

Slotted Aloha

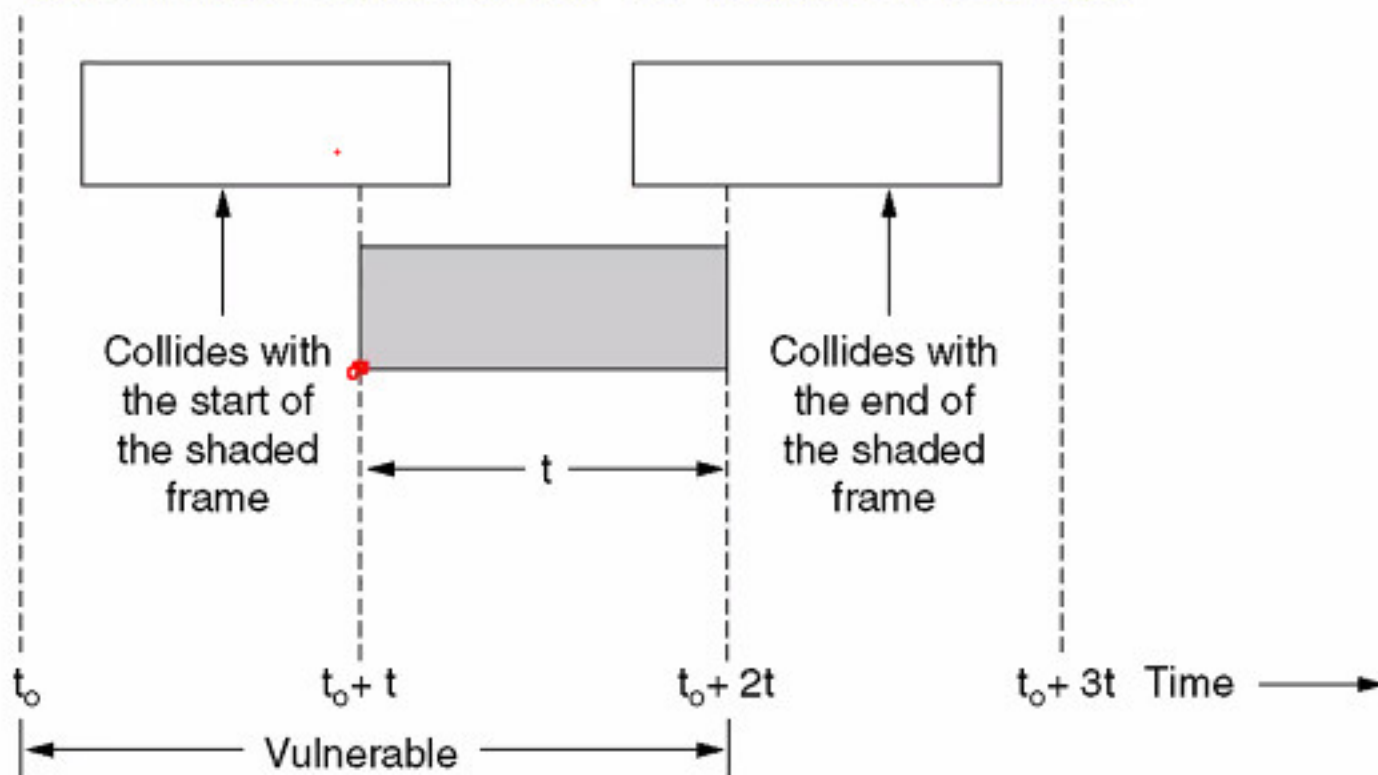
- ◆ Due anni dopo Aloha, Roberts trova un metodo per migliorarla:
- ◆ Assume che il tempo sia ***slotted***, tramite una stazione principale che manda un segnale di sincronizzazione
- ◆ In tal modo, una stazione deve attendere prima di trasmettere il pacchetto, non c'è trasmissione istantanea



Slotted Aloha

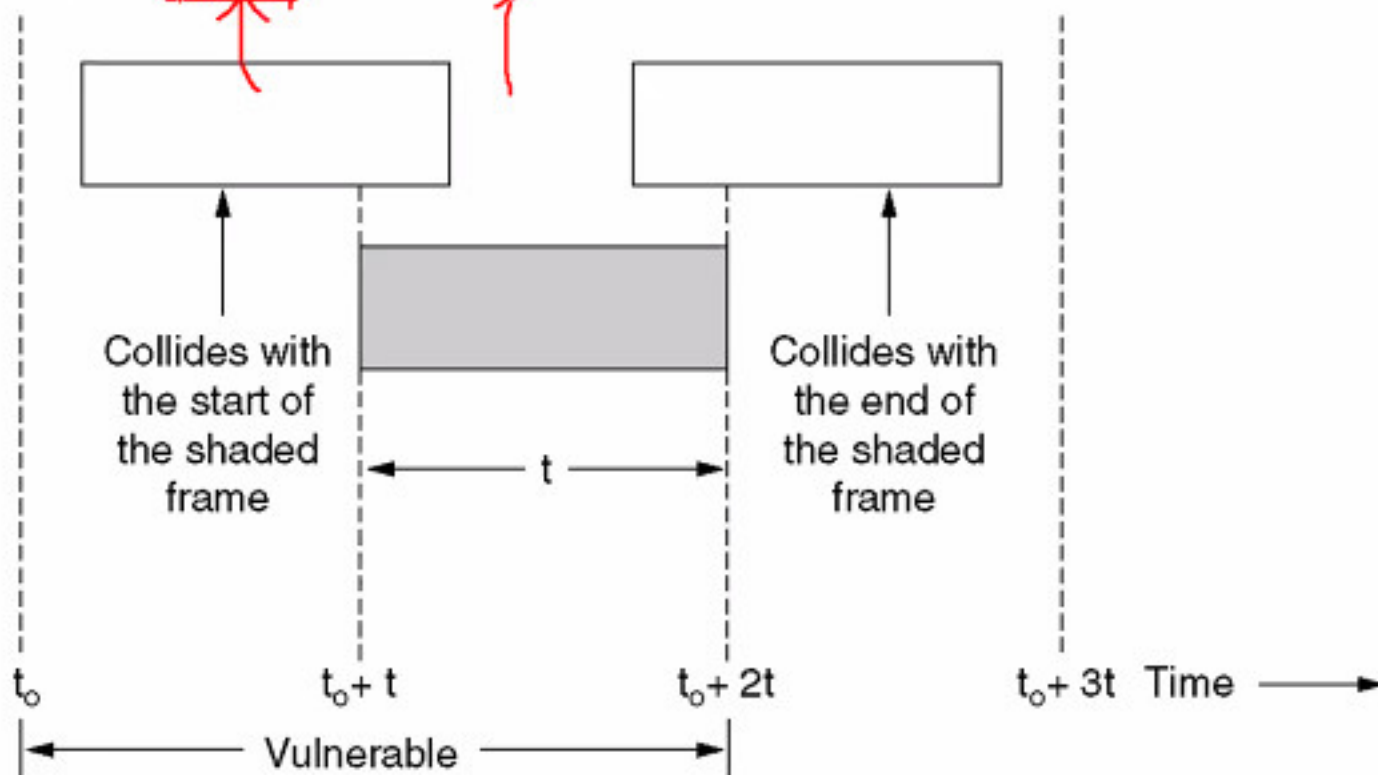


- ◆ Come cambiano le prestazioni?
- ◆ Il periodo "critico" che può generare conflitti stavolta è ***dimezzato***



Slotted Aloha

- ◆ Ora la probabilità di successo
($\Pr[k] = (G^k * e^{-G}) / k!$)
è $\Pr[1] = G * e^{-G}$



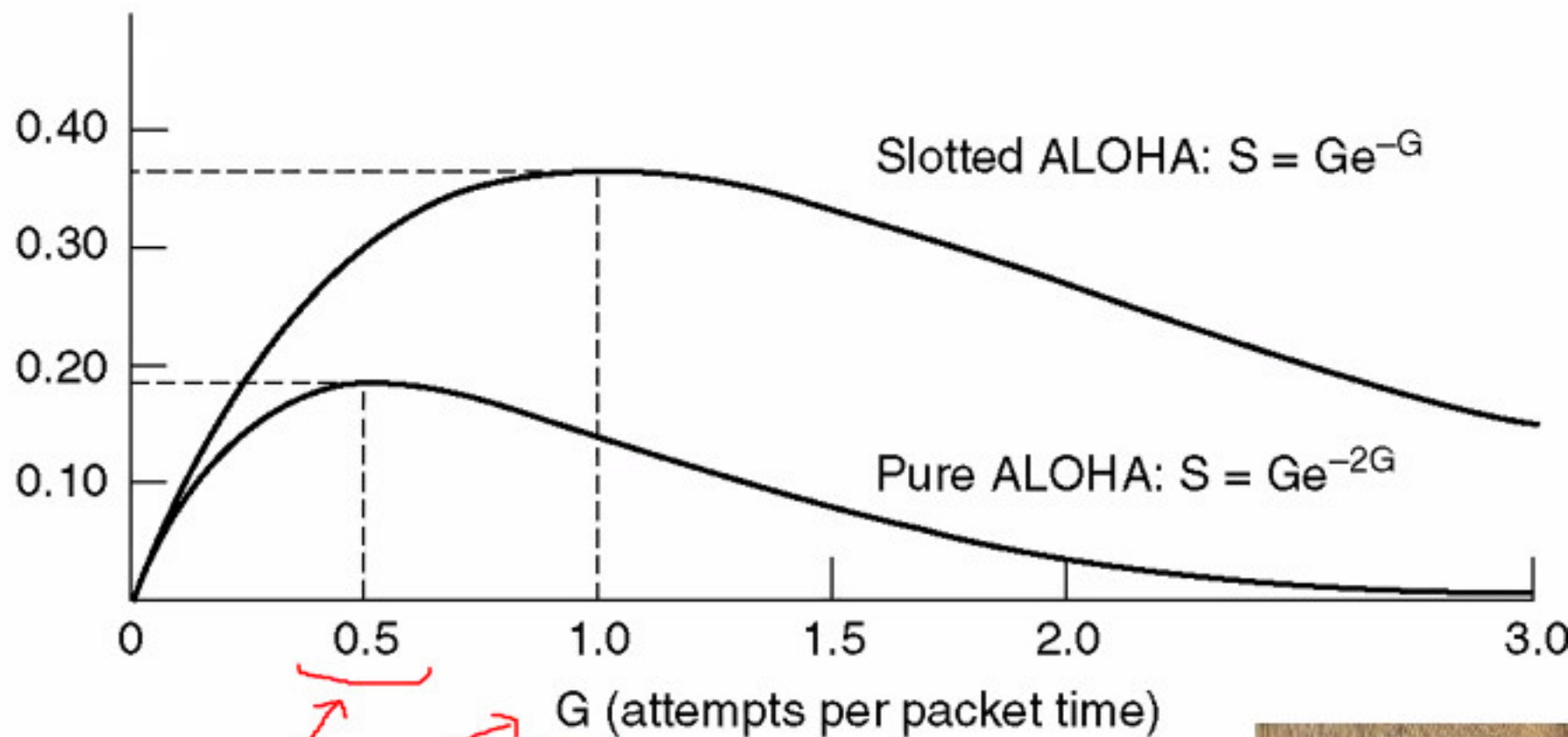
Quindi... G^*e^{-G}

- ◆ Il numero massimo di frame al secondo si ha con $G=1$, e diventa:
 $1 / e$
- ◆ Cioè circa **36.8%**, il doppio di Aloha (!)



Pure e Slotted Aloha a confronto

S (throughput per frame time)



Notare..

- ◆ Abbiamo raddoppiato la velocità, alterando una delle opzioni a nostra disposizione...
- ◆ Si può fare ancora meglio, proseguendo su questa linea...?



Carrier Sense

- ◆ Gli Aloha, pure e slotted, hanno ***no carrier sense***, cioè non hanno modo di sapere se ci sono già trasmissioni in atto
- ◆ Magari introducendo il carrier sense le cose migliorano...



1-persistent CSMA

- ◆ CSMA sta per ***Carrier Sense Multiple Access*** protocol
- ◆ → prima di trasmettere, controlla che non ci sia già una trasmissione (carrier sense)
- ◆ Se c'è una trasmissione, controlla il canale, e non appena si libera, trasmette

1-persistent CSMA

- ◆ Può lo stesso succedere che ci siano collisioni (Murphy...): in questo caso?
- ◆ Si riaspetta un certo ***tempo casuale***, e poi si riprova
- ◆ Performance: aumenta a più del **50%** (!!)

Difetto di 1-CSMA?

- ◆ Se molte stazioni stanno aspettando durante una trasmissione, alla fine della trasmissione tutte cercano di trasmettere contemporaneamente, provocando una collisione
- ◆ Se fossero un po' meno egoiste, il mondo funzionerebbe meglio...

p-persistent CSMA



- ◆ Si ottiene rilassando l'ipotesi che uno debba sempre trasmettere ($p=1$) quando trova la linea libera, ma invece, trasmette con probabilità p (è più gentile ☺ e non deve sempre trasmettere a tutti i costi)
- ◆ Performance? Migliora, con molte stazioni, al diminuire di p
- ◆ Al limite (vedremo), può arrivare anche al **100% (!!!!)**

Nonpersistent CSMA

- ◆ Usa un metodo alternativo: invece di tenere d'occhio il canale per ricominciare la trasmissione non appena è libero, se trova il canale occupato aspetta un periodo di tempo casuale e poi riprova

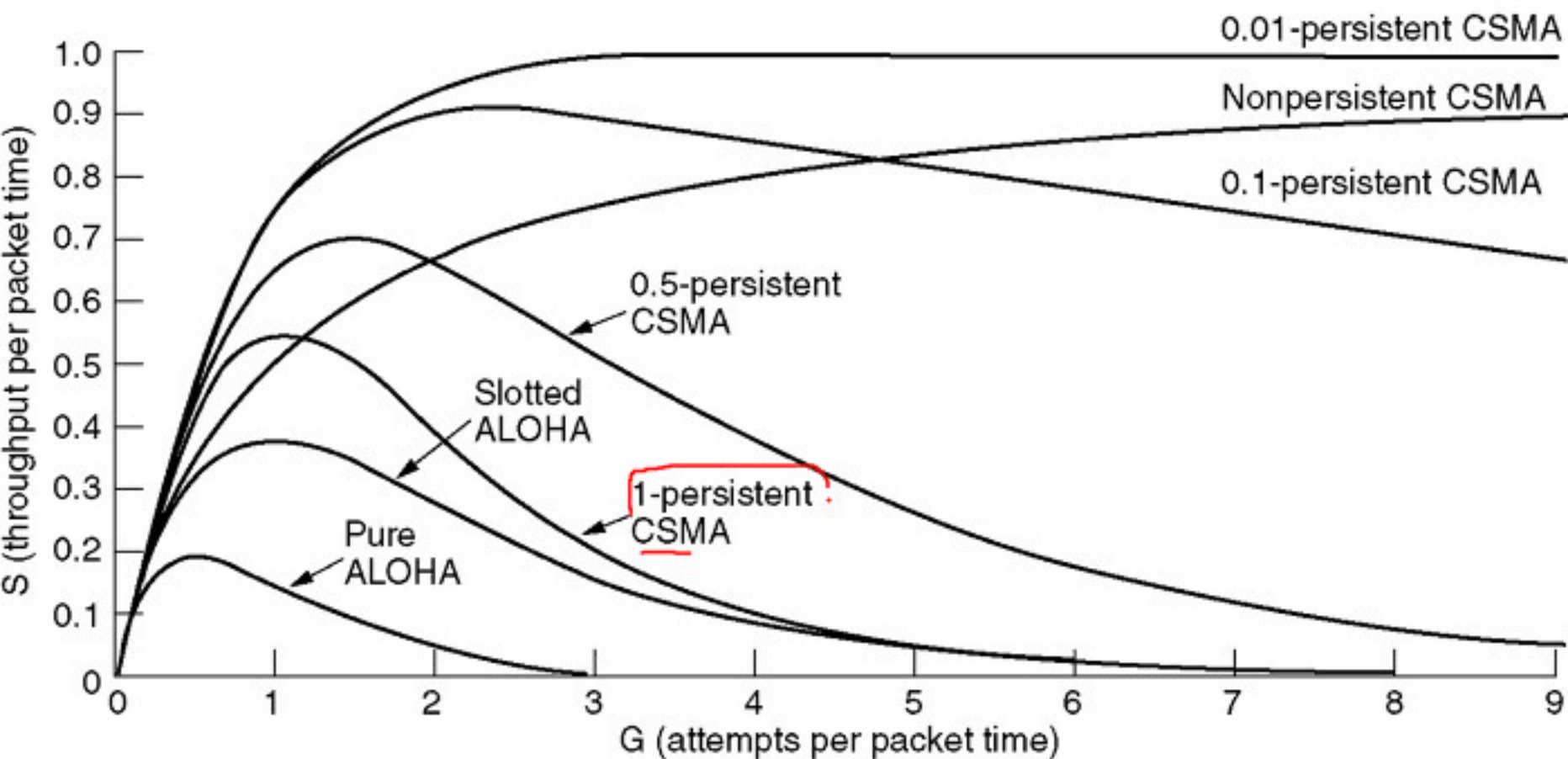
Nonpersistent CSMA

- ◆ Usa un metodo alternativo: invece di tenere d'occhio il canale per ricominciare la trasmissione non appena è libero, se trova il canale occupato aspetta un periodo di tempo casuale e poi riprova

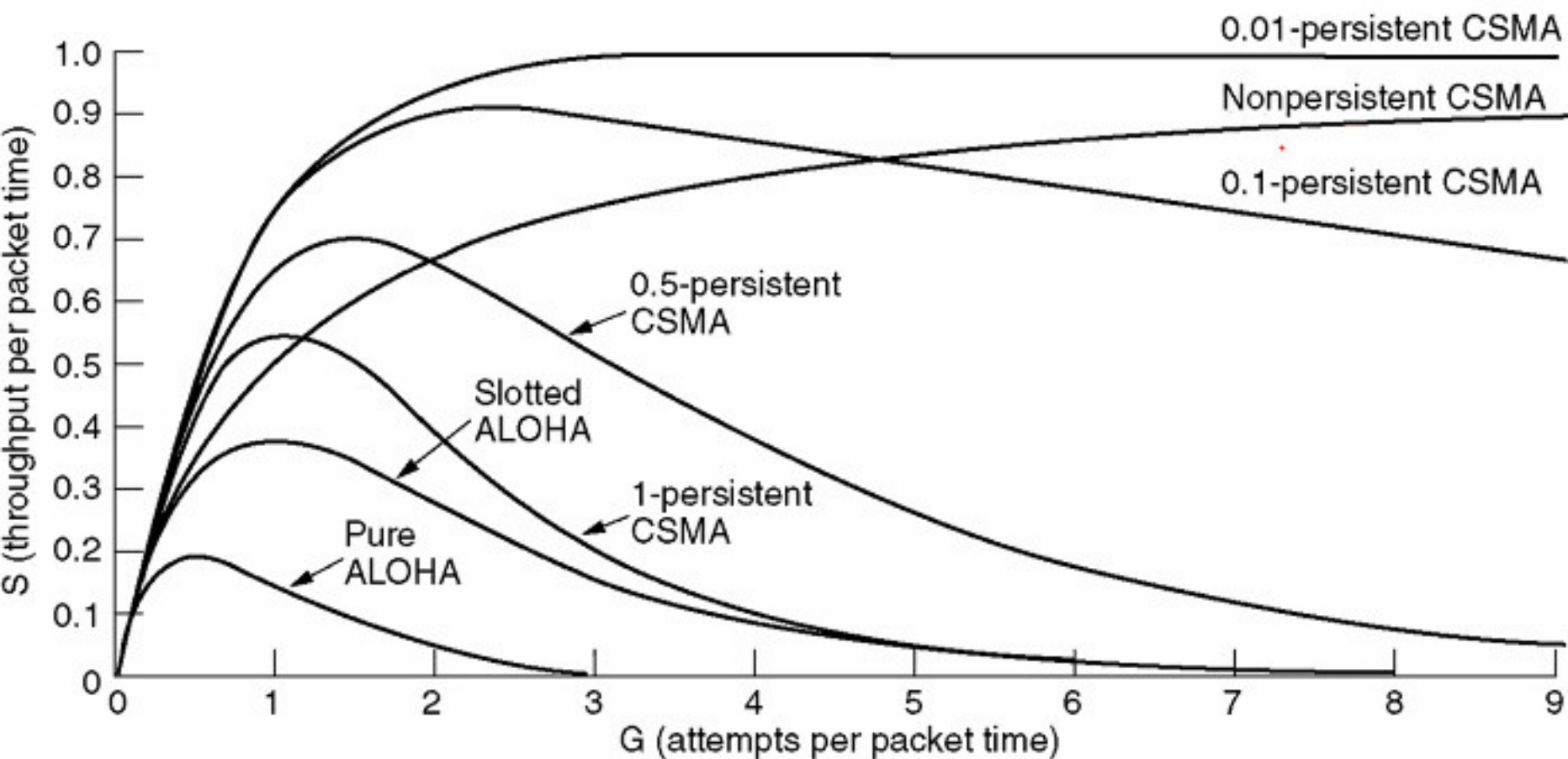
Nonpersistent CSMA

- ◆ Performance: la curva di performance si comporta differentemente da p-CSMA, e può raggiungere performance vicine al **90%** (!)

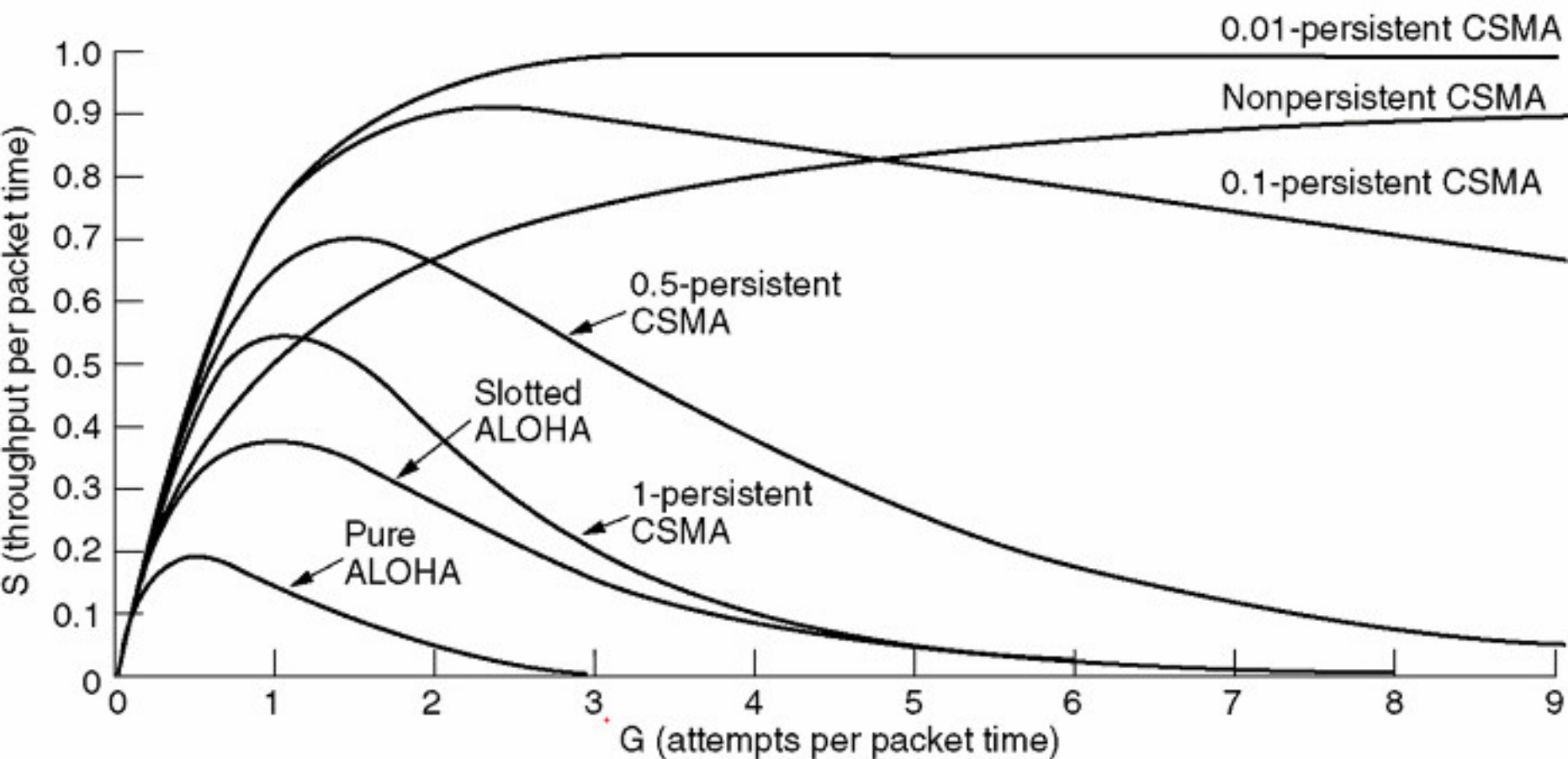
CSMA: confronto



CSMA: confronto



CSMA: confronto

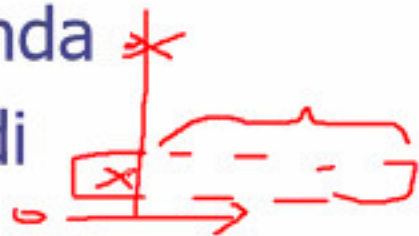




CSMA/CD



- ◆ Avendo il carrier sense, possiamo anche essere più furbi quando trasmettiamo: non controlliamo solo quando vogliamo spedire il segnale, ma anche durante tutta la trasmissione, e se ci accorgiamo di una collisione, ritrasmettiamo dopo:, sprecando dunque meno banda
- ◆ (e magari smettiamo anche subito di trasmettere...)
- ◆ Quindi, rilassiamo l'assunzione ***station model***, possiamo fare anche altro mentre trasmettiamo

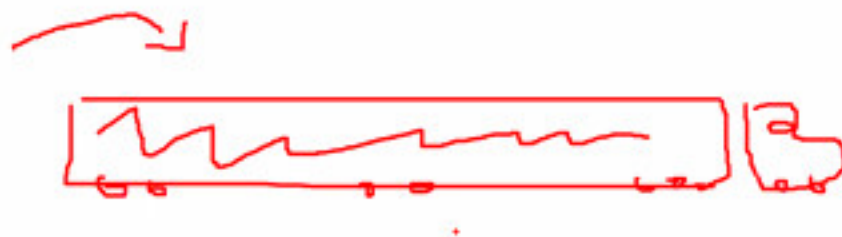


CSMA/CD



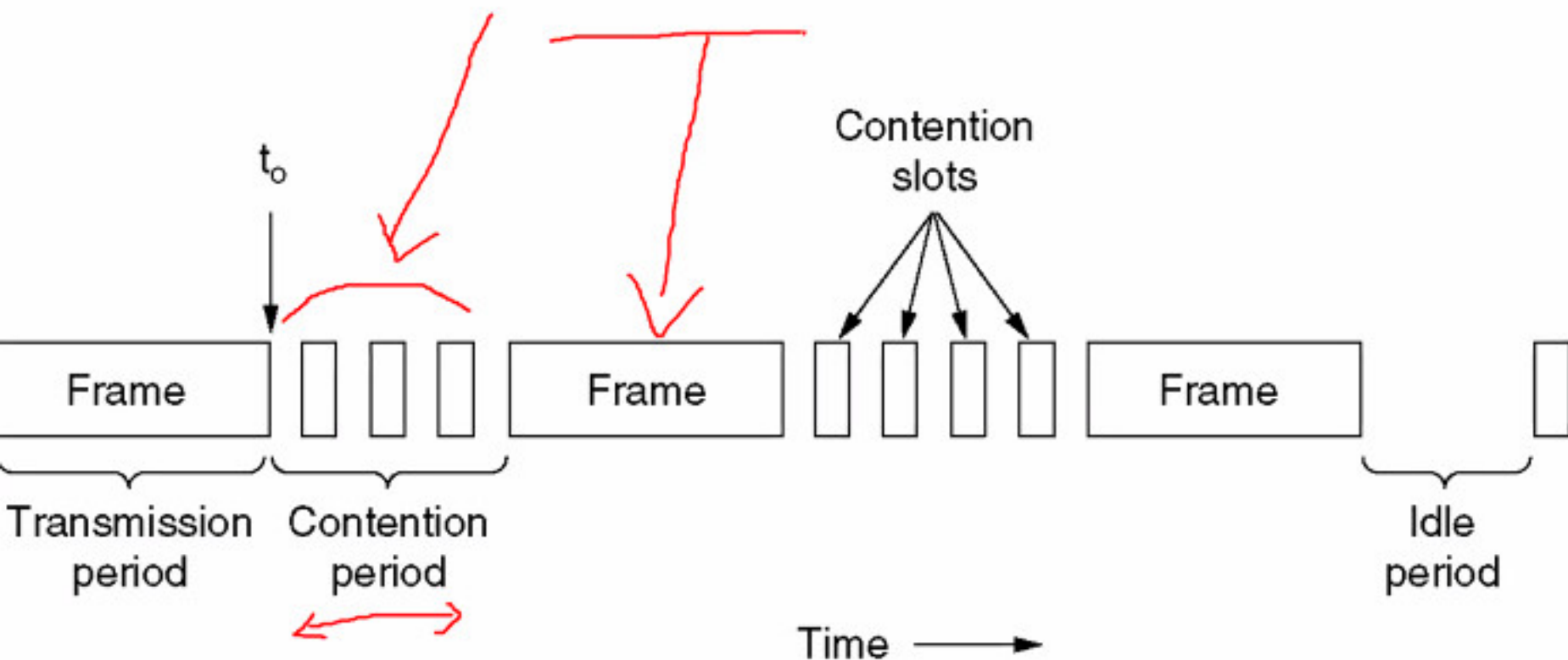
- ◆ Questo tipo di trasmissione dà luogo ai ***CSMA con Collision Detection (CSMA/CD)***
- ◆ CSMA/CD tratta le collisioni come Aloha, cioè se c'è una collisione, aspetta un certo periodo di tempo casuale prima di riprovare a trasmettere

CSMA/CD

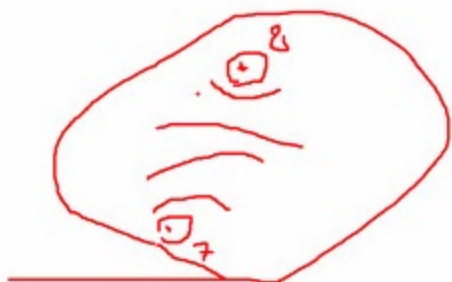


- ◆ In CSMA/CD, possiamo separare tre possibili stati della rete:
trasmissione,
contention,
e **idle:**

CSMA/CD



La parte importante



- ◆ E' nella parte di contention, cioè decidere **quanto tempo** allocare a quella zona in cui si decidono le sorti della trasmissione
- ◆ Il tempo da allocare sarà **due volte** il tempo di trasmissione tra le due stazioni più distanti
(→ il tempo di **roundtrip**)