Reti di Telecomunicazioni

Livello Data Link

Sottolivello MAC

Autori

Queste slides sono state scritte da

Michele Michelotto:

michele.michelotto@pd.infn.it

che ne detiene i diritti a tutti gli effetti

Copyright Notice

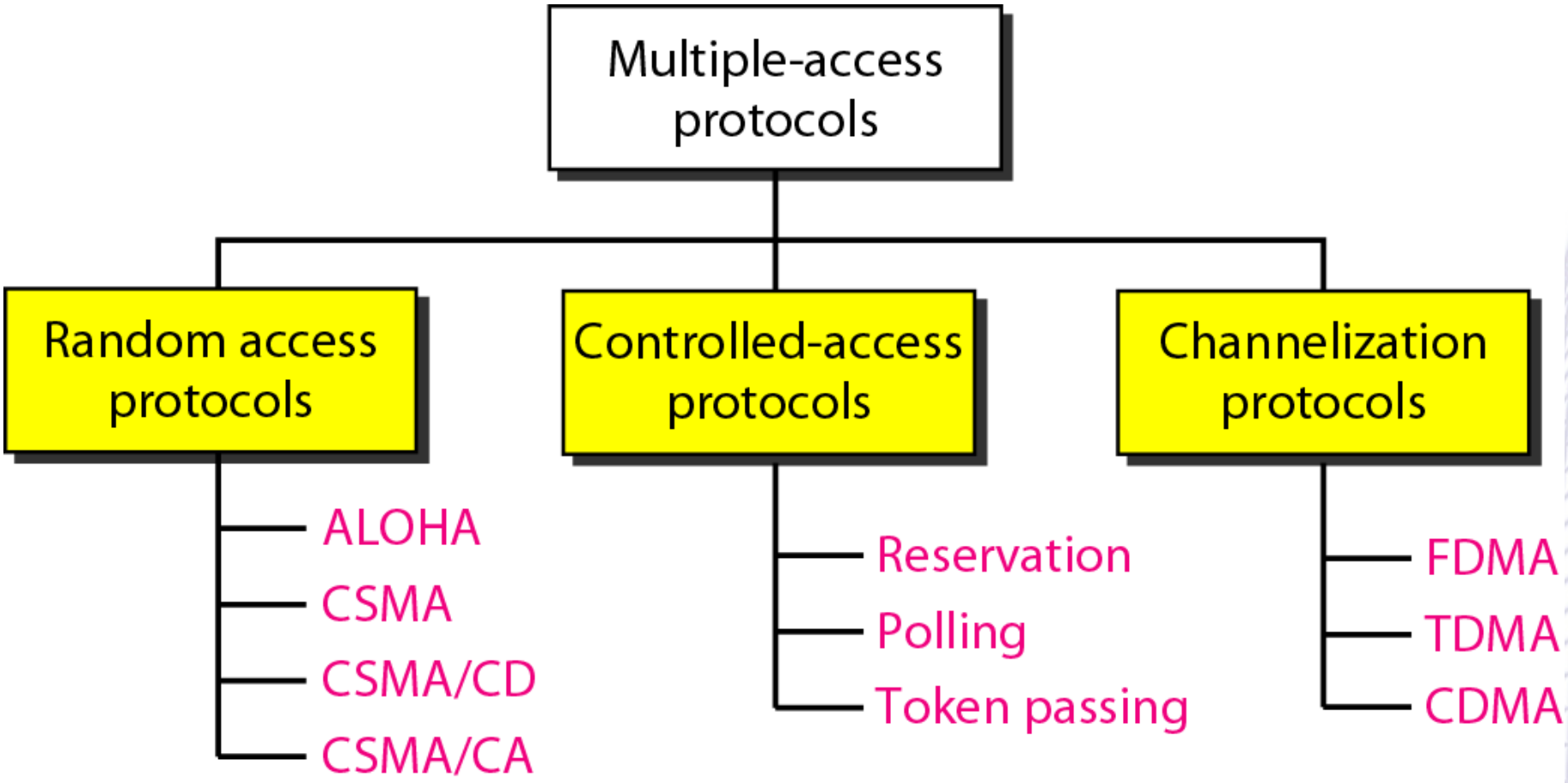
MAC sublayer

Introduzione
Layer: Modello OSI e TCP/IP
Physics Layer
Data Link Layer
MAC sublayer

Medium Access SubLayer

- Ci sono reti punto-punto e reti con canali broadcast
- Estensione del Data-Link necessario in una rete broadcast
  - Es: In una conferenza telefonica ho sei persone con sei telefoni, tutti connessi in modo che quando uno parla gli altri sentono. Come evitare che due inizino a parlare allo stesso istante? In un meeting dal vivo si evita il caos per esempio alzando una mano. Uso un moderatore? Automoderazione.
- In una rete si usa un sottolayer del Data Link chiamato Medium Access Control (**MAC**)
- Importante soprattutto in ambito LAN

Multiple Access



Come allocare il canale?

- Staticamente
  - Traffico telefonico con N utenti
  - TDM: Poco efficente. Ognuno parla ogni **m** secondi
  - Se uno non vuole parlare la sua slot temporale non viene utilizzata
  - In caso di FDM si divide la bandwidth totale in bande di frequenza, non c'è interferenza tra utenti e funziona bene quando c'è un numero piccolo e costante di utenti ognuno dei quali ha un traffico pesante bufferizzato. Lo spreco comunque rimane

Spreco con allocazione statica

- Con traffico regolare come il “voice”
  - Se ho meno di N utenti spreco banda
  - Se ho più di N utenti alcuni non possono parlare anche se altri stanno sotto-utilizzando la loro slot
  - Il problema è che quando uno non utilizza la sua slot la banda viene persa
- Con traffico “data” che si presenta in burst (peak/mean vicino a 1000) lo spreco è ancora maggiore

Calcolo delay

- Calcolo il delay di trasmissione per un canale con bandwidth C bps, con rate di immissione dei pacchetti λ frame/sec e ogni frame ha una lunghezza con distribuzione esponenziale con media $1/\mu$ bits/frame
- La teoria delle code ci dice che il tempo per gestire i pacchetti

$$T = \frac{1}{\mu C - \lambda}$$

  - Per esempio se C=100 Mbps, il frame medio **$1/\mu$** è di 10000 bit e λ è di 5000 frame/sec ottengo T=200μs
  - Se ignorassimo i problemi di accodamento e chiedessimo quanto ci mettono 10000 bit a passare una rete a 100 Mbps otterrei 100μs
  - Questo vale se non c'è contesa sul canale

Con allocazione statica

- Se ogni canale viene diviso in N canali indipendenti ognuno di essi ha solo C/N bandwidth e quindi il rate di ingresso di ogni sottocanale diventa λ/N

$$T = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda} = NT$$

- Quindi il delay medio usando FDM diventa N volte peggiore che se tutti i frame venissero magicamente gestiti da una coda centrale
- Lo stesso vale per TDM, ogni user devo aspettare il mio slot ogni N, o anche se separo fisicamente la mia rete. Es 10 reti da 10 Mbps

Accesso casuale

- Nel metodo ad accesso casuale (**random access**) detto anche a contesa (**contention**) nessuna stazione ha il controllo del mezzo trasmissivo, ma se lo devono guadagnare a fatica.
- Questo deriva dal fatto che l'accesso avviene in tempi random e inoltre non c'è una regola che specifichi chi è il prossimo a trasmettere
- L'accesso random implica che almeno due stazioni potrebbero trasmettere nello stesso momento (collisioni)

Random access

- ALOHA. Procedura molto semplice per Multiple Access
- CSMA. Come Aloha ma prima controllo se il mezzo è libero
- CSMA/CA e CSMA/CD aggiungo meccanismi per evitare le collisioni

5 Postulati

- Prima di vedere alcuni metodi di allocazione dinamica “random access” dobbiamo tenere conto di cinque assunzioni fondamentali:
  - N stazioni indipendenti (computer, telefoni) e il traffico viene generato in a ritmo costante
  - **Un unico canale condiviso per ricezione e trasmissione** (non c’è altro modo per comunicare)
  - Collisioni: Se due frames sono trasmessi simultaneamente queste si sovrappongono nel tempo, questo evento si chiama **collisione** e tutte le stazioni sono in grado di rivelarlo e quel frame può essere ritrasmesso in un secondo momento

Allocazione dinamica

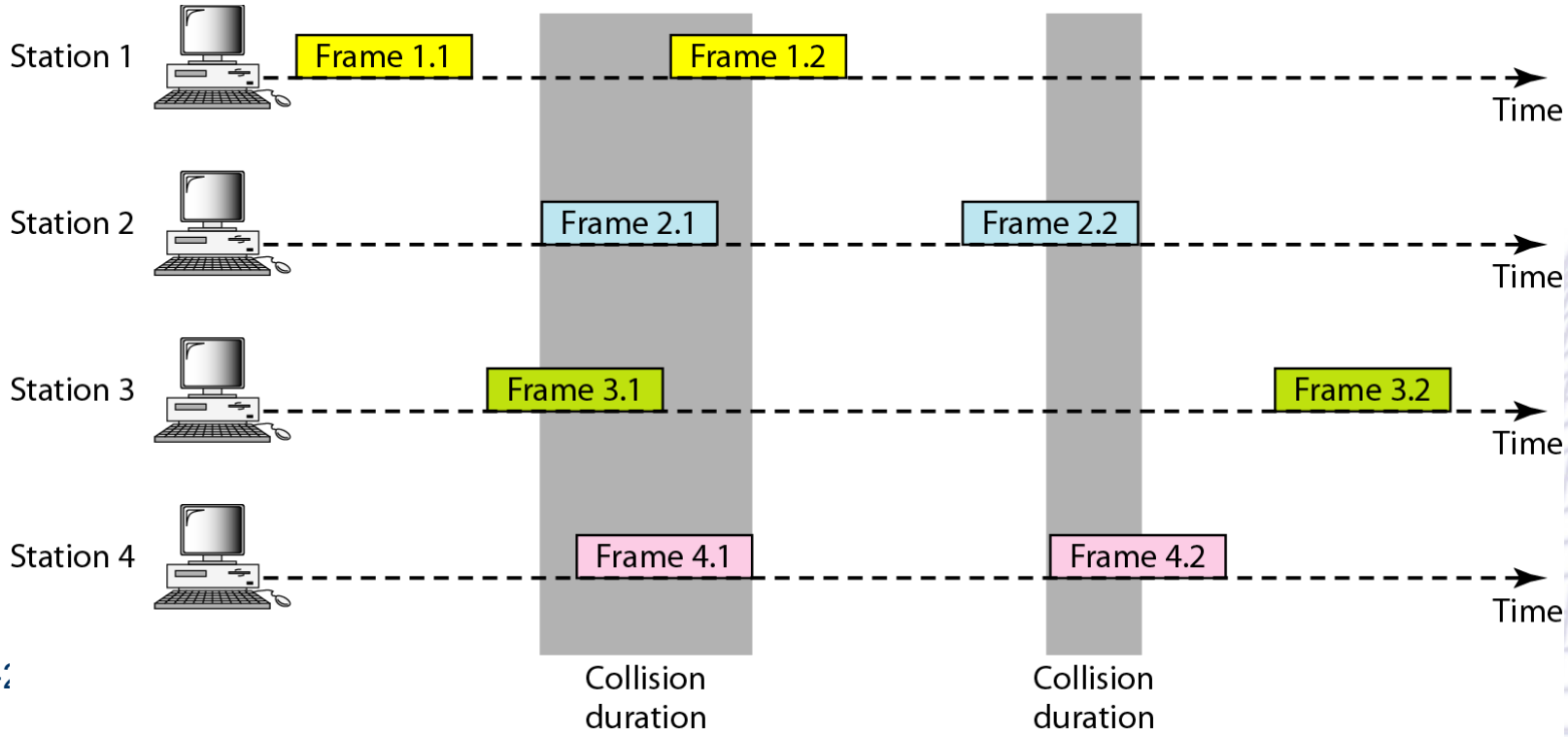
- Tempo
  - **Tempo continuo**: la trasmissione inizia in qualsiasi istante, non c'è un master clock
  - **Slotted Time**: la trasmissione inizia a intervalli discreti, uno slot contiene 0, 1 o più frame (slot idle, trasmissione normale, collisione)
- Carrier Sense
  - **Carrier Sense**: Le stazioni sanno se il canale è in uso prima di usarlo. In tal caso non viene usato finchè non torna idle
  - **No Carrier Sense**: Le stazioni non possono testare il canale. Semplicemente trasmettono. Solo in seguito si accorgono se la trasmissione ha avuto successo

La storia

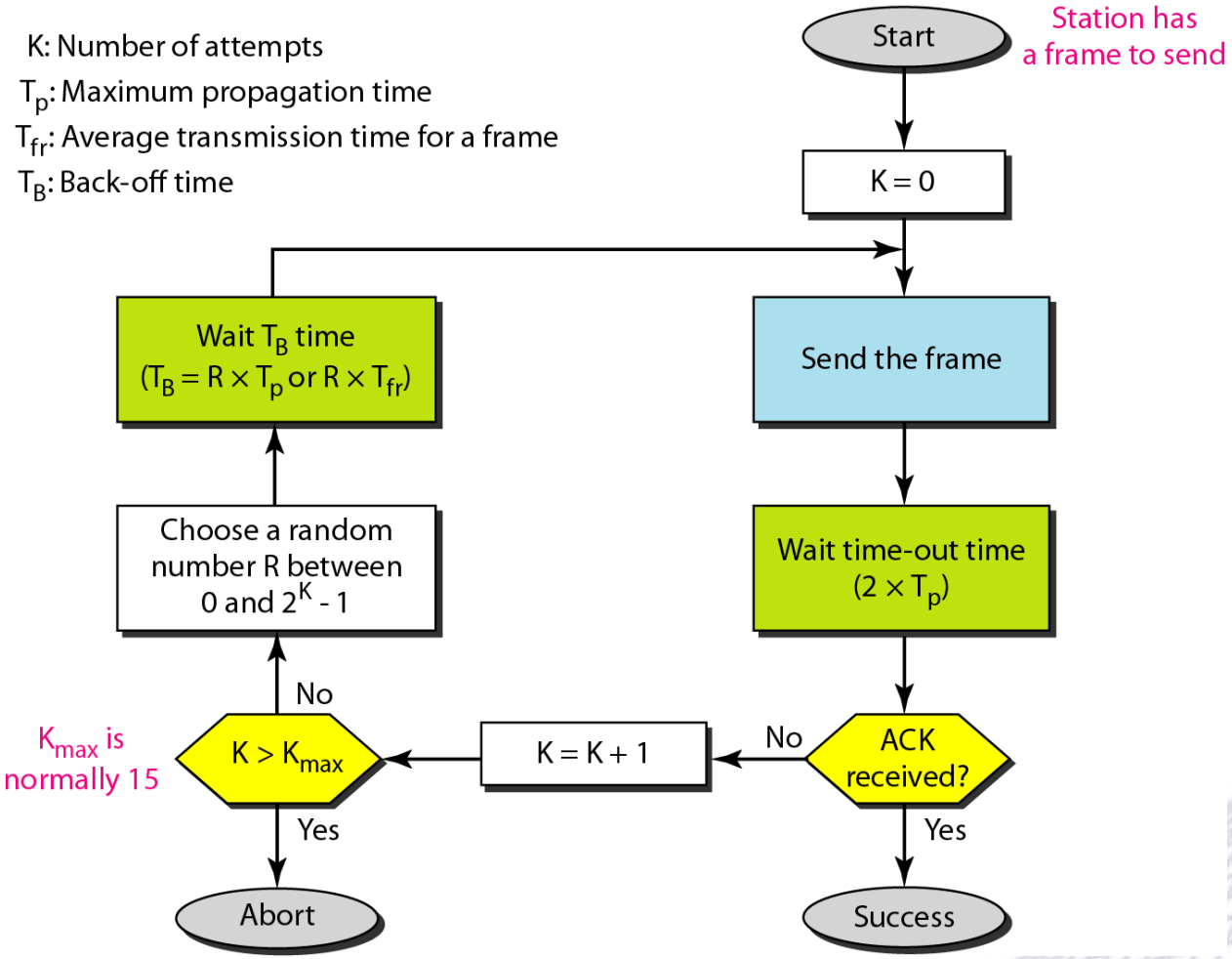
- Anni ‘70 - Aloha:
  - Stazioni radio al suolo. Ognuna trasmette quando le pare. Applicabile quando un certo numero di utenti non coordinati competono per una risorsa.
- Pure Aloha
  - Frame trasmesse a tempi arbitrari, se anche solo l’ultimo bit di un frame si sovrappone al primo di un altro si butta tutto.
  - Si aspetta un tempo **random** (altrimenti c’è collisione di nuovo di sicuro) e si ritrasmette

ALOHA puro

- Frame di dimensioni costanti
- Il checksum non distingue (non deve farlo) tra un scontro frontale e una toccatina



Protocollo ALOHA



Quale efficienza?

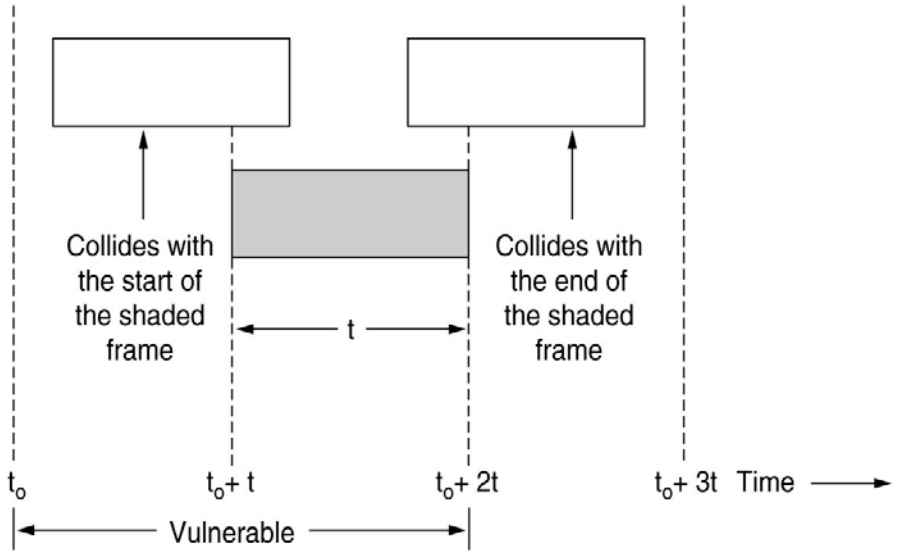
- Quale frazioni di frame sfugge a collisioni, date le circostanze caotiche?
- **Frame-time** è il tempo necessario per trasmette una frame di lunghezza fissa (“frame lenght”/“bit rate”)
- Supponiamo che vengano immessi pacchetti secondo una distribuzione di Poisson con **N** frame per ogni frame time
- Se **N>1** vengono generati più frame di quanti il canale ne possa gestire e praticamente tutti soffriranno una collisione
- Invece ci aspettiamo un throughput ragionevole per **0<N<1**

Quale efficienza?

- Oltre ai nuovi frame le stazioni generano anche frame di ritrasmissione di quelli che hanno subito collisione
- Supponiamo che la probabilità di k tentativi di tramissioni vecchie e nuove abbia distribuzione di Poisson con media G per frame time
- Chiaramente $G \geq N$.
  - Quando il carico è basso, $N \approx 0$ ci saranno poche collisioni, poche ritrasmissioni $G \approx N$
  - Ad alto carico molte collisioni $G > N$

Throughput

- Il throughput S è pari al carico che si presenta G per la probabilità P_0 che il frame non soffra di collisioni: $S=GP_0$
- Il frame (in grigio) non soffre di collisioni se nessun frame viene spedito nel periodo vulnerabile
- Probabilita che k frame vengano generati durante un certo frame time è data dalla distribuzione di Poisson



$$\Pr[k] = \frac{G^k e^{-G}}{k!}$$

Probabilità di collisioni

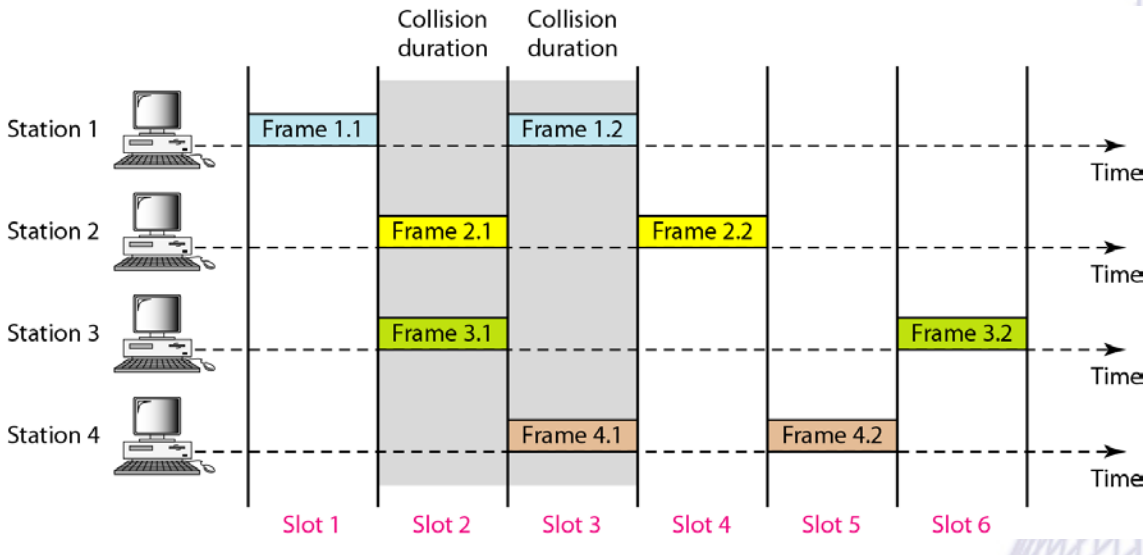
- Probabilità di avere 0 collisioni e^{-G}
- In un intervallo di due frame time vengono generati in media 2G frames e la probabilità di avere 0 collisioni nel periodo vulnerabile è e^{-2G}
- Quindi $S=Ge^{-2G}$
- Il throughput massimo: 0.184 quando G=1/2

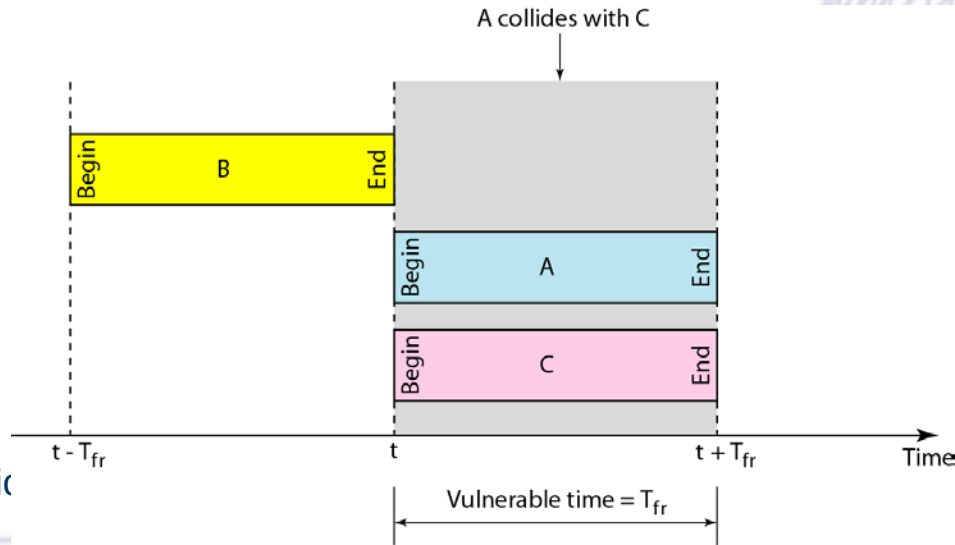
Slotted ALOHA

- Slotted Aloha (Roberts ’72)
  - Si divide il tempo in intervalli discreti, lunghi quanto un frame
  - Gli utenti devono mettersi d’accordo sui confini di uno slot, per esempio una stazione emette un segnale all’inizio di un intervallo come un master clock
  - La stazione deve aspettare e trasmettere solo nei momenti giusti
  - Mentre Pure Aloha ha un’efficienza massima del 18% , la slotted Aloha arriva a 37%

Slotted Aloha

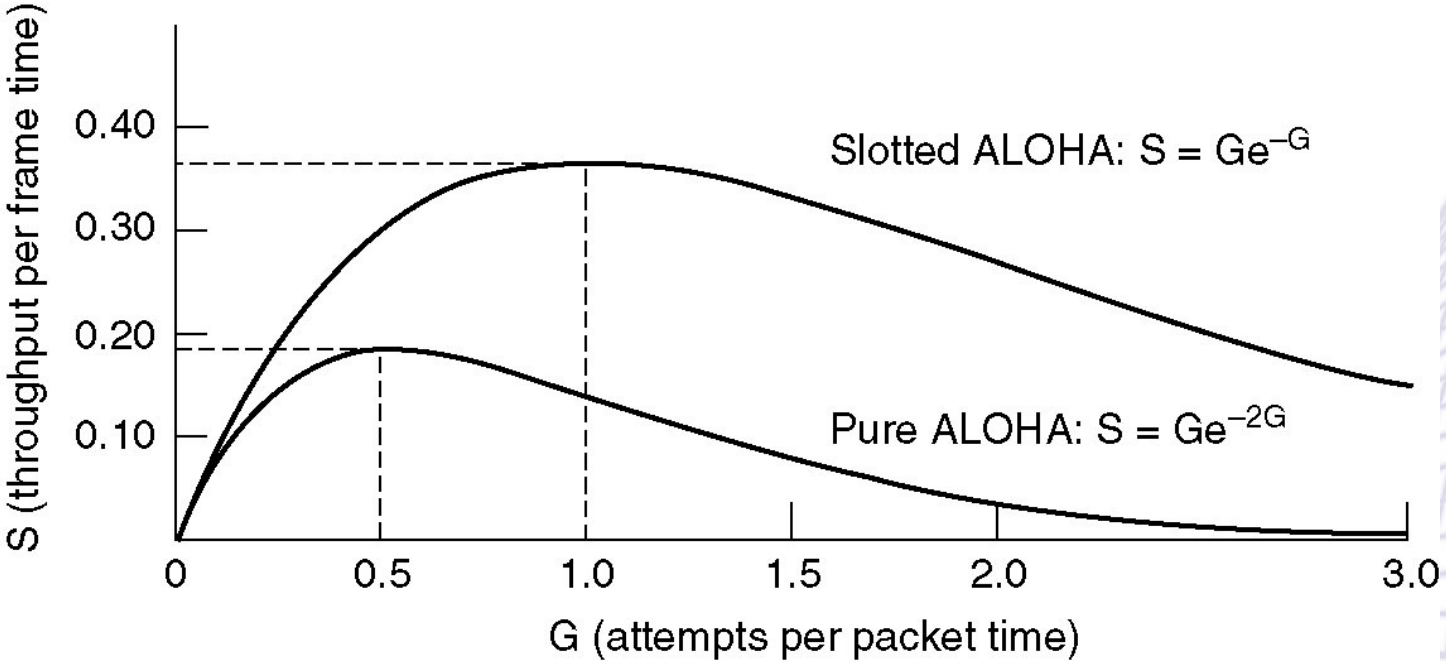
- Come si vede il tempo di vulnerabilità si dimezza
- Quindi $S=Ge^{-G}$
- Il throughput massimo migliora: 0.368 quando G=1





Confronto Throughput

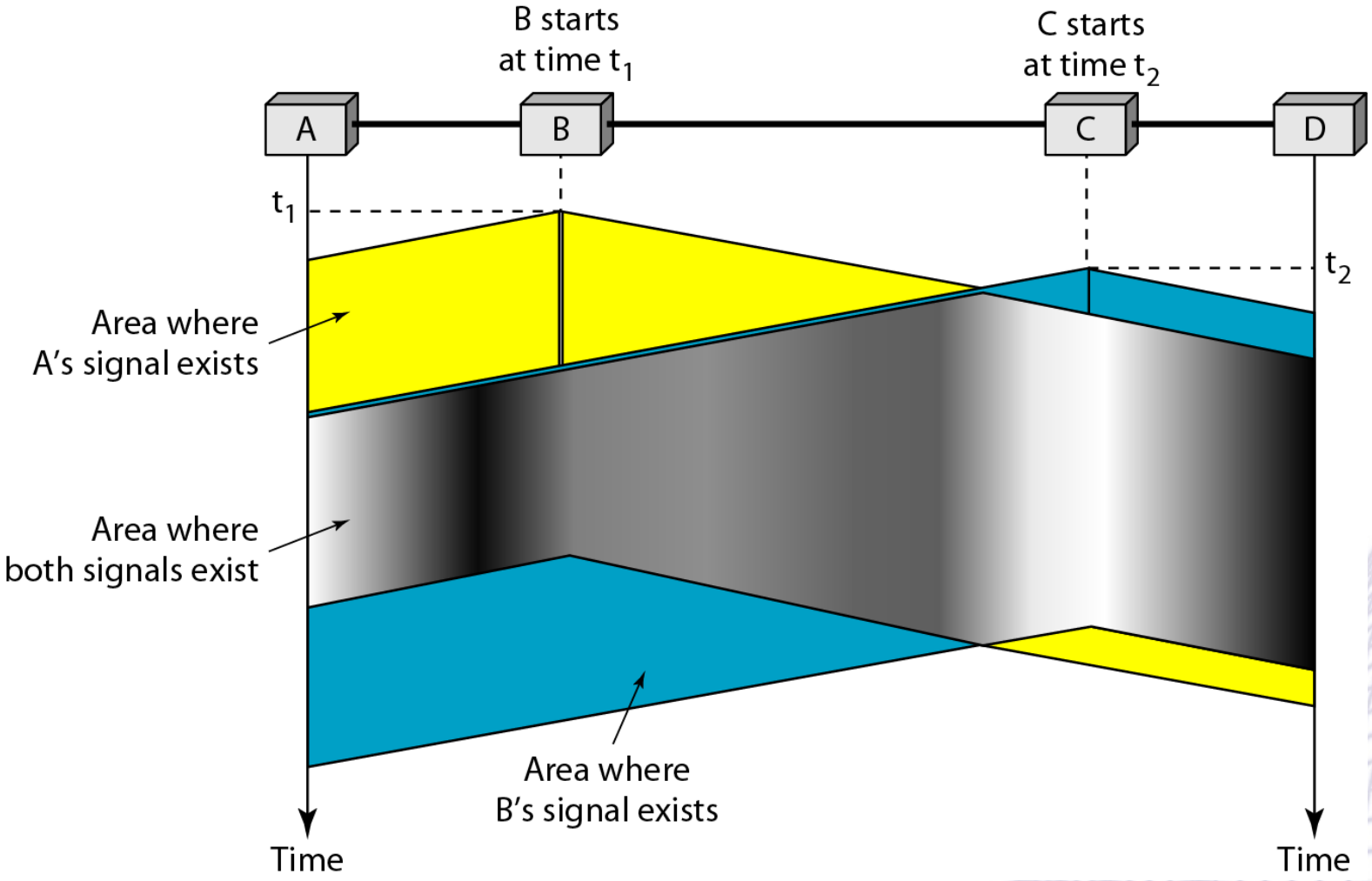
- Pure Aloha: al massimo utilizzo il canale al 18%, non molto! Ma cosa si pretende se tutti parlano quando gli pare?
- Slotted Aloha. Con G=1 abbiamo 37% *empty*, 37% *success*, 26 % *collision*. Con G maggiori si riducono gli *empties* ma aumentano esponenzialmente le collisioni
- G tentativi in un frame time (dove frame time → tempo per trasmettere un frame di lunghezza fissa)



Carrier Sense MA

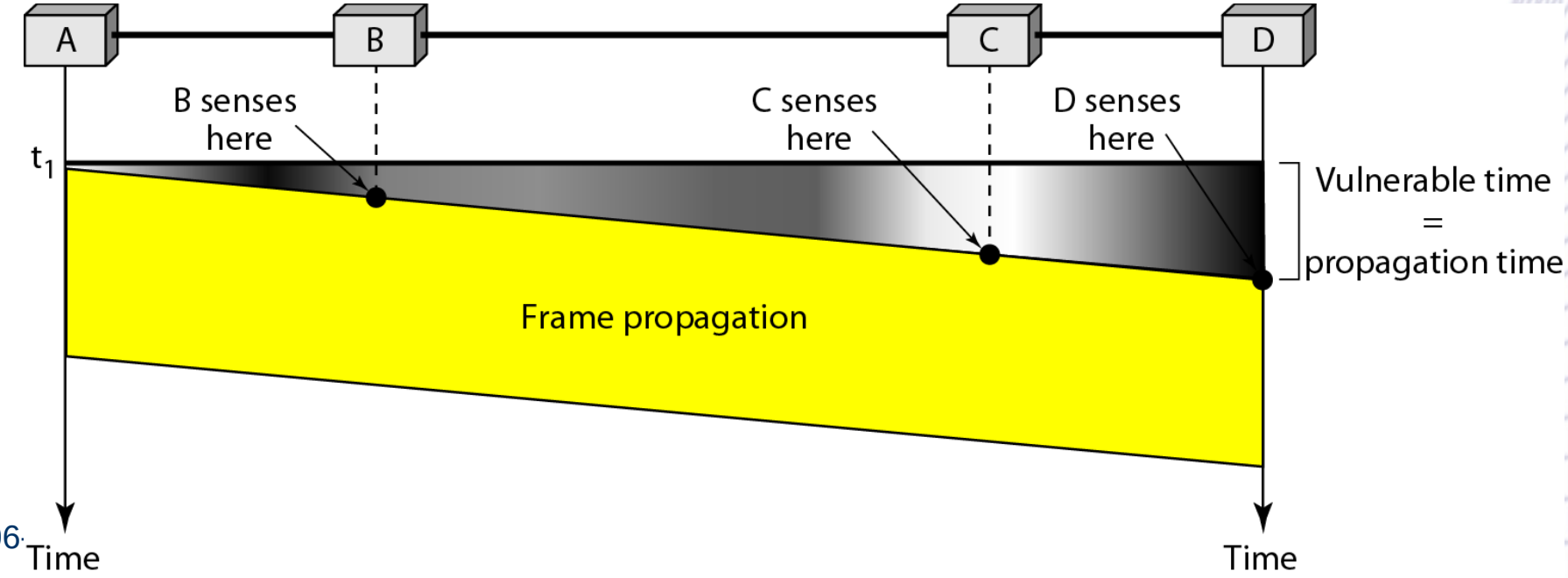
- Efficienza massima di ALOHA è 1/e
- Non è una sorpresa visto che ognuno trasmette senza prestare attenzione a quello che fanno gli altri
- In una LAN è possibile tuttavia per una stazione sentire cosa stanno facendo le altre e agire di conseguenza
- Kleinroch e Tobagi 1975 – hanno studiato diversi protocolli di tipo **Carrier Sense**:
- Collisioni meno frequenti ma ancora possibili

Collisioni protollo CSMA



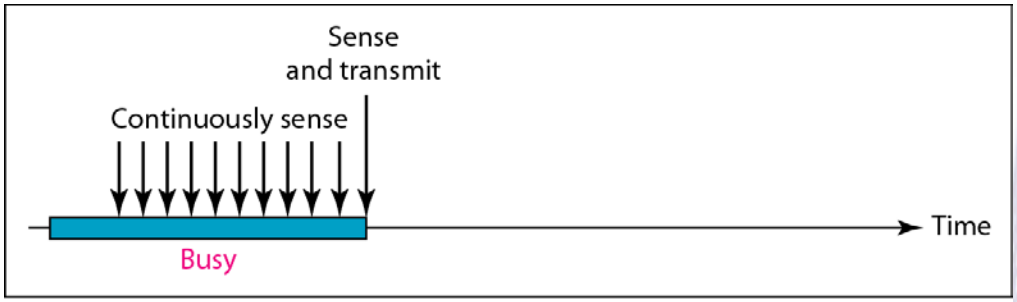
Vulnerabilità CSMA

- Il tempo di vulnerabilità si ricava dal Tempo di propazione da un estremo all'altro del mezzo trasmissivo
- Vediamo il caso peggiore:

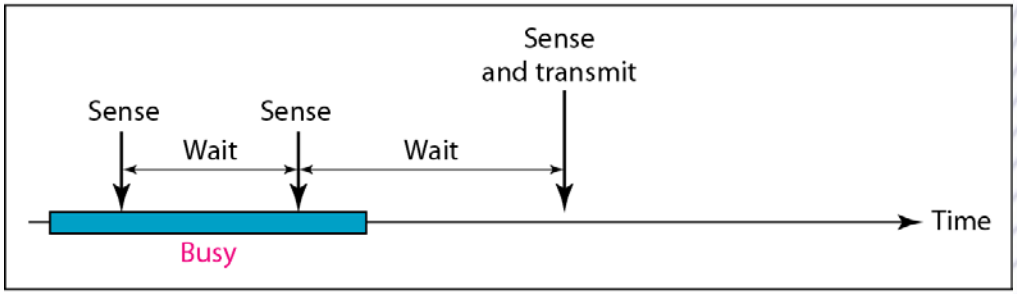


Metodi di insistenza

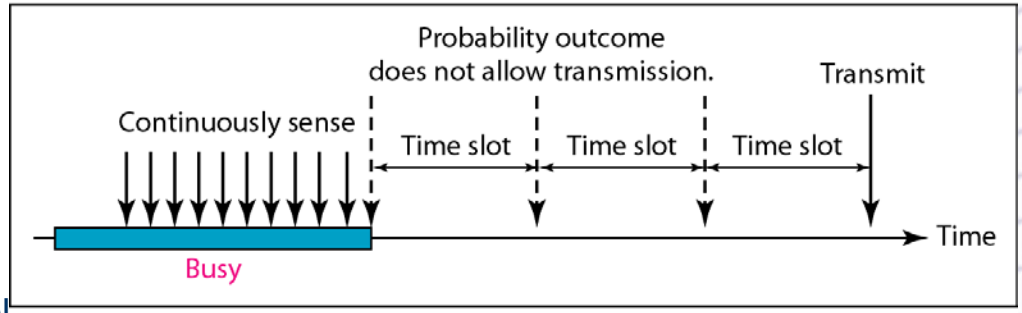
- Cosa fare se il mezzo risulta occupato? O se invece risulta libero?
- 1-persistent
- Non persistent
- P-persistent



a. 1-persistent



b. Nonpersistent



c. p-persistent

CSMA Persistent

- **1-persistent CSMA** (Carrier Sense Multiple Access) Quando la stazione vuole trasmettere controlla prima se il canale è libero trasmette ma se sente occupato aspetta finchè torna libero. Se comunque c'è collisione aspetta un tempo random e poi ci riprova. Si chiama **1-persistent** perchè trasmette con probabilità 1 (100%) quando trova il canale idle
- Importante per le prestazioni il **delay di propagazione**. Appena una stazione sta per trasmettere anche un'altra lo vuole fare. La seconda sente il canale libero e comincia a trasmettere perchè non è ancora arrivato il frame della prima.
- Anche con delay zero succedono collisioni. Es: due stazioni aspettano che una terza finisca di trasmettere e immediatamente cominciano. Collisione immediata! Se solo non fossero così impazienti!
- Comunque questo protocollo risulta più efficiente di ALOHA

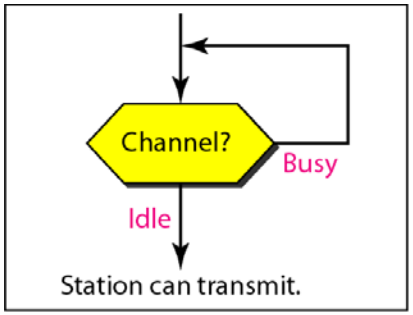
Non Persistent

- **nonpersistent** CSMA:
  - Il protocollo cerca di essere meno avido. Se è idle comincia a trasmettere, ma se il canale è in uso non continua a testarlo per prenderne possesso appena libero. Aspetta invece un periodo random
  - Migliore utilizzazione del canale (difficile che due aspettino lo stesso tempo random) ma peggiore delay rispetto a 1-persistent

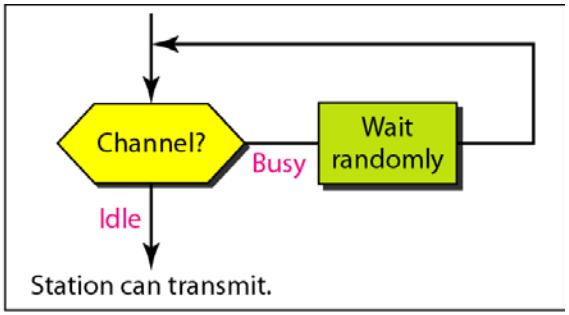
P-Persistent

- **p-persistent** CSMA:
  - su canali slotted, quando la larghezza degli slot è maggiore o uguale al tempo di propagazione
  - Se sente il canale libero
    - trasmette con probabilità ***p*.**
    - Altrimenti non trasmette (quindi con probabilità ***q=1-p***) e aspetta la prossima slot. Se anche questa è idle trasmette o aspetta con probabilità di nuovo *p* e *q*
  - Se invece è occupato si comporta come se ci fosse stata una collisione con una procedura di backoff

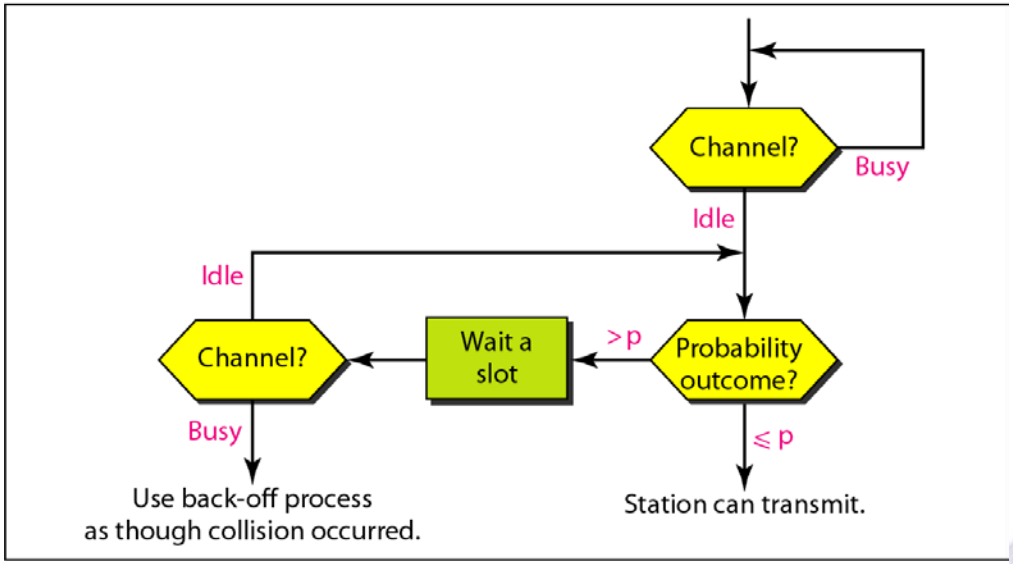
Diagrammi di flusso



a. 1-persistent

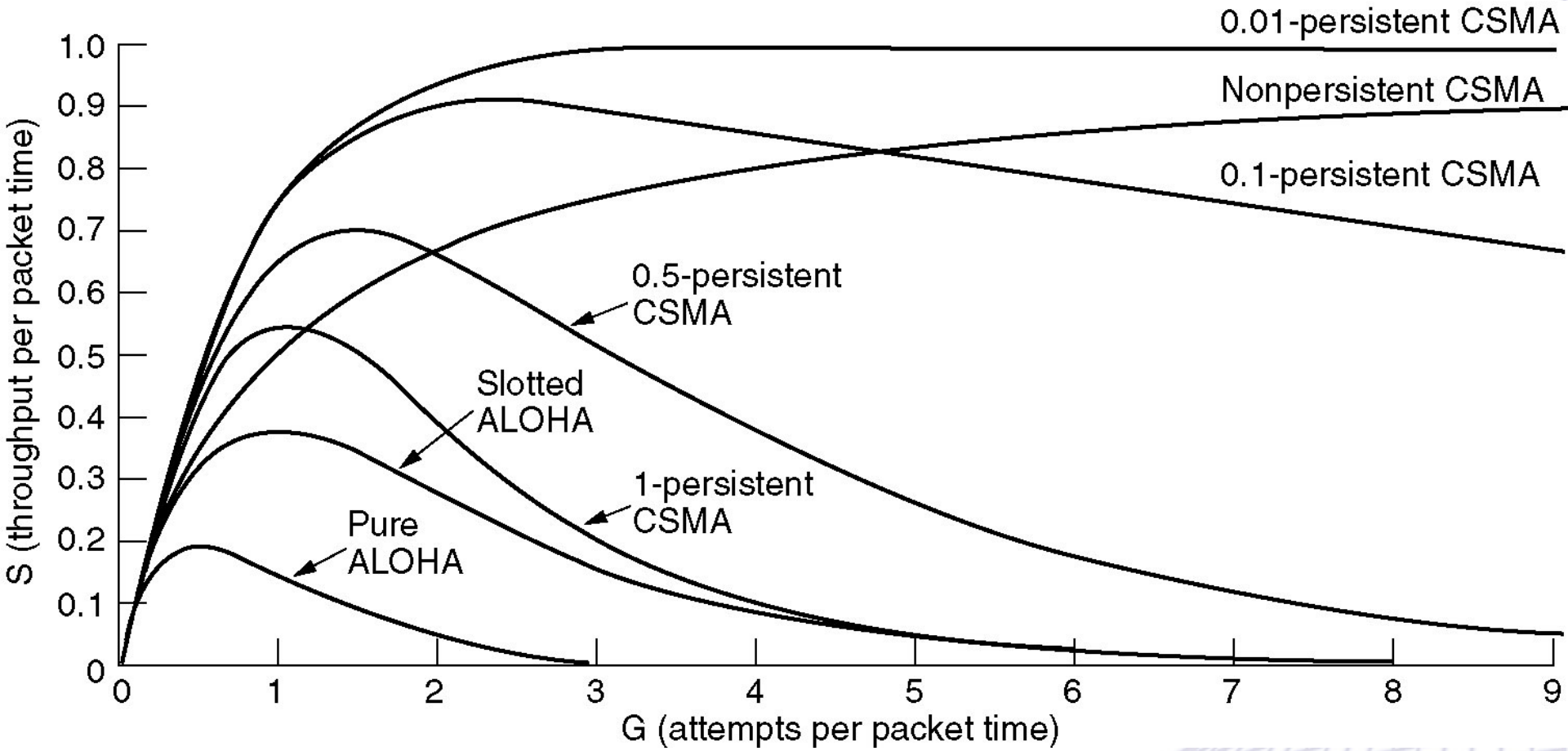


b. Nonpersistent



c. p-persistent

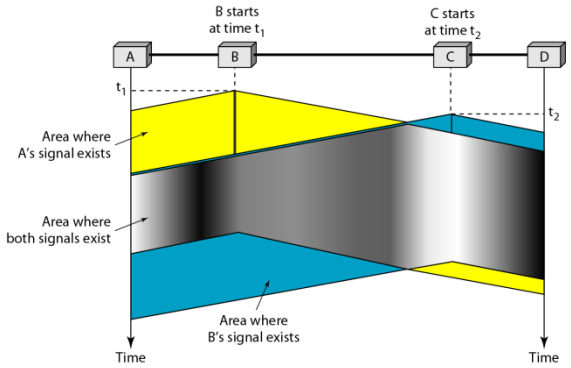
Throughput degli Aloha

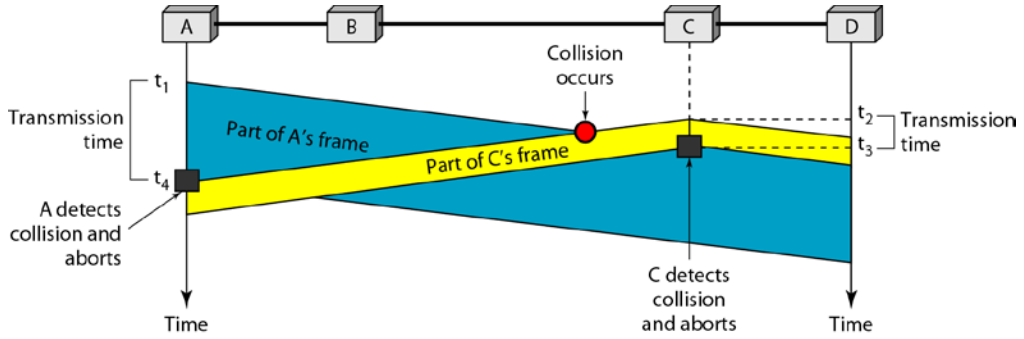


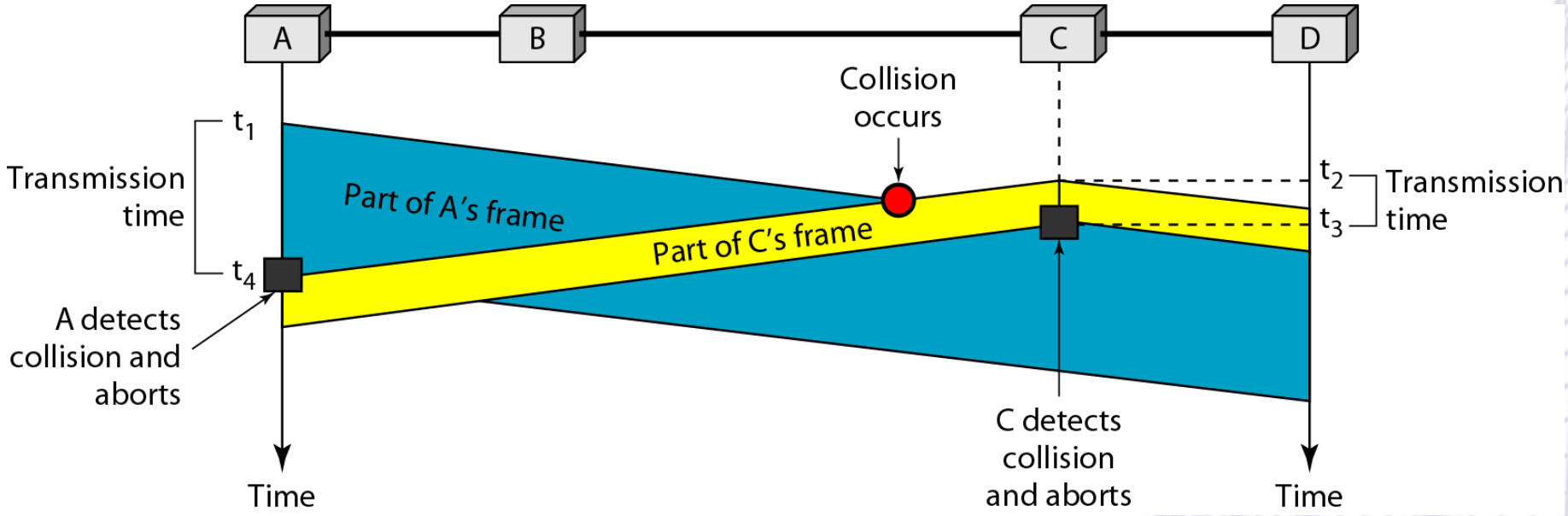
Collision Detection

- I protocolli **non-persistent** e **persistent** sono un miglioramento non trasmettendo quando sentono il canale busy
- Altro miglioramento: Interrompere la comunicazione non appena sentono la collisione. Capire subito che i frame sono compromessi risparmia tempo.
- Il protocollo si chiama **CSMA/CD** (CSMA with **Collision Detection**) e viene usato moltissimo nelle LAN a livello MAC per esempio nelle **Ethernet**

Interruzione dopo collisione

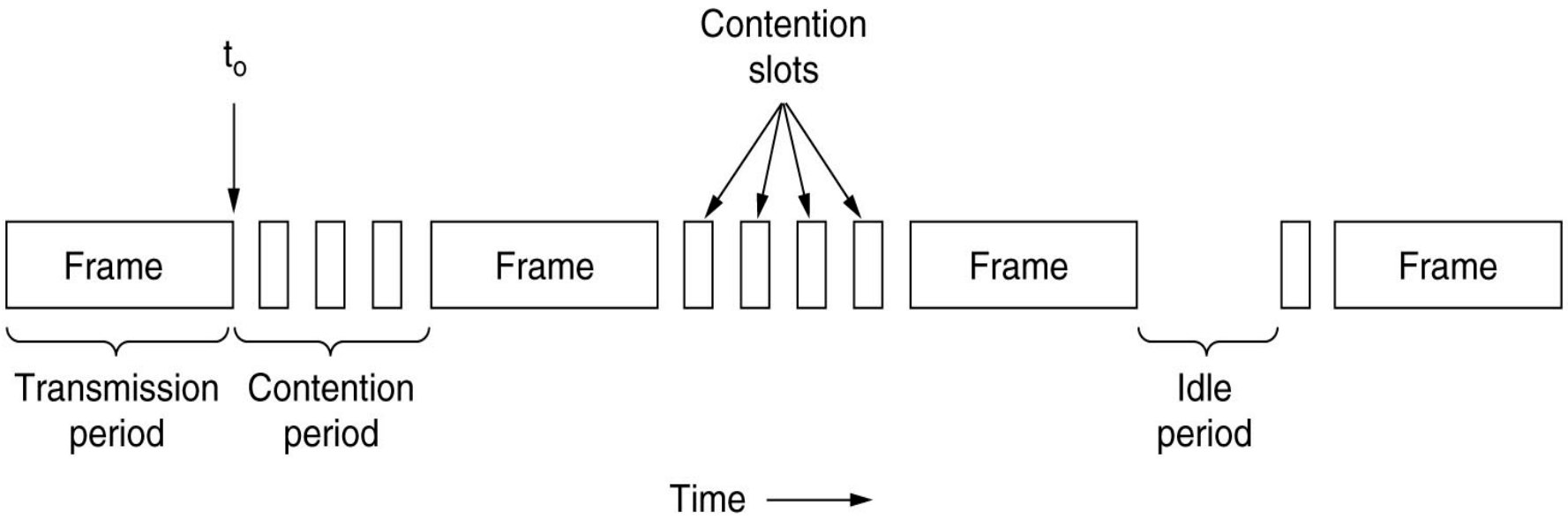






Gli stati del CSMA/CD

- Contention
- Trasmissione
- Idle

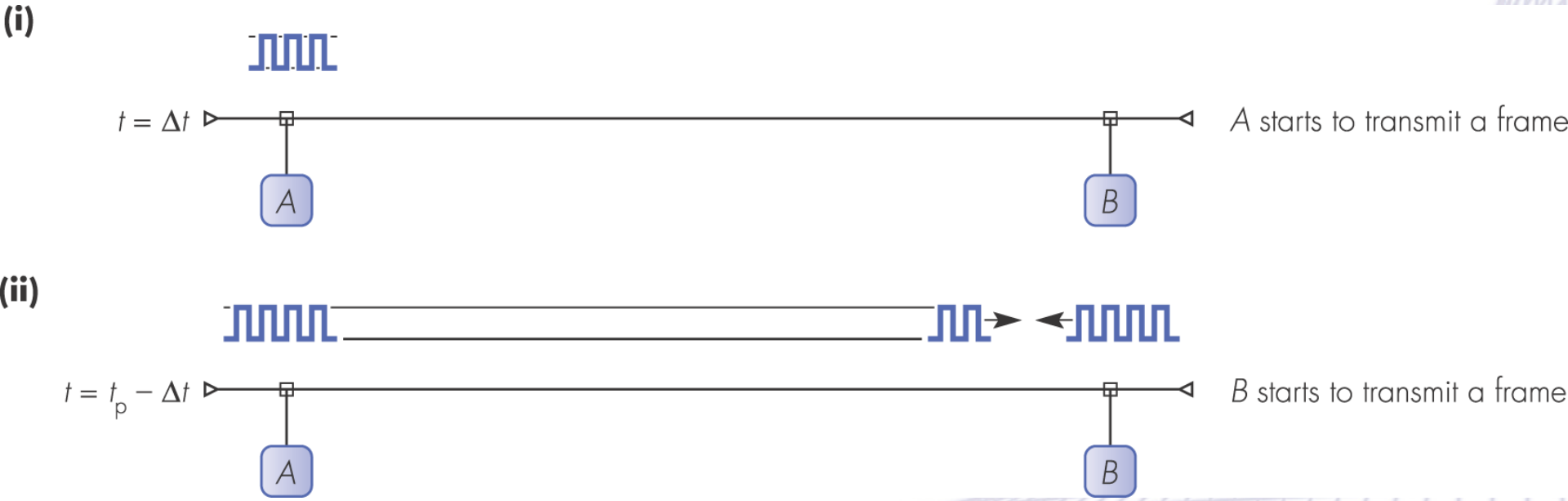


Algoritmo di contention

- Prendiamo due stazioni che vogliono trasmettere a t_0
  - Quanto devono aspettare per capire se c'è stata collisione? Importante per determinare la lunghezza del periodo di contention e quindi determinare il delay e il throughput
  - Risposta: E' il tempo τ necessario per progagare il segnale tra le due stazioni più lontane.

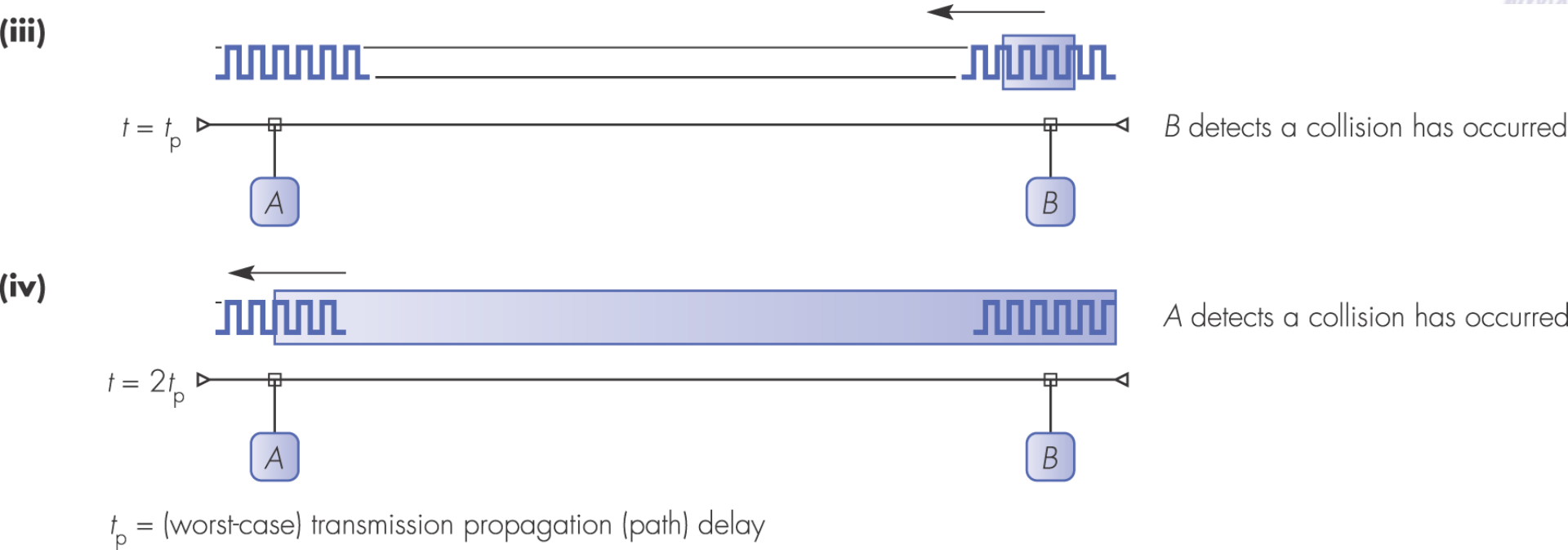
Algoritmo di contention

- Questo non significa che dopo aver aspettato quel tempo τ ha il possesso esclusivo del canale. Se al tempo t_0 una stazione trasmette, al tempo $\tau - \varepsilon$, un istante prima che il segnale arrivi alla stazione più distante un'altra comincia a trasmettere. Questa sente quasi subito la collisione e si ferma ma la prima non sente la collisione fino al tempo $2\tau - \varepsilon$.



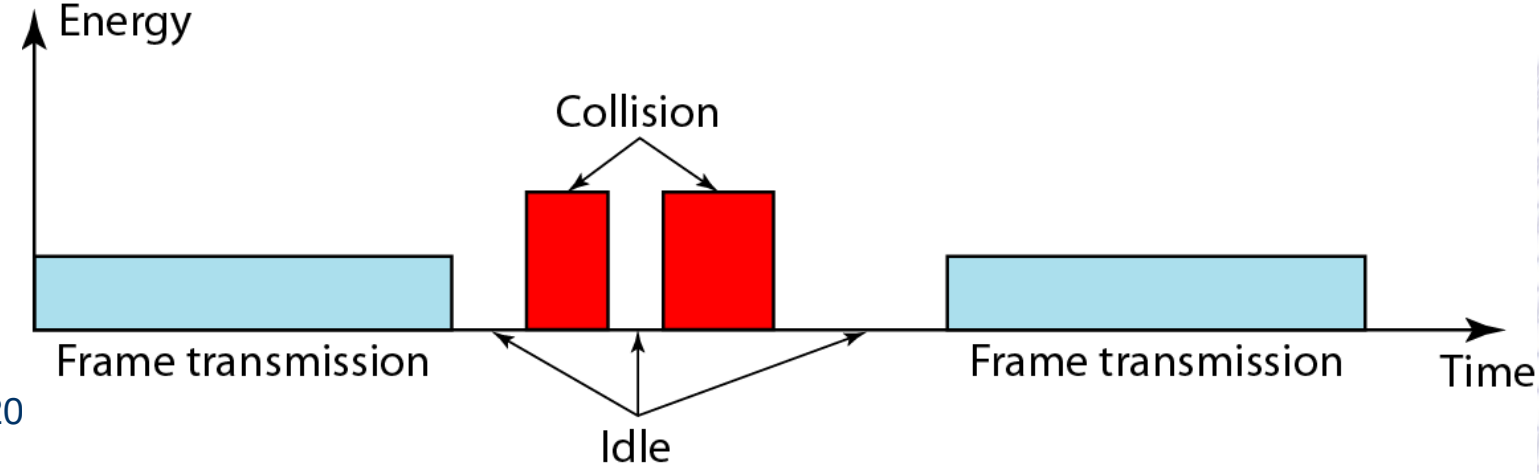
Algoritmo di contention

- Quindi non posso essere sicuro di aver avuto il canale fino al tempo 2 τ dopo la trasmissione
- Quindi il tempo di trasmissione deve essere almeno il doppio del tempo di propagazione massimo



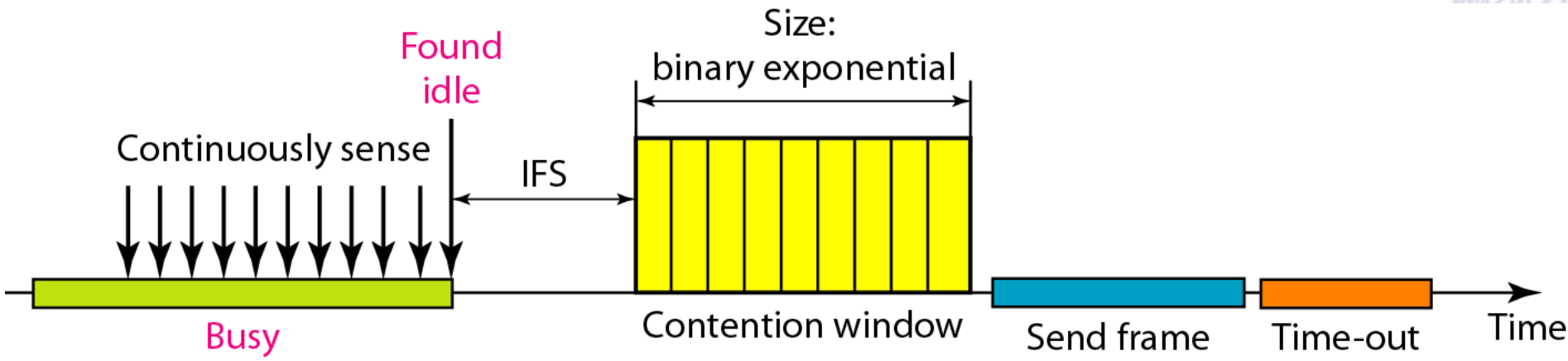
Collision Detection

- Importante: La collision detection è un fenomeno analogico.
- Se la stazione vede sul canale qualcosa di diverso da quanto trasmesso sa che c'è una collisione.
- Serve un encoding particolare essendo impossibile vedere una collisione tra due segnali a 0-volt
- Impossibile trasmettere e ricevere allo stesso momento. L'elettronica di ricezione è occupata a sentire le collisioni durante la trasmissione → il sistema è di tipo half-duplex
- Posso avere solo 3 tipi di livelli di energia del segnale, zero (idle) normale (trasmissione) e anormale (collisione)



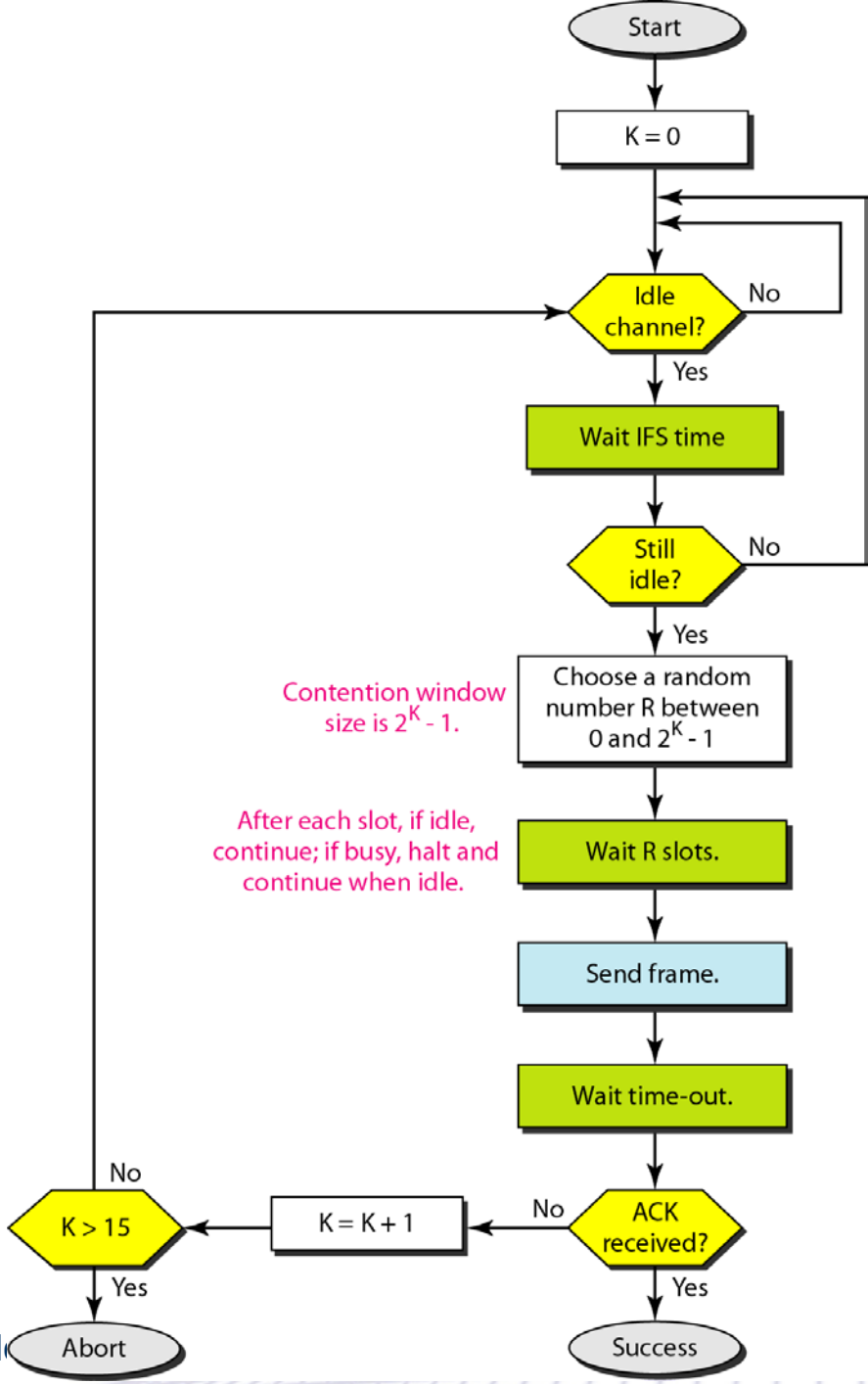
CSMA/CA

- Collision Avoidance.
  - In un sistema wireless si cercano di evitare le collisioni, dal momento che non è facile capire dai livelli di energia se c'è stata una collisione (non si raddoppia l'energia di segnale, magari solo 5 o 10% aggiuntivo, l'energia si perde nella trasmissione)
  - Vedremo nelle prossime slides (8-Unife-xxxx) le tecniche



Contest window

- Se dopo IFS è ancora idle aspetto un tempo chiamato **finestra di contesa**
- Si tratta di un tempo diviso in intervalli, la stazione dopo IFS aspetta un numero random di intervalli (scelto con il backoff esponenziale), quindi la prima volta al massimo un intervallo e poi ogni volta raddoppia

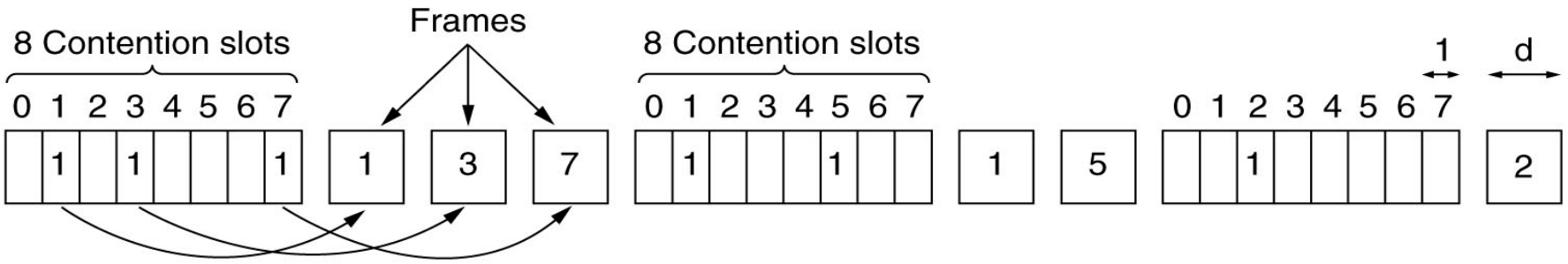


Accesso Controllato

- A prenotazione
  - prima di spedire la stazione prenota il canale, il tempo viene diviso in intervalli, ognuno dei quali è preceduto da un frame di prenotazione, fatta da N parti quante sono le stazioni
- A Polling
  - Una stazione master chiede prima chi deve parlare e poi le secondarie seguono le direttive della master
- A Token
  - Le stazioni sono in un anello logico (e magari anche fisico)
  - La stazione aspetta l'arrivo del token, quando arriva lo tiene, spedisce i propri dati e quindi libera il token
  - Se una stazione non ha dati da spedire lascia circolare il token

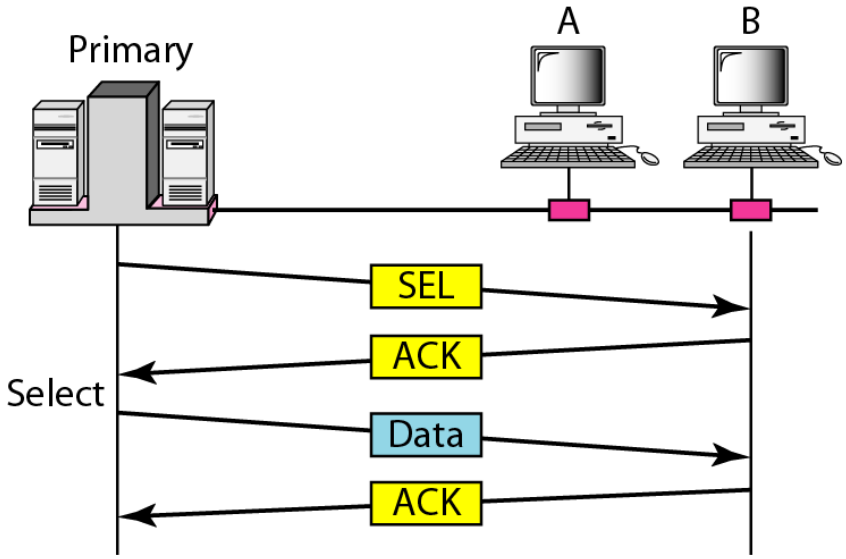
Reservation protocol

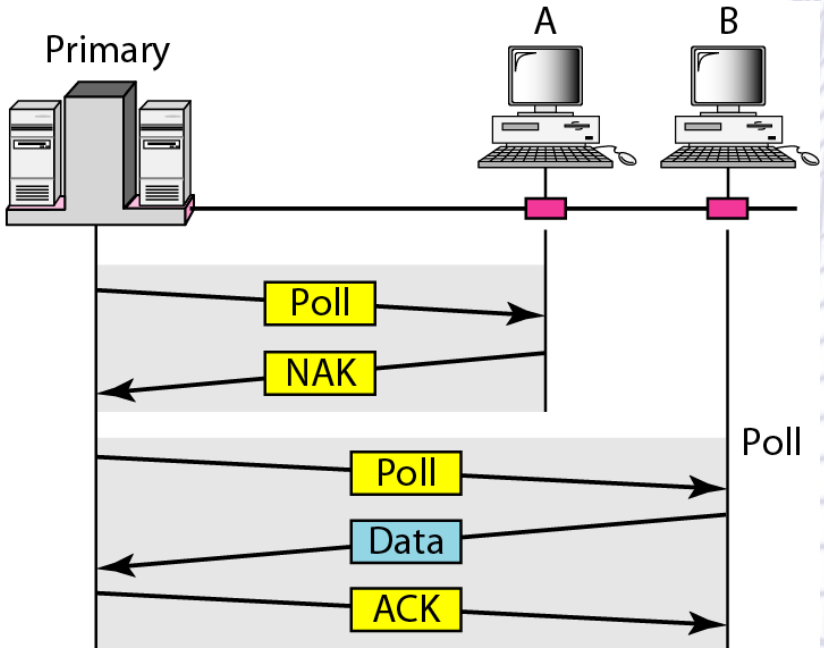
- Prenotazione con 8 stazioni
  - Prima si prenotano 1,3 e 7
  - poi le stazioni 1 e 5
  - poi la stazione 2



Polling

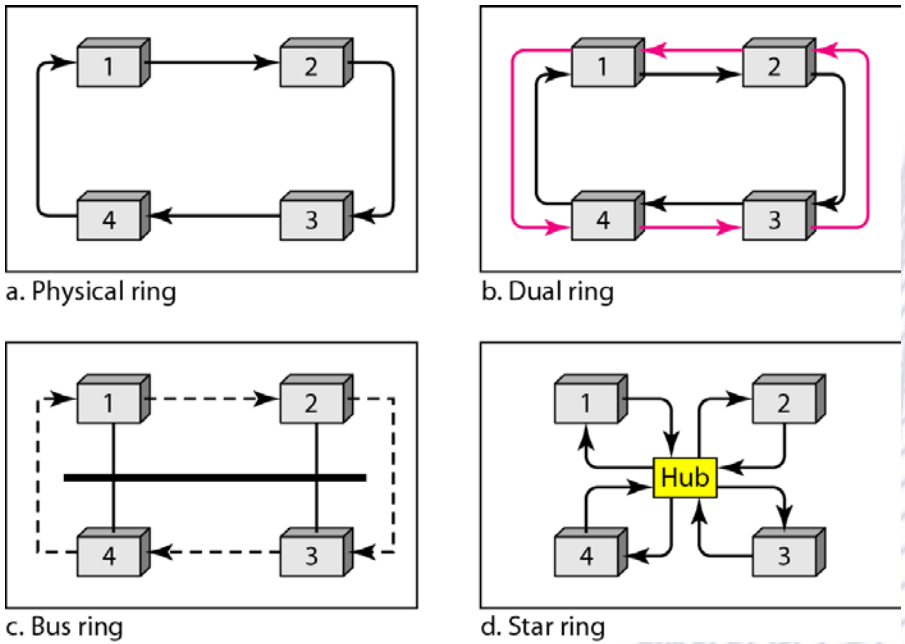
- Funzione di polling (a destra): la master fa un sondaggio (polling) tra le secondarie per sapere se devono mandare dati
- Funzione di select (a sinistra): la master chiede se qualcuno ha dati da spedire e raccoglie i dati





Token

- a) L'anello logico è anche anello fisico (token ring)
- b) doppio anello (FDDI)
- c) l'anello logico è invece un bus fisico
- d) oppure una strutta ad hub in cui magari la numerazione delle porte definisce l'anello



Canalizzazione

- La larghezza di banda viene divisa tra tutte le stazioni in FDMA, TDMA o CDMA