di errore, possono includere headers

- Error detection non è affidabile al 100%
  - il protocollo può non accorgersi di errori, ma ciò è raro!
  - campi EDC più ampi permettono rilevazione e correzione migliore

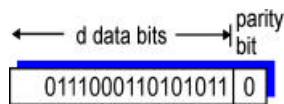


Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Controllo di parità: Parity Checking

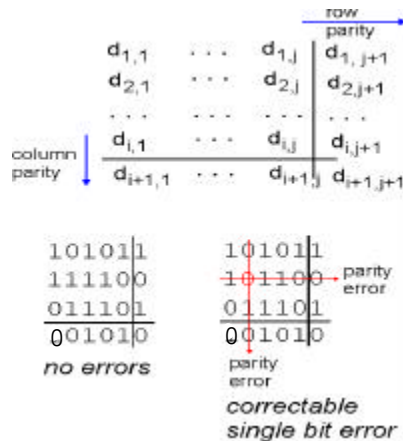
### Parità a singolo bit:

Rileva errori singoli su un bit



### Parità bidimensionale:

Rileva e *corregge* errori singoli e non



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Internet checksum

**Obiettivo:** rilevare errori su bit in una sequenza di bit, ma si usa a livello 4 (trasporto)

### Mittente:

- ⌘ Considera la sequenza di bit come interi a 16 bit
- ⌘ checksum: esegue la somma dei valori della sequenza complementati a uno
- ⌘ Il valore finale della somma viene posto nel campo UDP checksum e spedito insieme ai bit

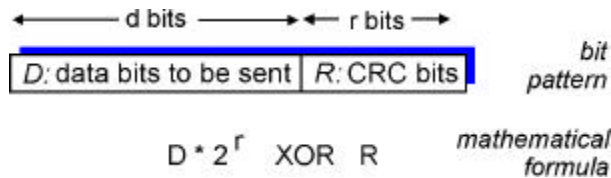
### Destinatario:

- ⌘ Calcola il checksum sui bit ricevuti
- ⌘ Controlla se il valore ottenuto è uguale a quello indicato dal mittente
  - ⌘ NO - errore rilevato!
  - ⌘ YES - non si rilevano errori (ma potrebbero essercene comunque...)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Cyclic Redundancy Check

- ✦ Siano considerati i bit di dati,  $D$ , come un numero binario
- ✦ Siano scelti una sequenza di  $r+1$  bit (generatore),  $G$
- ✦ obiettivo: trovare  $r$  bit detti **CRC**, tali che
  - ✦  $\langle D, \text{CRC} \rangle$  sia divisibile per  $G$  (modulo 2)
  - ✦ Anche il destinatario conosce  $G$ , e dopo avere ricevuto  $\langle D, \text{CRC} \rangle$  divide  $\langle D, \text{CRC} \rangle$  per  $G$ . Se il resto non è zero: errore rilevato!
  - ✦ Vantaggio: possono essere rilevati tutte le sequenze errate fino a  $r+1$  bits
- ✦ Molto usato in pratica (es. ATM, HDLC)



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## \*CRC Example

Want:

$$D \cdot 2^r \text{ XOR } R = nG$$

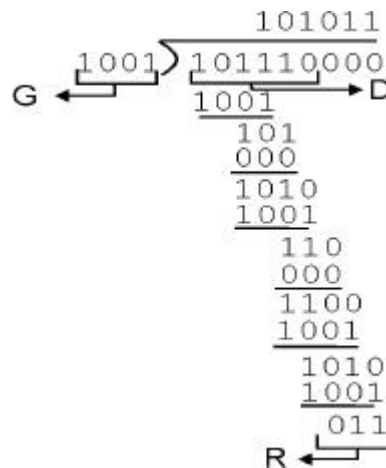
equivalently:

$$D \cdot 2^r = nG \text{ XOR } R$$

equivalently:

if we divide  $D \cdot 2^r$  by  $G$ , want remainder  $R$

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



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Link e protocolli di accesso (multiplo) al mezzo trasmissivo (MAC)

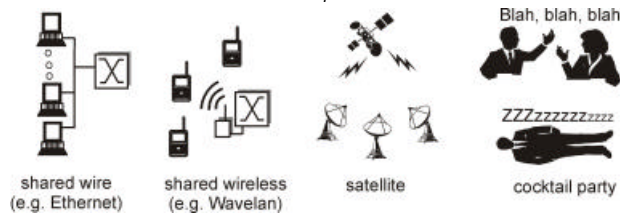
Tre tipi di "links":

✍ point-to-point (es. cavetto singolo)

✍ Es. Protocolli PPP, SLIP)

✍ **broadcast** (cavetto o mezzo condiviso)

✍ es. Protocolli Ethernet, IEEE802.11 Wi-Fi



✍ commutato (es. switched Ethernet, ATM...)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Protocolli MAC e accesso multiplo: canali broadcast

✍ Esiste un singolo canale di comunicazione condiviso

✍ Le reti LAN si basano molto su questa ipotesi (costi).

✍ Due o più trasmissioni simultanee causano collisione!

✍ Solo un host alla volta può trasmettere sul canale condiviso

✍ **Protocolli MAC ad accesso multiplo:**

✍ Algoritmi distribuiti che determinano come le stazioni condividono il canale, cioè: quando trasmette la stazione x?

• N.B. mettersi d'accordo per trasmettere equivale a dover trasmettere!!!

✍ Fattori critici dei protocolli MAC di accesso multiplo:

• Sincroni o asincroni

• Che informazione è necessario conoscere riguardo al resto del mondo?

• Robustezza (rispetto agli errori del canale)

• Prestazioni

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## \*Multiple Access protocols

- ✍ claim: humans use multiple access protocols all the time
- ✍ class can "guess" multiple access protocols
  - ✍ multiaccess protocol 1:
  - ✍ multiaccess protocol 2:
  - ✍ multiaccess protocol 3:
  - ✍ multiaccess protocol 4:

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## MAC Protocols: tassonomia

Esistono tre classi generali di protocolli MAC:

- ✍ **Suddivisione di canale (Channel Partitioning)**
  - ✍ Il canale è diviso "a fette" (time slots, frequency bands)
  - ✍ Ogni fetta viene allocata a un nodo per suo uso esclusivo
  - ✍ Protocolli rigidi rispetto ad esigenze "leggere"
- ✍ **Accesso causale (Random Access)**
  - ✍ Possono nascere collisioni, ma sono semplici e "resistenti"
  - ✍ In caso di collisioni si deve rimediare: si spreca risorse
- ✍ **Prenotazione ciclica o "a turni" ("Taking turns")**
  - ✍ L'accesso viene coordinato da leggi che evitano collisioni
  - ✍ Sono protocolli rigidi e poco resistenti a guasti

**Obiettivi:** efficiente, equo, semplice, distribuito

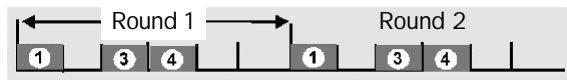
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)



## MAC protocols e suddivisione canale: TDMA

### TDMA: time division multiple access

- ✍ Si accede al canale (diviso a fette) in "rounds"
- ✍ TDM (Time Division Multiplexing): si divide il tempo del canale in slot
  - ✍ Ogni stazione ottiene l'assegnazione di uno slot di lunghezza fissa (pari alla dimensione del frame) in ogni round



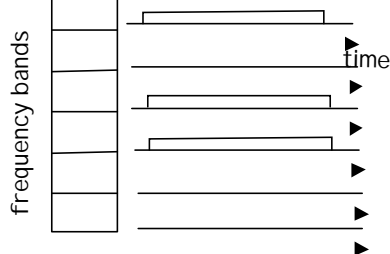
- ✍ Slot inutilizzati rimangono vuoti (va male se stazioni trasmettono molto, ma di rado)
- ✍ Es.: LAN con 6 host, 1,3,4 hanno dati, 2,5,6 non hanno dati

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## MAC protocols e suddivisione canale : FDMA

### FDMA: frequency division multiple access

- ✍ FDM (Frequency Division Multiplexing): La banda totale del canale (spettro delle frequenze) è diviso in bande
- ✍ Ogni stazione vede assegnata una banda diversa per tutto il tempo del collegamento
- ✍ Le bande possono non essere usate in certi intervalli di tempo
- ✍ Es. precedente LAN, 1,3,4 trasmettono, 2,5,6 sprecano le loro bande non avendo dati da trasmettere



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

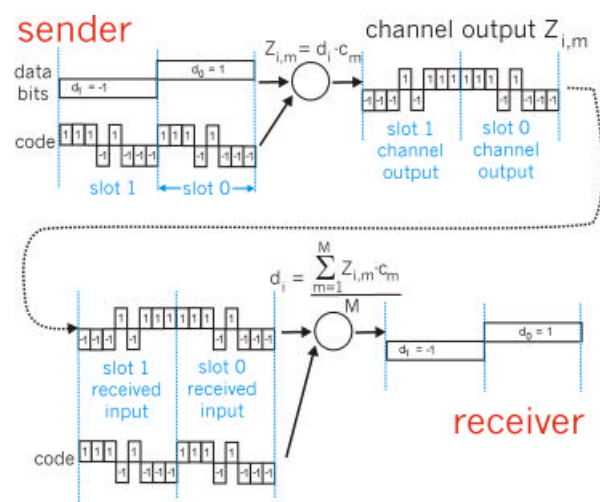
## MAC con suddivisione canale (CDMA)

### CDMA (Code Division Multiple Access)

- ✗ “Codice” unico assegnato ad ogni utente (partizione di codice)
- ✗ Usato in canali wireless broadcast (reti cellulari, satellite,...)
- ✗ Tutti gli utenti usano lo stesso spettro di frequenza, ma ogni utente usa la propria “chipping sequence” (il codice) per codificare i dati
- ✗ *Segnale codificato* = (dati) X (chipping sequence)
- ✗ *decodifica*: prodotto interno dei segnali codificati e della chipping sequence
- ✗ Permette a utenti multipli di “coesistere” e trasmettere simultaneamente evitando interferenza (se i codici sono “ortogonali”)

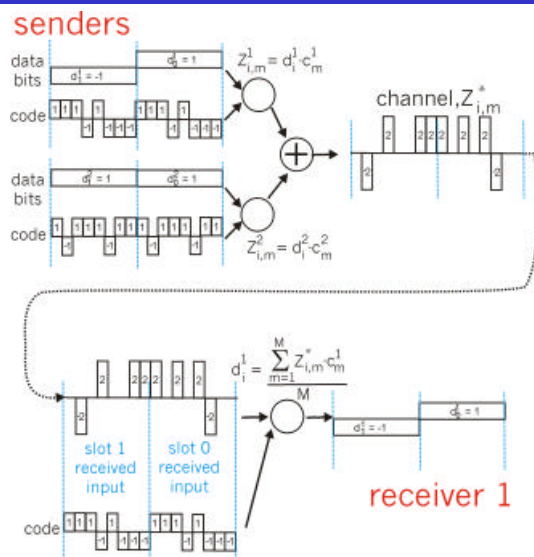
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## CDMA: Codifica/Decodifica



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## CDMA: interferenza tra due sender



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

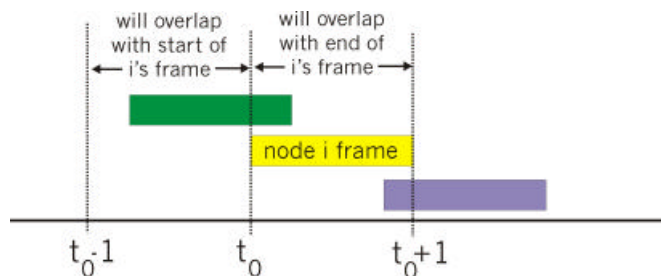
## Protocolli MAC Random Access

- ✍ Quando un nodo ha pacchetti da spedire
  - ✍ Trasmette sfruttando a pieno il data rate R.
  - ✍ Non c'è coordinamento precedente alle trasmissioni
- ✍ Due o più trasmissioni ricevute -> "collisione",
- ✍ **Protocollo MAC random access** specifica:
  - ✍ Come rilevare le collisioni
  - ✍ Come evitare o risolvere le collisioni (es. Ritardando i tentativi)
- ✍ Esempi di random access MAC protocols:
  - ✍ ALOHA, slotted ALOHA
  - ✍ CSMA, slotted CSMA, CSMA/CA and CSMA/CD

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

# ALOHA

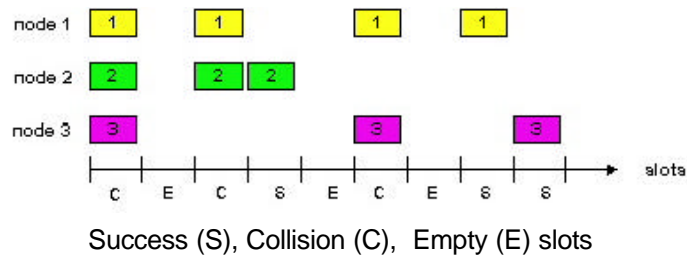
- ✍ Aloha: è il più semplice (non sincronizzato)
- ✍ Se c'è un pacchetto da spedire
  - ✍ spedisce appena possibile
- ✍ Vulnerabilità del pacchetto
  - ✍ Spedito a  $t_0$  collide con quelli spediti in  $[t_0-1, t_0+1]$



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

# Slotted Aloha

- ✍ Tempo suddiviso in slot = dimensione frame
- ✍ Se un nodo deve trasmettere trasmette all'inizio dello slot successivo
- ✍ Se c'è collisione: ritrasmette in uno slot successivo con probabilità  $p$ , finché non riceve Ack (ARQ)



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## CSMA: Carrier Sense Multiple Access

**CSMA:** ascolta prima di trasmettere

- ⌘ Se il canale è libero: trasmette il frame
- ⌘ Se il canale è occupato: aspetta
  - ⌘ **Persistent CSMA:** riprova subito con probabilità  $p$  appena il canale diventa libero
  - ⌘ **Non-persistent CSMA:** riprova dopo un tempo casuale
- ⌘ Analogia umana: non interrompe chi parla
- ⌘ Usato per Ethernet (CSMA/CD) e IEEE 802.11 (CSMA/CA)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## CSMA collisions

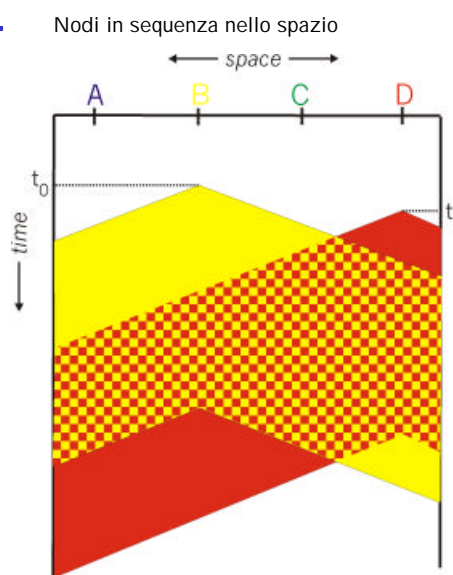
Una collisione si verifica:

A causa del ritardo di propagazione due nodi possono iniziare insieme senza rilevarsi

**Collisione**

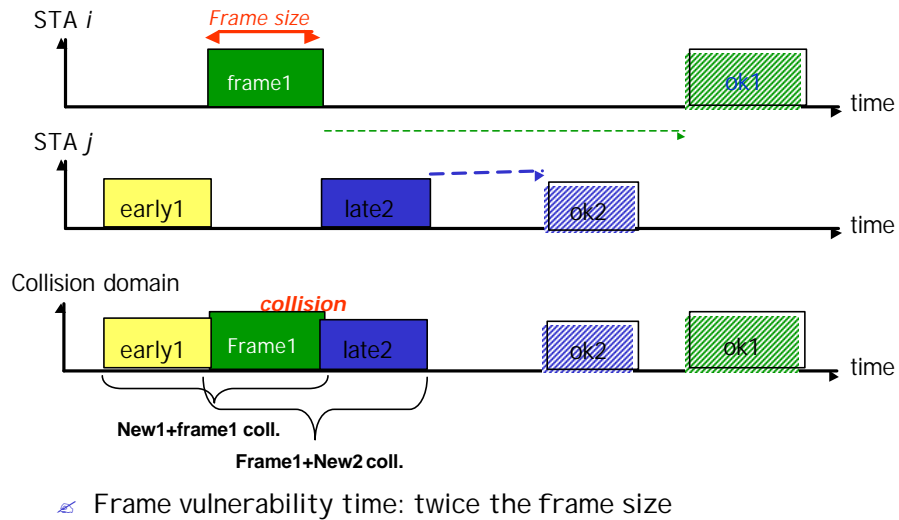
Va perduto tutto il frame e tutto il tempo di trasmissione

Distanza e probabilità di collisione sono in relazione tra loro



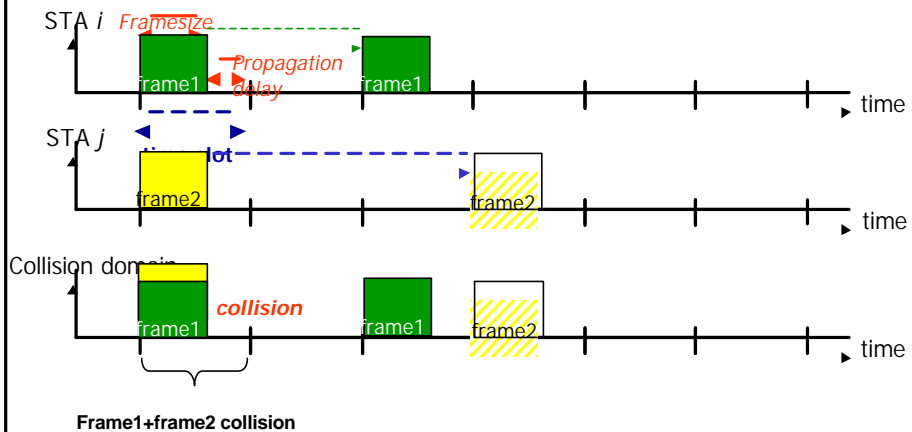
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## The ALOHA protocol



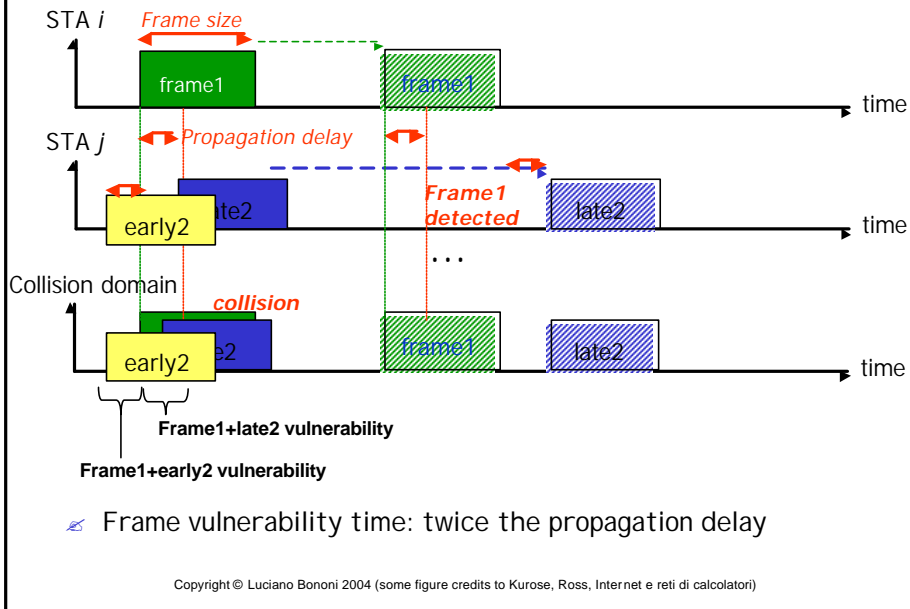
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Slotted ALOHA

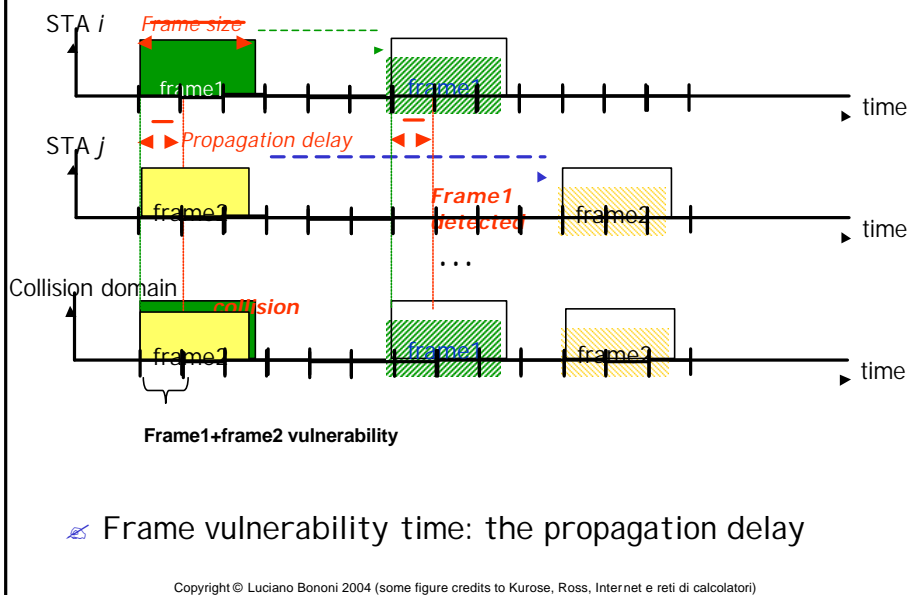


Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

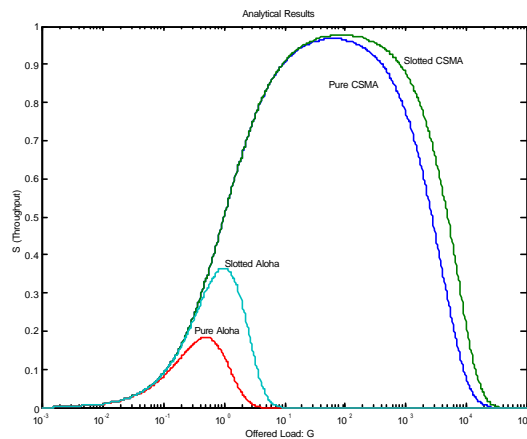
## CSMA Protocol



## Slotted CSMA Protocol



## Confronto tra throughput



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## CSMA/CD (Collision Detection)

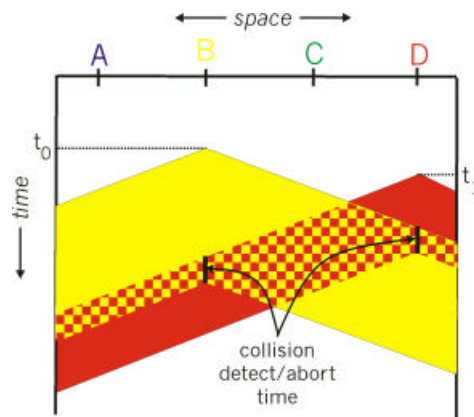
**CSMA/CD:** come CSMA ma...

- ✍ Le collisioni sono rilevate quasi immediatamente
- ✍ Collisioni interrotte appena rilevate
- ✍ Ritrasmissione persistente o non persistente
- ✍ collision detection (CD):
  - ✍ Fattibile in reti wired: misuro la forza relativa dei segnali
  - ✍ Difficile in reti wireless!!

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)



## CSMA/CD collision detection



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## MAC protocols "a turni"

### Protocolli MAC a partizione di canale:

- ⌘ Efficiente condivisione per alti carichi
- ⌘ Inefficiente per bassi carichi: ritardo di accesso,  $1/N$  banda allocata anche se c'è solo un nodo attivo

### Protocolli MAC Random access

- ⌘ Efficiente per bassi carichi: un solo nodo usa tutto il canale
- ⌘ Alti carichi: troppe collisioni

### protocolli MAC a turni

Compromesso tra i due (ma non sempre pratico)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

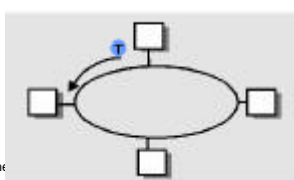
## MAC protocols "a turni"

### Polling:

- ✍ Un nodo master invita gli altri nodi a trasmettere
- ✍ RTS/CTS
- ✍ svantaggi
  - ✍ polling overhead
  - ✍ latenza
  - ✍ Singolo punto di fallimento (master)

### Token passing:

- ✍ Un pacchetto TOKEN viene passato a rotazione
- ✍ Chi ha il token trasmette!
- ✍ svantaggi
  - ✍ token overhead
  - ✍ Latenza
  - ✍ Singolo punto di fallimento (token)



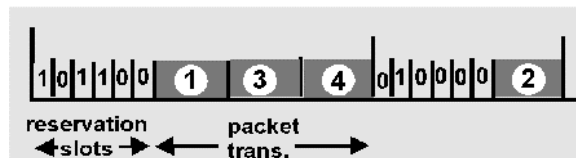
Copyright © Luciano Bononi 2004 (some

alcolatori)

## Protocolli Reservation-based

### Polling Distribuito:

- ✍ Tempo suddiviso in slots
- ✍ All'inizio ci sono N piccoli reservation slots
  - ✍ reservation slot = ritardo propagazione end-to-end
  - ✍ Chi vuole trasmettere occupa un reservation slot che viene visto da tutte le stazioni.
  - ✍ Dopo gli slot di reservation si trasmette in base alla priorità di accesso



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)