

Corso di Laurea in Ingegneria Informatica



**Corso di Reti di Calcolatori
(a.a. 2011/12)**

Roberto Canonico (roberto.canonico@unina.it)

Giorgio Ventre (giorgio.ventre@unina.it)

Lo strato di collegamento:
introduzione e concetti generali
Protocolli ad accesso multiplo

28 novembre 2011

**I lucidi presentati al corso sono uno strumento didattico
che NON sostituisce i testi indicati nel programma del corso**

Nota di copyright per le slide COMICS



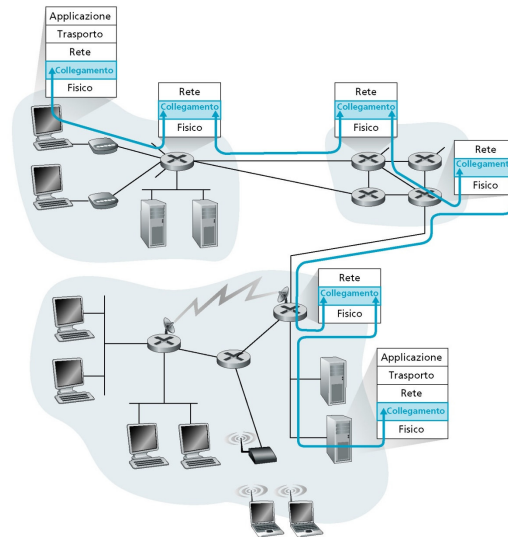
Nota di Copyright

Questo insieme di trasparenze è stato ideato e realizzato dai ricercatori del Gruppo di Ricerca COMICS del Dipartimento di Informatica e Sistemistica dell'Università di Napoli Federico II. Esse possono essere impiegate liberamente per fini didattici esclusivamente senza fini di lucro, a meno di un esplicito consenso scritto degli Autori. Nell'uso dovranno essere esplicitamente riportati la fonte e gli Autori. Gli Autori non sono responsabili per eventuali imprecisioni contenute in tali trasparenze né per eventuali problemi, danni o malfunzionamenti derivanti dal loro uso o applicazione.

Autori:

Simon Pietro Romano, Antonio Pescapè, Stefano Avallone,
Marcello Esposito, Roberto Canonico, Giorgio Ventre

Il livello data link nella rete

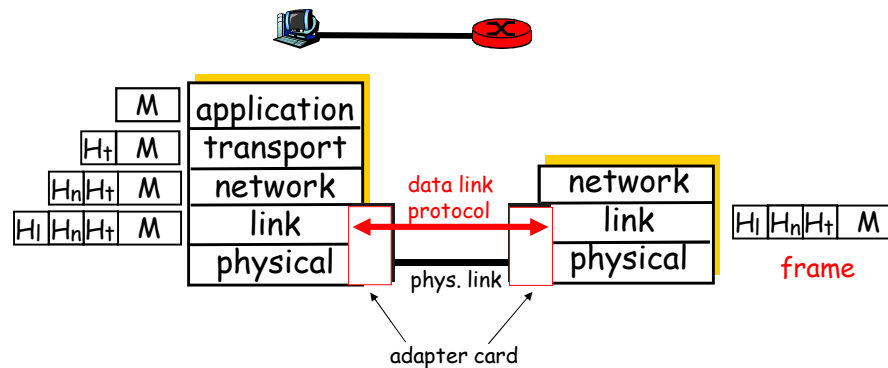


3

Caratteristiche del livello data link



- Si occupa della comunicazione tra due dispositivi *fisicamente connessi*:
 - host-router, router-router, host-host
- Unità di dati: *frame*



4

Servizi del livello data link (1/2)



- **Framing (incorniciatura) ed accesso al link:**
 - Incapsulamento di datagrammi all'interno di frame, aggiunta di campi di intestazione (header e trailer)
 - Gestione dell'accesso al canale, in caso di mezzo condiviso
 - Utilizzo di 'indirizzi fisici' all'interno delle frame, per identificare nodo sorgente e destinazione:
 - NB: indirizzi fisici diversi dagli indirizzi di rete!
- **Trasferimento affidabile dei dati tra due dispositivi fisicamente connessi:**
 - Utile soprattutto in caso di collegamenti con alta probabilità di errore, quali i link wireless
- **Controllo di flusso:**
 - Per regolare la velocità di trasmissione tra mittente e destinatario

5

Servizi del livello data link (2/2)



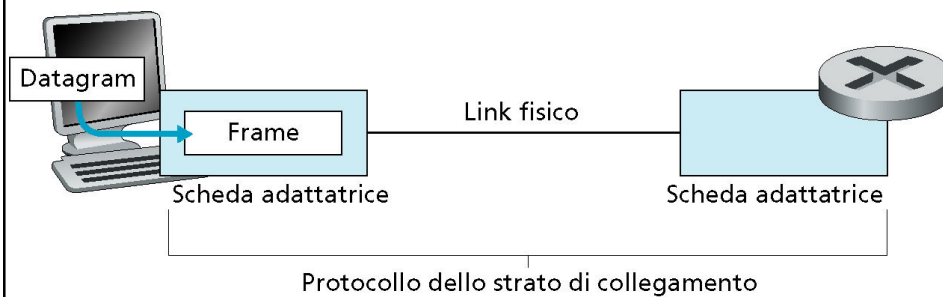
- **Rilevazione degli errori:**
 - Errori causati da attenuazione del segnale o da presenza di rumore (interferenza)
 - Il ricevente rileva la presenza di errori e:
 - Segnala tale evento al mittente...
 - ...oppure elimina la frame ricevuta
- **Correzione degli errori:**
 - Il ricevente identifica e *corregge* errori su alcuni bit della frame, evitando ritrasmissioni da parte del mittente
- **Trasferimento dati di tipo half-duplex o full-duplex**

6

Interfacce di rete



- Un adattatore è un circuito (es: scheda PCMCIA) che si occupa di:
 - Ricevere *datagram* dallo strato di rete
 - Incapsulare i datagram ricevuti all'interno di *frame*
 - Trasmettere le frame all'interno del link di comunicazione
 - In ricezione, effettuare le operazioni inverse...

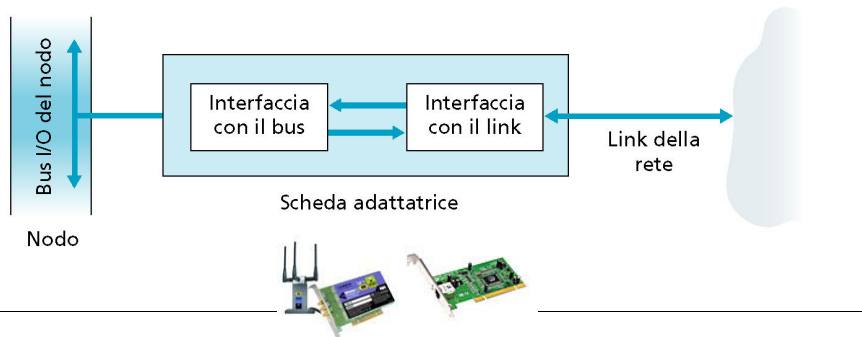


7

Funzionamento degli adattatori di rete

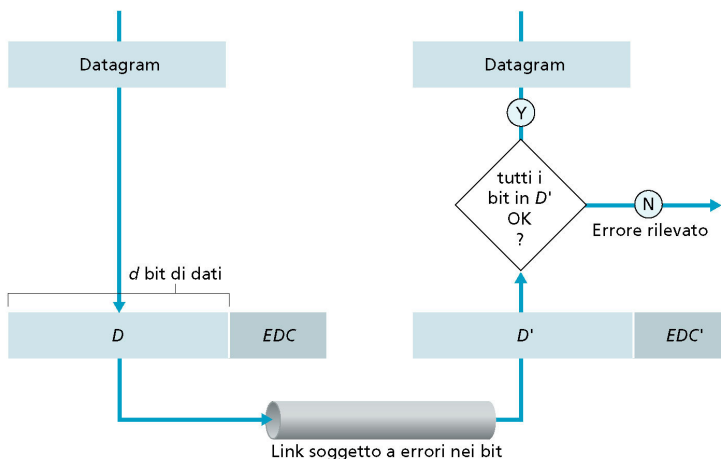


- Gli adattatori sono entità semi-autonome:
 - Alla ricezione di una frame, il nodo è coinvolto solo se i dati devono essere passati al livello rete
 - Un nodo che spedisce un datagram, delega completamente all'adattatore di rete la trasmissione sul link
- Un adattatore è, dunque, dotato di memoria RAM, di un DSP, di chip di elaborazione delle frame e di interfacce verso il bus dell'host e verso il link



8

Rilevazione degli errori



EDC: Error Detection & Correction
D: Dati

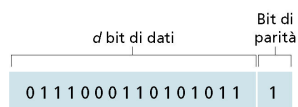
9

Controllo di parità



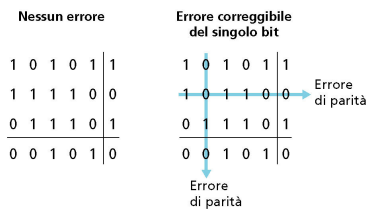
Parità ad un bit:

Rilevazione errori su un singolo bit



Parità a due bit:

Rilevazione e correzione di errori su un singolo bit
Rilevazione di errori su due bit



10

Metodi di checksum



Obiettivo: rilevare “errori” sui bit all’interno dei segmenti trasmessi (tecnica tipicamente utilizzata al livello trasporto)

Mittente:

- Tratta il contenuto del segmento come una sequenza di interi espressi su 16 bit
- checksum: complemento ad 1 della somma in complementi ad 1 del contenuto del segmento
- La checksum calcolata viene inserita all’interno di un apposito campo dell’header del segmento

Ricevitore:

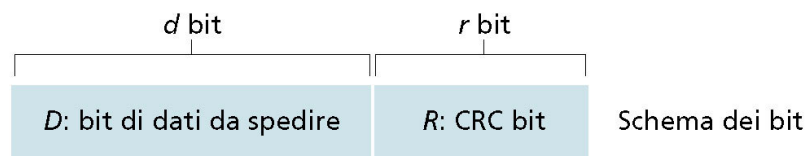
- Calcola la somma in complemento ad 1 dei dati ricevuti (compresa la checksum)
- Risultato composto da tutti 1?
 - NO: errore!
 - SI: nessun errore rilevato...
 - ...il che non vuol dire che non vi siano stati errori ☹

11

Codici CRC -- Cyclic Redundancy Check



- Si considerano i bit di dati, **D**, come un numero binario
- Si sceglie un opportuno pattern di $r+1$ bit (generatore), **G**
- Obiettivo: scegliere r bit di controllo CRC, **R**, tali che:
 - $\langle D, R \rangle$ sia divisibile esattamente per **G** (modulo 2)
 - Il ricevente, che deve conoscere **G**, divide $\langle D, R \rangle$ per **G**:
 - Se il resto della divisione non è nullo:
 - errore rilevato!
- Con tale tecnica si possono rilevare tutti gli errori che coinvolgono meno di $r+1$ bit
- Il CRC è molto utilizzato nella pratica, al livello data link



$D \cdot 2^r \text{ XOR } R$ Formula matematica

12

Un esempio di codice CRC



Obiettivo, trovare R in modo tale che esista un n:

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

In maniera equivalente:

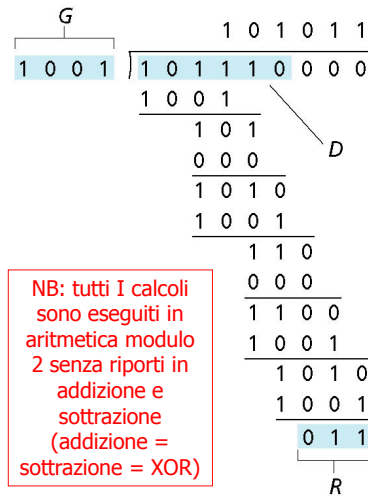
$$D \cdot 2^r = nG \text{ XOR } R$$

ossia:

se dividiamo $D \cdot 2^r$ per G, vogliamo un resto pari a R:

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

Un esempio con r=3



NB: tutti i calcoli sono eseguiti in aritmetica modulo 2 senza riporti in addizione e sottrazione (addizione = sottrazione = XOR)

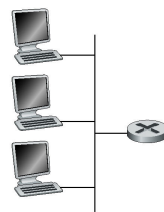
Tipi di link di rete



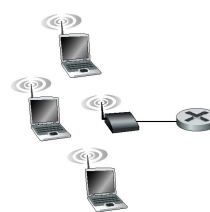
Due tipi di "link":

- punto-punto:
 - Point to Point Protocol – PPP
 - Serial Line IP – SLIP
- Broadcast:
 - mezzo condiviso
 - Ethernet
 - Wavelan
 - Satellite
 - Ecc.

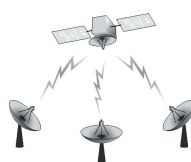
Condivisione con cablaggio (per esempio, Ethernet)



Condivisione senza fili (per esempio, Wifi)



Satellite



Cocktail party



Protocolli di accesso multiplo



- Un unico canale di comunicazione condiviso
- Due o più trasmissioni simultanee da parte dei nodi della rete:
 - interferenza
 - Solo un nodo può inviare dati con successo!
- **Protocolli di accesso multiplo:**
 - Un algoritmo distribuito determina le modalità di condivisione del canale, vale a dire *quando* una stazione può trasmettere
 - Le comunicazioni per regolare l'accesso al canale utilizzano il canale stesso!
 - Caratteristiche di un protocollo di accesso multiplo:
 - Sincrono o asincrono
 - Necessità di informazioni riguardanti le altre stazioni
 - Robustezza (ad esempio, in relazione agli errori sul canale)
 - Prestazioni

15

Protocolli di accesso multiplo: tassonomia



- **Channel Partitioning**
 - Suddivide il canale in "porzioni" più piccole (slot temporali o di frequenza)
 - Ogni nodo gode dell'uso esclusivo di una di tali porzioni
- **Random Access**
 - Consente collisioni dovute a trasmissioni simultanee
 - Gestisce le collisioni
- **Taking turns**
 - Coordina opportunamente l'accesso al mezzo, in modo da evitare le collisioni

16

Protocolli di accesso multiplo: obiettivi



- Per un canale con velocità di R bit/sec:
 - Se un solo nodo ha dati da inviare:
 - Quel nodo ha un throughput di R bit/sec
 - Se M nodi hanno dati da spedire:
 - Ognuno di essi ha un throughput medio di R/M bit/sec
 - Il protocollo per la gestione dell'accesso è distribuito:
 - Assenza di "single points of failure"
 - Il protocollo è semplice:
 - Implementazione economica

17

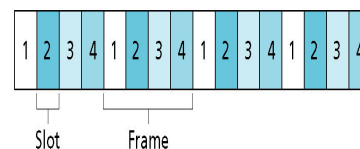
Protocolli di suddivisione del canale: TDMA



Time Division Multiple Access

- L'accesso al canale avviene a "cicli":
 - Ogni stazione ottiene uno slot di trasmissione di lunghezza fissa in ogni ciclo
 - Gli slot inutilizzati da una stazione vanno deserti
- Vantaggi:
 - Elimina le collisioni
 - È equo
- Svantaggi:
 - Throughput max per un nodo, in una rete con N stazioni:
 - R/N bit/sec anche se il nodo in esame è l'unico ad avere frame da spedire
 - Un nodo deve sempre aspettare il suo turno nella sequenza di trasmissione

TDM



Legenda:

2 Tutti gli slot etichettati "2" sono dedicati a una specifica coppia sender-receiver

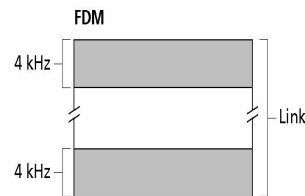
18

Protocolli di suddivisione del canale: FDMA



Frequency Division Multiple Access

- Lo spettro di trasmissione è diviso in bande di frequenza
- Ad ogni stazione è assegnata una banda di frequenza fissa
- Il tempo di trasmissione inutilizzato nelle singole bande di frequenza risulta sprecato
- Vantaggi:
 - Come per il TDMA
- Svantaggi:
 - Come per il TDMA



19

Protocolli di suddivisione del canale: CDMA

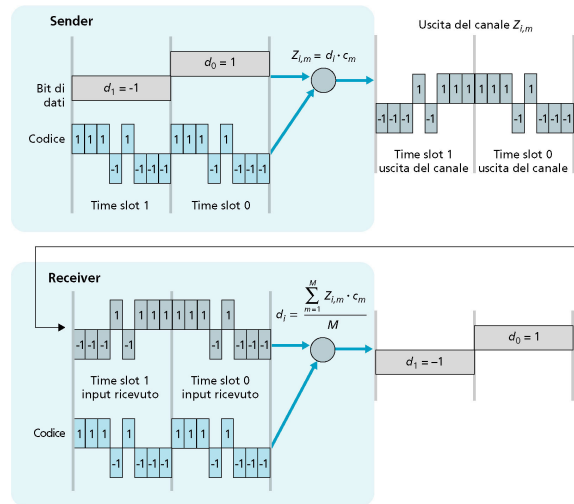


CDMA (Code Division Multiple Access)

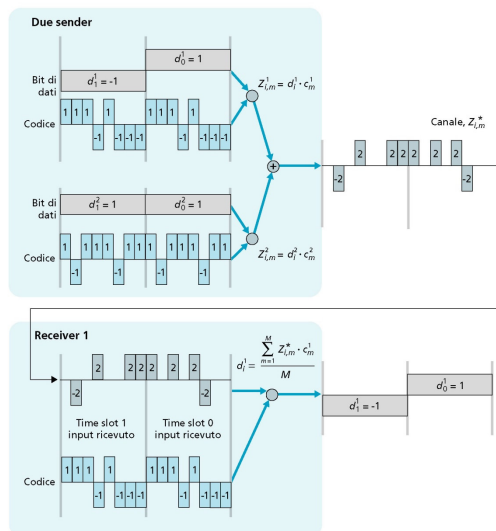
- Un codice unico è assegnato ad ogni utente:
 - *code set partitioning*
- Usato principalmente nei canali wireless di tipo broadcast (reti cellulari, satellitari, ecc.)
- Tutti gli utenti condividono la stessa frequenza di trasmissione, ma ognuno di essi possiede un codice unico (noto come "*chipping sequence*") per codificare i dati
- **segnale codificato** = (dati originali) X (chipping sequence)
- **decodifica**: prodotto scalare del segnale codificato e della chipping sequence
- Consente a diversi nodi di trasmettere simultaneamente, riducendo al minimo l'interferenza nel caso in cui si siano scelti codici "ortogonali"

20

CDMA: codifica e decodifica



CDMA: interferenza tra due mittenti*



* CDMA lavora nell'ipotesi che i segnali dei bit trasmessi con interferenza siano cumulativi. 22

Protocolli ad accesso casuale



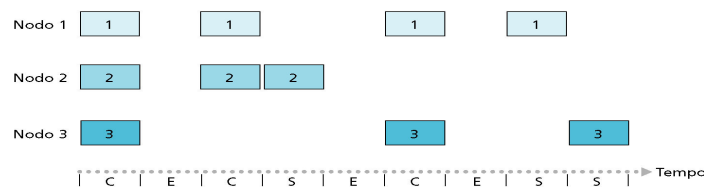
- Quando un nodo ha un pacchetto da trasmettere:
 - Trasmette alla massima velocità consentita dal canale
 - Non esiste nessuna forma di coordinamento *a priori* tra i nodi
- Trasmissione simultanea di due o più nodi:
 - Collisione!
- Un **protocollo ad accesso casuale** specifica:
 - Come rilevare le collisioni
 - Come risolvere le collisioni:
 - Es: tecniche di ritrasmissione delle frame
- Esempi di protocolli *random access*:
 - slotted ALOHA
 - ALOHA
 - CSMA e CSMA/CD

23

Slotted Aloha



- Tutti i pacchetti sono lunghi L bit e il tempo è diviso in slot di uguale durata (L/R s)
- Se un nodo ha dati disponibili per la trasmissione:
 - Trasmette all'inizio del primo slot disponibile*
- In caso di collisione:
 - Ritrasmette il pacchetto negli slot successivi con probabilità p , finché la trasmissione non va a buon fine



Legenda:
C = Slot con collisione; E = Slot vuoto; S = Slot con successo

* Tutti i nodi sono sincronizzati: sanno quando iniziano gli slot

24

Slotted ALOHA: efficienza



- Qual è la percentuale massima di slot in cui la trasmissione ha successo?
- Supponiamo che N stazioni abbiano frame da trasmettere:
 - Ogni stazione trasmette in un determinato slot, con probabilità p
 - La probabilità S che una trasmissione abbia successo è data da:
 - Per il singolo nodo:
 - $S = p(1-p)^{N-1}$
 - Dato che ci sono N nodi:
 - $S = \text{Prob}(\text{solo uno degli N nodi trasmetta}) = N p (1-p)^{N-1}$
 - ...il **valore ottimo** di S , per N che tende ad infinito, è:
 - » $1/e$, pari a circa il 37%

25

Slotted ALOHA: efficienza massima



$$E(p) = Np(1-p)^{N-1}$$
$$E'(p) = N(1-p)^{N-1} - Np(N-1)(1-p)^{N-2}$$
$$= N(1-p)^{N-2}((1-p) - p(N-1))$$

$$E'(p) = 0 \Rightarrow p^* = \frac{1}{N}$$

$$E(p^*) = N \frac{1}{N} \left(1 - \frac{1}{N}\right)^{N-1} = \left(1 - \frac{1}{N}\right)^{N-1} = \frac{\left(1 - \frac{1}{N}\right)^N}{1 - \frac{1}{N}}$$

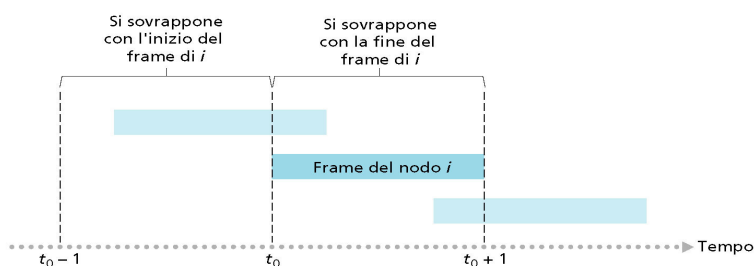
$$\lim_{N \rightarrow \infty} \left(1 - \frac{1}{N}\right) = 1 \quad \lim_{N \rightarrow \infty} \left(1 - \frac{1}{N}\right)^N = \frac{1}{e}$$

26

ALOHA puro



- ALOHA “unslotted”:
 - Più semplice
 - Non richiede sincronizzazione:
- In trasmissione:
 - Invia la frame non appena i dati sono disponibili
- La probabilità di collisione raddoppia:
 - Una frame inviata al tempo t_0 può collidere con altre frame inviate in $[t_0-1, t_0+1]$

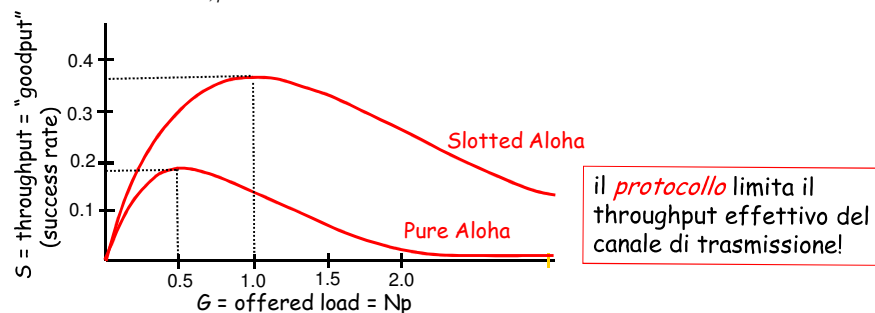


27

ALOHA puro: prestazioni



- $P(\text{successo di un dato nodo}) =$
 $P(\text{nodo trasmetta}) \cdot P(\text{nessun altro nodo trasmetta in } [t_0-1, t_0]) \cdot$
 $P(\text{nessun altro nodo trasmetta in } [t_0, t_0+1]) =$
 $p \cdot (1-p)^{(N-1)} \cdot (1-p)^{(N-1)} = p \cdot (1-p)^{2(N-1)}$
- $P(\text{successo di uno su } N \text{ nodi}) = N \cdot p \cdot (1-p)^{2(N-1)}$
 - ...il valore ottimo di p , per N che tende ad infinito, è:
 - » $1/2e$, pari a circa il 18%



28

CSMA: Carrier Sense Multiple Access



CSMA:

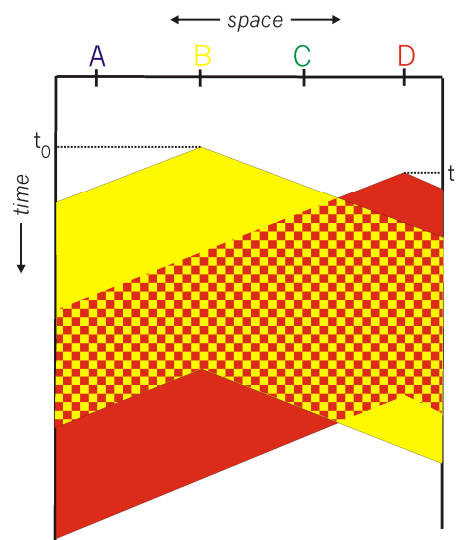
- “Ascolta prima di parlare” (*listen before talking*):
 - Se il canale è libero
 - Trasmetti la frame
 - Se il canale è occupato
 - Rimanda la trasmissione
- CSMA persistente:
 - Riprova immediatamente con probabilità p quando il canale si libera
- CSMA non persistente:
 - Riprova dopo un intervallo casuale

29

CSMA: collisioni



- Col CSMA si possono avere collisioni:
 - Il ritardo di propagazione fa sì che due nodi possano non ascoltare le reciproche trasmissioni
- In caso di collisione:
 - Il tempo di trasmissione della frame risulta completamente sprecato
- La *distanza* ed il *ritardo di propagazione* concorrono a determinare la probabilità di collisione



30

CSMA/CD: CSMA con Collision Detection

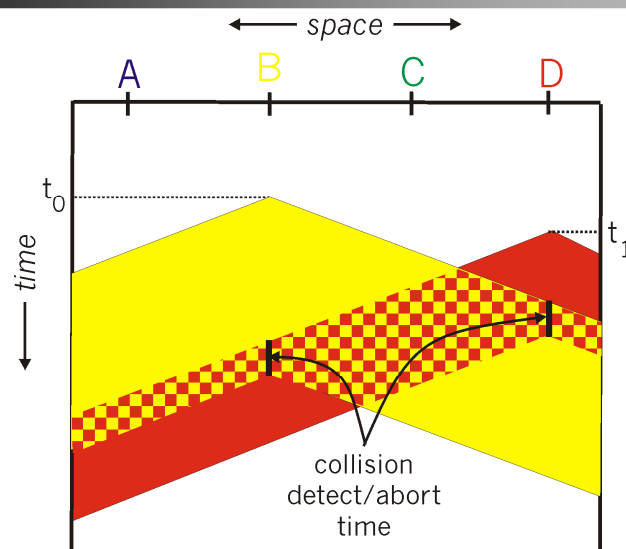


CSMA/CD:

- “Ascolta mentre parli” (*listen while talking*):
 - Analogo al CSMA, ma in più:
 - Rileva le collisioni durante la trasmissione
 - Le trasmissioni che collidono vengono terminate, riducendo lo spreco di risorse del canale trasmissivo
 - Ritrasmissioni persistenti o non persistenti
- Collision Detection:
 - Si misura la potenza del segnale ricevuto e la si compara con quella del segnale trasmesso

31

CSMA/CD: funzionamento



32

Protocolli di tipo “Taking Turns”



Protocolli “channel partitioning”:

- Condivisione efficiente del canale con carico elevato
- Poco efficienti con carico leggero:
 - Ritardo nell’accesso al canale
 - Banda pari ad $1/N$ anche se solo uno degli N nodi presenti è attivo!

Protocolli “random access”:

- Efficienti con carico leggero:
 - Un singolo nodo può utilizzare a pieno il canale
- Carico elevato:
 - Overhead dovuto alla presenza di collisioni

Protocolli “taking turns”:

- Cercano di prendere il meglio dai due approcci precedenti!

33

Tipici protocolli “Taking Turns”

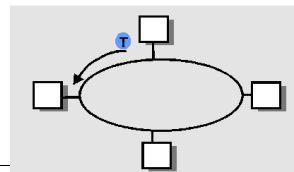


Polling:

- Un nodo *master* “invita” i nodi *slave* a trasmettere in maniera alternata
- Impiego di messaggi del tipo “Request to Send”, “Clear to Send”
- Problemi da affrontare:
 - Overhead dovuto al polling
 - Latenza
 - Presenza di un single point of failure (master)

Token passing:

- Un “gettone” (*token*) di controllo **viene** passato da un nodo al successivo in maniera sequenziale
- Il possesso del *token* dà diritto a trasmettere
- Problemi da affrontare:
 - Overhead nella gestione del token
 - Latenza
 - Presenza di un single point of failure (token)



34