Capitolo 5

L' ACCESSO CASUALE

5.1 Origini dell'accesso casuale

Nel problema dell' Accesso Multiplo (AM), l'approccio più naturale è stato, fin dalle origini, quello di operare uno scambio di informazioni fra le stazioni utenti in modo da poter preordinare l'accesso al canale. Chiameremo questo approccio Accesso Preordinato (AP).

Si immagini che nell'accesso venga coinvolto un grande numero M di stazioni e che M sia così grande che, con traffico finito, il traffico fornito da ciascuna di queste sia costituito da pacchetti o messaggi estremamene rari. In questo caso le singole stazioni trasmettono rararmente e ad ogni istante il numero di stazioni che ha qualcosa da trasmettere è solo di qualche unità. L'Accesso Preordinato soffre tuttavia della necessità del coordinamento, che necessariamente deve far uso della risorsa di trasmissione che si vuole, appunto, condividere. L'Accesso Preordinato, per esempio a token, continua ad interrogare tutto il grande numero di stazioni anche se a trasmettere sono solo poche e la maggior parte del tempo viene spesa nell'interrogazione portando l'efficienza a zero.

L'Accesso Casuale nasce appunto per evitare, o limitare in qualche modo, questa difficoltà. Come caratteristica base si accetta il fatto che, a seguito della mancanza di coordinamento, stazioni diverse possano trasmettere in modo tale che le loro trasmissioni vengano a sovrapporsi, anche parzialmente, (al ricevitore). Queste sovrapposizioni vengono chiamate *collisioni*.

L'effetto delle collisioni dipende dal mezzo trasmissiovo e dalle tecniche di modulazione/codifica utilizzate sul livello fisico. Se nella collisione una componente prevale a livello di energia sulla somma di tutte le altre allora può essere possibile la *cattura* da parte del ricevitore della componente che prevale. Può anche succedere che a ricevitori diversi si catturi una componente diversa. Spesso però questo non succede e una collisione porta alla distruzione di tutti i pacchetti coinvolti. Questa è l'assunzione base che utilizzeremo nel seguito sino ad esplicito avviso. I pacchetti collisi devono venire in qualche modo scoperti e ritrasmessi. Le assunzioni e le regole che sovraintendono a tali procedure sono appunto i protocolli d'accesso.

I protocolli d'accesso descrivono le regole che ogni stazione deve seguire per trasmettere. Le infor-

mazioni su cui il protocollo opera possono essere molteplici. Sicuramente vi è necessità del riscontro (ACK) dell'avvenuta corretta ricezione. Questo, in generale, lo si suppone fornito in modo indipendente rispetto al canale usato e può essere fornito da osservazione diretta (es caso di canale broadcast o con eco) o indiretta (è il ricevente che segnala la corretta ricezione). Altre informazioni tipiche sono quelle fornite dal cosiddetto *feedback di canale*, che sono le informazioni che una stazione può ottenere da un osservazione, più o meno articolata, del canale. Una distinizione fondamentale riguarda proprio le assunzione sul feedback di canale, che possono essere molteplici e che specificheremo di volta in volta.

5.2 Protocolli senza feedback

5.2.1 ALOHA [Abramson, 1970]

E' il sistema più semplice e fa a meno di ogni forma di coordinamento fra le stazioni. Necessita solo di una forma di riscontro dell'avvenuta corretta trasmissione, solitamente supposta ottenuta dalla stazione ricevente, e viene perciò classificato come "senza feedback di canale".

- Pacchetti nuovi vengono trasmessi appena generati.
- Pacchetti collisi vengono ritrasmessi dopo un tempo X variabile casuale di opportune caratteristiche.

La seconda parte è evidentemente dettata dalla necessità di scorrelare trasmissioni che hanno precedentemente colliso. Leggi deterministiche di trasmissione, prescindendo da ogni coordinamento possono portare a collisioni e a ritrasmissioni. Di primaria importanza è dunque l'analisi del protocollo che fornisca prestazioni di throughput e di ritardo.

Lo studio del processo $\mathbf{N(t)}$, definito nel precedente capitolo, risulta molto complesso, per cui sono stati proposti dei metodi di analisi approssimati. Una prima valutazione molto semplice si può ottenere assumendo che *il processo del traffico di canale, ossia delle complessive trasmissioni e ritrasmissioni, sia assimilabile a un processo puntuale di Poisson*, la cui frequenza, è indicata con $G = \lambda T$, assumendo pacchetti di durata costante T. Il modello citato è chiamata: *modello di popolazione infinita* perchè l'ipotesi Poissoniana è giustificato quando i flussi di ogni singola stazione sono piccolissimi ($G/M \to 0$).

Con questo modello, la probabilità che un pacchetto venga trasmesso senza essere disturbato da altri è la probabilità che nessun altro pacchetto venga trasmesso T secondi prima e T secondi dopo l'inizio della trasmissione del pacchetto in oggetto. Per l'assunzione di popolazione infinita il flusso disturbante ha ancora frequenza media G e la probabilità di non interferenza vale

$$e^{-2G}.$$

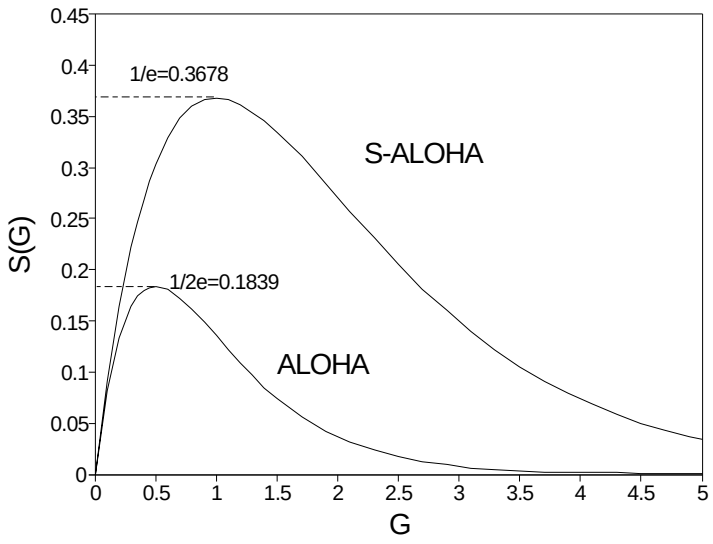

Figura 5.1 *Curve di throughput dei protocolli ALOHA e Slotted Aloha.*

Il numero medio di pacchetti trasmessi con successo, e dunque il throughput trasportato per ogni slot, risulta allora essere

$$S = Ge^{-2G}. \tag{5.1}$$

L'analisi della curva (5.1) riportata in Figura 5.1, mostra che il massimo throughput trasportato è $1/2e = 0.1839$ pacchetti per tempo T in $G = 1$, ma soprattutto che la curva non cresce monotonicamente e che al crescere di G scende asintoticamente a zero.

Si noti che l'ipotesi alla base del modello, prima ancora di assumere la distribuzione di trasmissioni e ritrasmissioni assume che queste costituiscano un processo stazionario a media G. Resta da vedere se S nuove trasmissioni per tempo T, originano un processo che unito alle ritrasmissioni si stabilizza nel processo stazionario ipotizzato. Ciò è messo in dubbio dal comportamento *instabile* della curva (5.1) se si interpreta come se fosse un legame fra grandezze deterministiche. In questo caso un aumento di G, in un punto che opera in $G > 0.5$ riduce il throughput S e ciò aumenta l'accumulo e dunque G. Nella realtà il traffico di canale è un processo casuale e il ragionamento citato fa pensare che anche le fluttuazioni del processo del traffico di canale oltre la media G possano causare una diminuzione di throughput e dunque un accumulo di fallimenti che tende ad aumentare la frequenza del traffico di canale. In altre parole, siamo in presenza di un feedback positivo che può portare a un effetto valanga, ossia una crescita senza limiti di G e una riduzione a zero di S.

Si noti però che il ragionamento indicato sopra è troppo qualitativo. parlare di cambiamenti di valori medi (G) è non corretto per definizione. In realtà ci si può riferire a valori medi presi su una finestra temporale finita. In ogni caso il modello visto non permette una corretta analisi ed occorre ricorrere a un modello più acuurato, che predìca effettivamente la dinamica del sistema ed eventualmente possa indicare come stabilizzarlo.

5.2.2 Slotted ALOHA [Roberts, 1972]

Questa tecnica assume, rispetto al puro ALOHA, un parziale coordinamento fra le stazioni, consistente nel sincronismo dei possibili istanti di trasmissione, che distano T (slotting). Per il resto, opera come il precedente.

Appare chiaro che il sincronismo evita il caso di interferenze parziali. l'interferenza si ha solo se altri pacchetti partono nello stesso istante del pacchetto scelto. Dunque, col modello di popolazione infinita, la probabilità di non interferenza diventa

$$e^{-G}$$

e il throughput

$$S = Ge^{-G}. \tag{5.2}$$

Il massimo throughput è questa volta $1/e = 0.3678$, (Figura 5.1) raddoppiato rispetto al precedente caso. Resta però sempre il problema della stabilità.

5.2.3 Modello a popolazione finita

Un leggero raffinamento del modello di popolazione infinita in cui si possa evidenziare il contributo di singole stazioni è basato sulla seguente assunzione:

ogni stazione trasmette sul canale un pacchetto, in un intervallo T, con probabilità σ.

In presenza di M stazioni, il throughput si ottiene facilmente con un ragionamento analogo a quello visto col modello di popolazione infinita:

$$S = M\sigma(1-\sigma)^{2(M-1)} = G(1-\frac{G}{M})^{2(M-1)} \tag{5.3}$$

Si vede facilmente che nel limite $M \to \infty$ con G costante, la (5.3) tende alla (5.1). Analogamente, nel caso slotted si ha

$$S = M\sigma(1-\sigma)^{(M-1)} = G(1-\frac{G}{M})^{(M-1)} \tag{5.4}$$

Si vede che nel caso con due soli utenti il throughput è più alto e parte da 0.5 nel caso slotted. E' poi possibile estendere l'analisi al caso in cui le stazioni abbiano traffico diverso. E' lasciato al lettore di analizzare la modifica del caso precedente in cui un solo utente trasmette con probabilità $\sigma_1 > \sigma$ (grande utente). In questo caso è possibile scrivere il throughput S_1 del grande utente che 'e diverso dal Throughput S_2 degli altri utenti complessivamente, e mostrare che $S = S_1 + S_2$ è, in questo caso maggiore che nel caso uniforme.

5.2.4 Modello Single Buffer

Un modello più raffinato, in grado di fare luce sul fenomeno della stabilità del protocollo ALOHA è il seguente.

Se si desidera studiare il sistema in condizioni in cui M è molto elevato, allora in condizione di equità fra le stazioni, ciascuna di queste genererà un traffico medio molto basso, limitato superiormente da $1/M$, al limite molto piccolo. Ciò significa che ciascuna stazione emetterà pacchetti a distanza molto grande uno dall'altro, cosicchè si può pensare che nella stazione quasi mai sarà presente più di un pacchetto alla volta in attesa di essere trasmesso. Questa osservazione suggerisce l'uso del cosiddetto *single buffer model* che, per il sistema a slots, si basa sulle seguenti assunzioni:

- ogni stazione ha un solo buffer;
- ad ogni slot, con buffer vuoto, un pacchetto viene generato (e trasmesso) con probabilità α;
- Ad ogni slot, con buffer pieno, un pacchetto viene ritrasmesso con probabilità β.

Il processo $n(t)$ numero di buffer occupati è un processo markoviano che può essere studiato facilmente. Posto

$$a(i,n) = \left(\begin{array}{c} M-n \\ i \end{array} \right) \alpha^i (1-\alpha)^{M-n-i}$$

la probabilità di i arrivi (trasmissioni) in uno slot essendoci n buffers occupati, e

$$b(i,n) = \left(\begin{array}{c} n \\ i \end{array} \right) \beta^i (1-\beta)^{n-i}.$$

la probabilità di i ritrasmissioni in uno slot essendoci n buffers occupati si ha la seguente matrice delle transizioni

$$p_{n,n+i} = \begin{cases} a(i,n) & 2 \leq i \leq (M-n) \\ a(1,n)[1-b(0,n)] & i = 1 \\ a(1,n)b(0,n) + a(0,n)[1-b(1,n)] & i = 0 \\ a(0,n)b(1,n) & i = -1 \end{cases} \tag{5.5}$$

Con le eccezioni seguenti:

$$\begin{cases} p_{0,0} = a(0,0) + a(1,0) \\ p_{0,1} = 0 \\ p_{M,M} = 1 - b(1,M) \end{cases} \tag{5.6}$$

La distribuzione stazionaria π_n può essere ottenuta numericamente se M è finito. E da qui il throughput

$$S = E[s(n)] = \sum_n \pi_n s(n) \tag{5.7}$$

con

$$s(n) = a(1,n)b(0,n) + a(0,n)b(1,n) \tag{5.8}$$

mentre il traffico sul canale risulta invece

$$g(n) = (M-n)\alpha + n\beta \tag{5.9}$$

e

$$G = E[g(n)] = M\alpha + E[n](\beta - \alpha) \tag{5.10}$$

A parità di traffico offerto $M\alpha$, il throughput, almeno per M sufficientemente grande è una funzione decrescente di M e per $M \to \infty$ tende a zero. In realtà, nel limite, la catena di Markov non è più ricorrente positiva e dunque tutte le probabilità di trovare un numero finito di buffers occupati è nulla. Dunque per ogni $\beta > 0$ ad ogni slot si ha una collisione sicura.

Introduciamo ora un meccanismo di analisi approssimato che però ha qualche vantaggio. In primo luogo offre comunque un'analisi nel caso in cui M sia troppo elevato per ottenere una soluzione numerica esatta. In secondo luogo, esso mette in luce un comportamento bistabile della cadena che i soli valori medi non riescono a mettere in luce e che è difficilmente osservabile con la sola distribuzione. Infine questo metodo fornisce una sufficiente dimostrazione che il throughput tende a zero quando M diverge.

Il meccanismo che introduciamo è un meccanismo di analisi ai valori medi condizionati. Il numero medio di arrivi durante la permanenza nello stato n è

$$a(n) = (M-n)\alpha \tag{5.11}$$

con media

$$A = E[a(n)] = \sum_n \pi_n a(n) \tag{5.12}$$

Le condizioni di equilibrio della catena di Markov implicano anche che, in condizioni di stazionarietà, si abbia $A = S$, ossia il numero di arrivi medio in uno slot deve eguagliare il numero di partenze medio.

Se $a(n)$ e $s(n)$ fossero non medie ma grandezze deterministiche (es. arrivi e successi periodici) allora gli stati di equilibrio (stazionarietà) del sistema sarebbero quei valori n per i quali si abbia

$$a(n) = s(n) \tag{5.13}$$

Ciò suggerisce un metodo di analisi approssimato che chiameremo "ai valori medi".

Per studiare il comportamento del sistema è comodo esprimere n in funzione di g, ricavato dalla (5.9). Questo permette di esprimere $a(n)$ e $s(n)$ come funzioni di g, $a(g)$ e $s(g)$. Queste curve continuano a esistere solo per valori di G discreti, corrispondenti a valori di n interi, ma per comodità le tracceremo come continue in g. L'espressione di $a(g)$ è

$$a(g) = M\alpha - (g - M\alpha)\frac{\alpha}{\beta - \alpha} \tag{5.14}$$

una retta, ed è riportata per vari valori dei parametri in Figura 5.2 assieme alla $s(g)$. Questa curva, nel limite $M \to \infty$ a $M\alpha$ costante tende alla ben nota (5.2), e il taffico di canale tende a un processo di Poisson e da qui il nome di modello di *popolazione infinita* attributo al semplice modello visto in precedenza. La curva $s(g)$ non cambia che pochissimo nelle zone di interesse dei parametri ($\alpha < \beta$) e si discosta pochissimo dalla (5.2), che infatti è quella disegnata. Si noti che le curve hanno senso solo per $g \geq M\alpha$, poichè tale estremo è il traffico offerto in $n = 0$ e dunque la famiglia delle rette parte da tale punto. Esse intersecano l'asse delle ascisse quando $n = M$ e, dunque, $g = M\beta$.

La condizione (5.13) si traduce allora nella

$$a(g) = s(g) \tag{5.15}$$

Consideriamo il caso della Figura 5.2 con la retta per $M = 100$ e $\alpha = 0.002$, che incontra la curva $s(g)$ in un solo punto, diciamo $g = g_e$. Nei punti $g < g_e$ la deriva $a(g) - s(g)$ è positiva e tende a far crescere g mentre per $g > g_e$ la deriva $a(g) - s(g)$ è negativa e tende a far decrescere g. Ci si aspetta dunque che in condizioni di stazionarietà il sistema oscilli attorno ad g_e. Se l'andamento di $a(g) - s(g)$ è abbastanza simmetrico attorno ad g_e, allora ci si aspetta che l'oscillazione (e dunque la distribuzione stazionaria π_g) sia di conseguenza simmetrica attorno ad g_e con g_e punto di massima probabilità. In queste condizioni, il throughput soddisfa all'approssimazione

$$\sum_g \pi_g s(g) \simeq= s(g_e) \tag{5.16}$$

Dunque g_e assume un significato di ipotetico *punto di funzionamento all' equilibrio* del sistema. Si noti però che questo è solo un concetto approssimativo, in quanto il traffico vero varia continuamente.

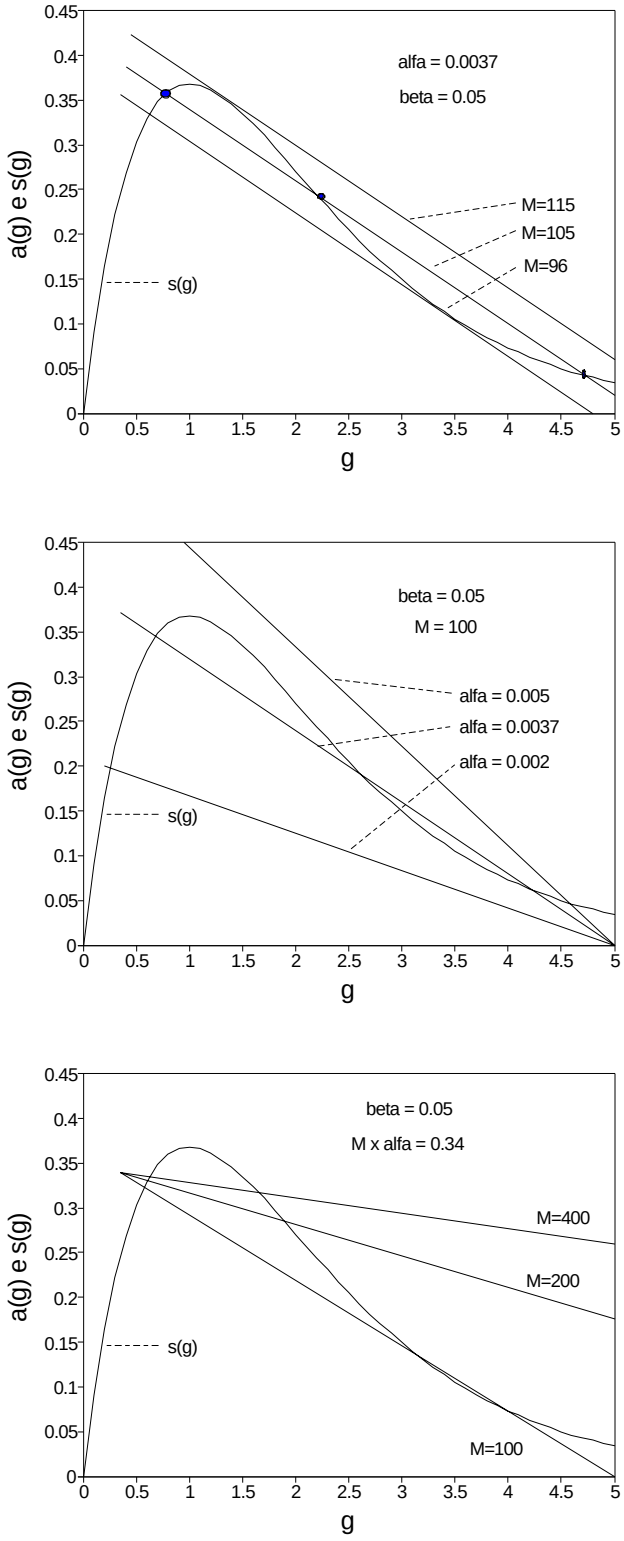

Figura 5.2 *Influenza dei parametri sui punti di equilibrio del throughput per S-ALOHA.*

Se l'andamento di $a(g) - s(g)$ non è più abbastanza simmetrico attorno ad g_e, allora il discorso fatto perde di precisione. Il sistema oscilla, probabilmente in modo meno simmetrico, attorno a valori vicini a g_e e l'appprossimazione (5.16) diventa sempre meno vera.

Al cambiare dei parametri della retta di carico $a(g)$, può succedere che questa intersechi la $s(g)$ in tre punti e la deriva $a(g)-s(g)$ è tale da allontanare il sistema dal punto centrale e da avvicinarlo a quelli estremi. Da ciò si arguisce che la distribuzione stazionaria π_g diventa bimodale attorno a questi due punti. Il sistema tende allora a funzionare in modo *bistabile*, nel senso che prevelentemente opera nell'intorno dei due punti stabili e, di quando in quando, a causa di fluttuazioni statistiche, i punti di funzionamento si scambiano. Il throughput medio è, appunto, una media del throughput nelle due zone, pesata dai rispettivi tempi di permanenza nelle due zone stesse, e si vede che può essere molto più basso del limite visto 0.367. Intuitivamente, la zona di operazione bimodale più probabile è quella più lontana dal punto centrale, detto di equilibrio instabile, perchè più lontano è questo punto e più difficile risulta superarlo e ricadere nell'altra zona.

Al variare dei parametri, la retta $a(g)$ può incrociare $s(g)$ in un solo punto, $M < 96$ in figura a), $\alpha = 0.002$ in figura b), oppure ($M > 115$ in figura a), $\alpha = 0.005$ in figura b), che è di funzionamento stabile. Nei primi due casi esemplificati però il throughput è sensibile, mentre nei secondi due è pressochè nullo.

Particolare rilevanza assume l'esempio in Figura 5.2c dove si vede che, a traffico offerto costante, al crescere di M, la retta diventa orizzontale e il terzo punto tende a $g = \infty, s = 0$. Ciò mostra che quando il sistema si incammina verso il terzo punto, e prima o poi questo succede, tende a procedere verso una situazione di non ritorno. Lo stato $g(t)$ del sistema cresce senza limiti, il che comporta un ritardo crescente senza limiti per i pacchetti arrivati. Si tenga presente che questo è un discorso ancora qualitativo e che una corretta dimostrazione non può che basarsi sulla non ergodicità della catena rappresentante il sistema.

Nelle Figure 5.3 viene mostrata la densità di probabilità stazionaria $pi(n)$ degli stati e il throuhput effettivo a confronto con i punti di equilibrio ottenuti dalle curve $s(n)$ (5.8) e $a(n)$ (5.11). Si noti che $s(n)$ non parte da zero come la $s(g)$ delle curve precedenti, che è approssimata dalla ge^{-g} mentre la vera parte da un valore $g > 0$.

Le figure mostrano una situazione in cui il carico sale e inizia il comportamento bistabile. Nella prima figura, la distribuzione stazionaria pi(n) mostra che il sistema si trova quasi sempre negli stati bassi, con predominanza dello stato zero. Il throughput vero S è più basso di quanto previsto dal punto di equilibrio. Considerazioni analoghe valgono anche per la seconda figura, dove è presente un secondo punto di equilibrio il cui effetto però non si avverte sulla distribuzione. La terza figura mostra una condizione di transizione in cui la distribuzione è bimodale. In pratica il sistema passa il 50% del tempo in zone attorno ai due punti di equilibrio stabile e il throughput è all'incirca la media aritmetica dei due throughput nei due punti di equilibrio. Infine nell'ultima figura il sistema si trova praticamente sempre nel punto di equilibrio a throughput più basso e in questo caso il throughput vero è esattamente predetto dal punto di equilibrio.

Per concludere, da quanto detto appare che il funzionamento del sistema non è soddisfacente. Nel caso di M finito, scegliendo β opportunamente basso, il sistema può essere fatto funzionare in modo stabile, ma in questo caso il ritardo di accesso al sistema è molto maggiore che nel caso TDMA e

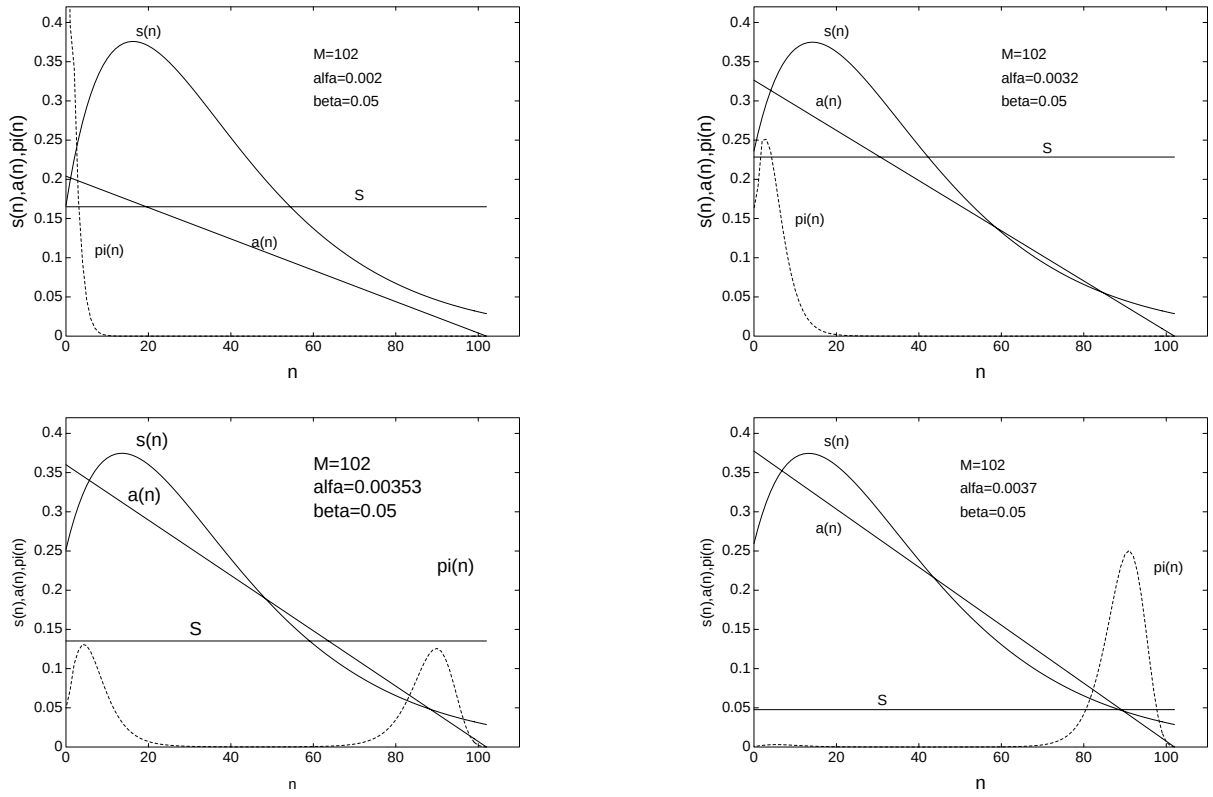

Figura 5.3 *Distribuzione di probabilità stazionaria pi(n) degli stati e il throuhput effettivo a confronto con i punti di equilibrio. La distribuzione non è in scala.*

dunque questa non è una buona soluzione. Per ottenere prestazioni favorevoli occorre tentare di stabilizzare il sistema.

5.3 Protocolli con feedback binario

In questa classe di protocolli si assume che ogni stazione possa osservare il canale e distinguere se ci sia stata una collisione o meno (feedback C, NC). Tale informazione si suppone disponibile non prima della fine dello slot (o della trasmissione) e, per semplicità, la supporremo senz'altro disponibile in tal momento.

5.3.1 Slotted ALOHA stabilizzato

Se, nel sistema descritto nel precedente paragrafo i nuovi arrivi entrassero comunque nei buffers e se si conoscesse il numero n di buffers occupati, da vecchie e nuove trasmissioni, allora la probabilità di trasmissione di tutti i presenti nei buffers, $\beta = \beta(n)$ potrebbe essere scelto in modo dinamico e tale da ottimizzare le prestazioni.

Conoscendo n, la migliore procedura di stabilizzazione è quella che massimizza il throughput ad ogni slot. Il throughput è dato da

$$S = n\beta(1-\beta)^{n-1} = G\left(1-\frac{G}{n}\right)^{n-1}$$

essendo $G = n\beta$. Per n sufficientemente grande ed analogamente al caso non stabilizzato, ma qui più facilmente perchè non c'è α, si ha

$$S \simeq Ge^{-G}.$$

S è in ogni caso massimizzato da $\beta = 1/n$, cioè $G = n\beta = 1, n > 0$.

Naturalmente, far si che tutti conoscano n è impossibile. Il problema è allora quello di ottenere una buona stima $\hat{n}_k$ di n_k, stato del sistema nello slot k e porre ad ogni slot $\beta_k = 1/\hat{n}_k$.

Purtroppo, non esiste alcun modo di ottenere una buona stima $\hat{n}$ basata sulla sola conoscenza del successo o meno della propria trasmissione. Occorre ipotizzare che le stazioni possano ottenere maggiore informazione sull'attività del canale.

Una metodo di stima, detto pseudo Bayesiano, è quello basato sul feedback binario (C, NC) e opera come segue.

Si suppone di effettuare ad ogni slot k una stima $\hat{n}_k$ delnumero di buffers occupati n_k e si assume che l'errore di stima sia tale da far si che il vero valore sia una variabile casuale $N = n_k$ distribuita secondo Poisson con valor medio $\hat{n}_k$, ossia si abbia

$$P(N = i) = \frac{\hat{n}_k^i}{i!} e^{-\hat{n}_k}.$$

L'assunzione è congruente col fatto che la miglior stima a priori di una variabile di Poisson , ossia quel valore che ha la probabilità più alta di verificarsi a priori, coincide con il valor medio.

Avendo a disposizione l'osservazione del canale, la stima, che diventa a posteriori, si modifica. Per esempio, se si osserva una trasmissione non collisa, la stima nel successivo slot va modificata come:

$$\hat{n}_{k+1} = \hat{n}_k + S - 1 \tag{5.17}$$

essendo S il numero medio di arrivi nello slot. Se si osserva uno slot vuoto, la stima nel successivo slot è quella che massimizza la probabilità a posteriori

$$P(N = i/\text{slot vuoto}) = \frac{P(N = i; \text{slot vuoto})}{P(\text{slot vuoto})} = \frac{(1-\beta)^i \frac{\hat{n}_k^i}{i!} e^{-\hat{n}_k}}{e^{-1}}$$

dove la probabilità incondizionata a denominatore (e^{-G}) deriva dal fatto che l'algoritmo tiene il sistema in $G = 1$. Ricordando che l'algoritmo impone $\beta = 1/n_k$ e sostituendo nell'espressione trovata si ha

$$P(N = i/\text{slot vuoto}) = \frac{(\hat{n}_k - 1)^i}{i!} e^{-(\hat{n}_k - 1)}. \tag{5.18}$$

La distribuzione a posteriori risulta Poisson con media $\hat{n}_k - 1$ e dunque la stima diventa

$$\hat{n}_{k+1} = \hat{n}_k - 1 + S \tag{5.19}$$

dove, ancora, S è stato aggiunto per tener conto degli arrivi. Le (5.17) e (5.19) sono uguali e si vede dunque che non serve distinguere fra slot vuoto e slot con corretta trasmissione.

Nel caso di collisione, la probabilità a posteriori corrispondente alla (5.18) è data da:

$$P(N = i/\text{collisione}) = \frac{\left[1 - \left(1 - \frac{1}{\hat{n}_k}\right)^i - i\frac{1}{\hat{n}_k}\left(1 - \frac{1}{\hat{n}_k}\right)^{i-1}\right] \frac{\hat{n}_k^i}{i!} e^{-\hat{n}_k}}{e^{-1}} \tag{5.20}$$

Per trovare la migliore stima occorre trovare il massimo della distribuzione sopra. Alternativamente, l'osservazione media,

$$P(\text{slot vuoto}) + P(\text{trasmissione}) + P(\text{collisione}) = 1$$

che corrisponde a informazione nulla, deve fornire variazione nulla e dunque se i nuovi arrivi sono in numero di S anche le partenze (la riduzione di $\hat{n}_k$) deve essere pari a S. Ma, sempre in condizioni ottimali (G=1) è $S \simeq 1/e$. La variazione x al valor medio m apportata dall' osservare una collisione deve essere dunque tale che

$$P(\text{slot vuoto})(-1) + P(\text{trasmissione})(-1) + P(\text{collisione})x = -1/e$$

e dunque dev'essere

$$-\frac{2}{e} + x\frac{e-2}{e} = -\frac{1}{e}$$

e fornisce

$$x = \frac{1}{e-2}$$

e dunque in questo caso si pone

$$\hat{n}_{k+1} = m + \frac{1}{e-2} + S = \hat{n}_k + \frac{1}{e-2} + S \tag{5.21}$$

L'algoritmo finale risulta essere:

$$\hat{n}_{k+1} = \begin{cases} \max[S, \hat{n}_k + S - 1] & \text{nel caso } NC \\ \hat{n}_k + S + (e-2)^{-1} & \text{nel caso } C \end{cases} \tag{5.22}$$

L'algoritmo illustrato permette un funzionamento stabile per tutti i valori di $S < e^{-1}$. Naturalmente, occorre conoscere anche S. Tuttavia l'algoritmo rimane comunque stabile se in esso si assume come S il valore e^{-1}.

5.3.2 Algoritmi ad albero [Capetanakis, 1977]

Nelle tecniche illustrate di seguito, viene introdotto per la prima volta il concetto di *Algoritmo per la soluzione delle collisioni*, o CRA (Collision Resolution Algorithm), in cui le stazioni usano un

metodo razionale per risolvere le loro contese. Infatti, il feedback di canale fa conoscere quando iniziano delle contese e da questo istante parte il CRA. Durante il CRA possono giungere alle stazioni nuovi pacchetti. I protocolli che bloccano i nuovi pacchetti fino a che il CRA è in corso sono detti del tipo BLOCKED ACCESS (BA), altrimenti se anche i pacchetti nuovi possono partecipare al CRA il protocollo è detto del tipo FREE ACCESS.

L'idea nell'algoritmo di Capetanakis consiste nel suddividere il gruppo di stazioni contendenti ad ogni collisione, e di applicarsi alla soluzione delle contese in un sottogruppo prima di passare agli altri. Quando in tutti gli slot dedicati alla trasmissione di ciascun sottogruppo non si ha collisione, allora la collisione del gruppo originale è stata risolta. La successiva applicazione di questo principio ai sottogruppi porta alla soluzione di un numero finito di collisioni in un tempo medio finito e con caratteristiche tali da garantire un throughput stabile.

La suddivisione in sottogruppi, può operare in diversi modi ma deve far uso di un esperimento casuale se l'algoritmo deve poter operare con un numero infinito di stazioni. L'algoritmo originale base suddivide in due sottogruppi in base al risultato del lancio di una moneta onesta, modo per riferirsi a un esperimento casuale binario con il 50% di probabilità di successo.

Esso è del tipo BLOCKED ACCESS, riferendosi al fatto che la prima collisione determina la partenza del periodo di soluzione delle collisioni (CRP), durante il quale, le stazioni non coinvolte nella collisione originale non possono trasmettere. L'algoritmo prosegue ricorsivamente nel seguente modo:

dopo ogni collisione, le stazioni coinvolte lanciano una moneta. Quelle che ottengono 1 trasmettono nello slot immediatamente successivo. Quelle che ottengono zero, nel primo slot dopo che tutte le collisioni fra quelle che hanno ottenuto 1 sono state risolte.

Tale algoritmo viene detto *ad albero* per il modo con cui possono essere rappresentate le varie collisioni e il percorso di soluzione.

In Figura 5.4 è mostrato un esempio.

L'algoritmo può essere studiato considerando il processo $U(k)$, k intero, rappresentante il numero di trasmissioni all'inizio di ciascun CRP. Il valore istantaneo di $U(k+1)$ rappresenta il numero di arrivi durante il CRP precedente e può essere determinato conoscendo la distribuzione della lunghezza $l(k)$ di tale periodo. D'altra parte, tale distribuzione dipende unicamente da $U(k)$ e dall'algoritmo scelto. Dunque $U(k+1)$ dipende unicamente da $U(k)$ ed è pertanto una catena di Markov. Si noti che i punti di inizio del CRP non costituiscono eventi di rinnovamento perchè $l(k+1)$ è legata a l_k e dunque il processo originale non è rigenerativo. In condizioni di stazionarietà ed ergodicità, il throughput può essere espresso, come per i processi rigenerativi, come

$$S = \frac{E[U]}{E[l]} \tag{5.23}$$

Il throughput massimo è il massimo dell'espressione sopra, al variare del traffico offerto. In particolare, può esistere toughput finito anche se il numeratore e il denominatore della (5.23) divergono.

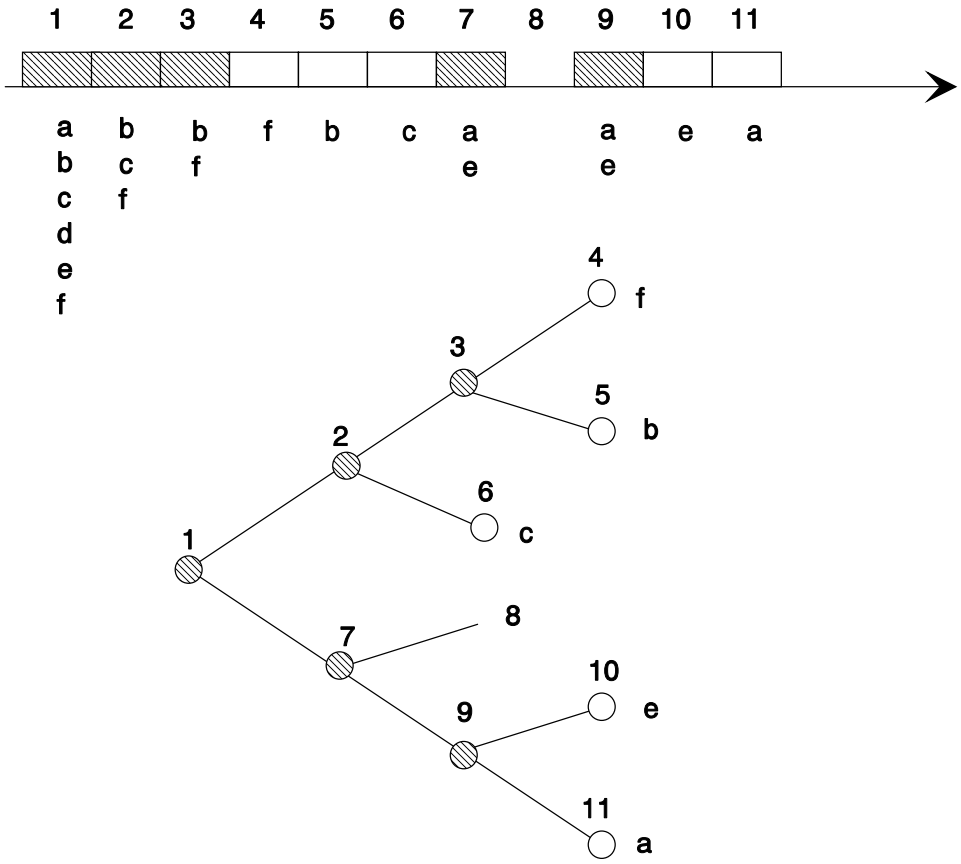

Figura 5.4 *Esempio di funzionamento dell'algoritmo di Capetanakis. In alto, la successione di slots con le stazioni che trasmettono. Sotto, l'albero che ciascuna stazione può tracciare. Ogni nodo intermedio rappresenta uno slot con collisione, mentre i nodi terminali rappresentano slots con non collisioni. La corrispondenza con gli slots sopra è indicata dalla numerazione. La suddivisione verso l'alto porta a slots in cui trasmettono coloro che hanno sorteggiato* 0*, e viceversa quelli verso il basso.*

Purtroppo, l'analisi esatta richiede, come detto, la conoscenza della distribuzione della lunghezza $l(k)$, non facile da ottenere. Tuttavia, il massimo throughput smaltibile dal sistema si può valutare più semplicemente. Infatti posto $L_n = E[l/n]$, il numero medio di slots necessario a risolvere n collisioni iniziali, è

$$S_{\max} = \lim_{n\to\infty} \frac{n}{L_n} = S_\infty \tag{5.24}$$

Se infatti fosse possibile che $S_{\max} > S_\infty$, ossia se fosse possibile operare in condizioni stazionarie con in ingresso $S_{\max}$ con $E[n] < \infty$, allora le oscillazioni statistiche farebbero prima o poi crescere n e, se S varia con continuità, farebbero scendere S verso il valore S_∞, determinando uno squilibrio fra ingresso e uscita e dunque una crescita $E[n] \to \infty$, contraddicendo l'ipotesi.

Data la ricorsività dell'algoritmo, si ha:

$$L_n = 1 + \sum_{i=0}^{n} \binom{n}{i} 2^{-n}(L_i + L_{n-i}) \tag{5.25}$$

Partendo da $L_0 = L_1 = 1$, la relazione sopra permette di ricavare $L_2 = 5$ e via via tutti gli altri valori. Si può poi mostrare che la soluzione è

$$L_n = 1 + \sum_{i=2}^{n} \binom{n}{i} (-1)^i \frac{2(i-1)}{1-2^{-i+1}}. \tag{5.26}$$

Si trova che la grandezza

$$\frac{L_n}{n} \triangleq \gamma(n)$$

non ammette limite perchè oscilla al crescere di n. Si può però scrivere

$$\gamma(n) = \gamma_0 + g(n) + O(n^{-1})$$

dove $O(n^{-1})$ è una quantità che tende a zero mentre $g(n)$ è una funzione oscillante entro limitati valori (più piccoli di 10^{-3}), periodica in $\log_2(n)$ e

$$\gamma_0 = \frac{2}{\ln 2} \approx 2.8854 \tag{5.27}$$

In pratica si riesce anche a valutare l'ampiezza dell'oscillazione massima che è contenuta entro 10^{-6}. Si ottiene dunque

$$S_{\max} \approx \frac{\ln 2}{2} = 0.346574 \tag{5.28}$$

Un algoritmo come quello proposto viene anche chiamato *stack algorithm*, perchè il suo funzionamento può essere realizzato anche con una pila in cui sono impilate le stazioni che hanno ottenuto zero al lancio della moneta.

5.3.3 Varianti all'algoritmo CRA base

Naturalmente, nasce subito il problema di vedere se la suddivisione binaria equivalente è la migliore delle suddivisioni possibili. Indicando con r il numero di suddivisioni e con $p_j, j = 1, ..., r, \sum_j p_j = 1$, la probabilità con cui viene scelta la classe j, si riesce ad esprimere le equivalenti delle (5.25), (5.26), e (5.27) come:

$$L_n = 1 + \sum_{j=1}^{r} \left\{ \sum_{l=1}^{n} \binom{n}{l} p_j^l (1-p_j)^{n-l} L_l \right\} \qquad n \geq 2 \tag{5.29}$$

$$L_n = 1 + \sum_{i=2}^{n} \binom{n}{i} (-1)^i \frac{r(i-1)}{1 - \sum_{j=1}^{r} p_j^i}. \tag{5.30}$$

$$\gamma_0 = \frac{r}{-\sum_j p_j \ln p_j} \tag{5.31}$$

Anche qui le oscillazioni sono molto contenute, anche se crescono con r (per $r = 4$ sono ancora minori di 10^{-3}). Si può dunque assumere $1/\gamma_0$ come approssimazione del throughput massimo. La (5.31) è simmetrica nei valori p_j e si vede che la scelta più conveniente è quella uniforme, $p_j = 1/r$. Il il valore più conveniente di r è 3 con un throughput massimo di $S_{\max} = 0.3662$.

Poichè però il numero di collisioni che ci si aspetta di risolvere ad ogni CRP cambia con la durata del precedente CRP, appare chiaro come il numero di suddivisioni ottimale possa cambiare da un CRP ad un altro. Capetanakis ha indicato che la migliore soluzione, per quanto riguarda il numero di suddivisioni equivalenti, è di aumentare tale numero dinamicamente al crescere del numero medio di arrivi nel (e quindi con la durata del) precedente CRP. Applicando tale procedura solo alla prima suddivisione ha mostrato che si ottiene un throughput massimo pari a 0.43. Ovviamente però una tale ottimizzazione richiede che sia noto il valor medio del traffico generato, cosa non realisticamente possibile.

5.3.4 Algoritmo base con modifica FREE ACCESS

Nei protocolli BA, perchè ogni stazione possa correttamente operare, occorre che la storia del canale sia completamente nota. I protocolli FA rilasciano la necessità di questa assunzione, supponendo che i pacchetti nuovi vengano comunque trasmessi subito. Il CRA viene poi applicato nel caso la trasmissione risulti in collisione, seguendo la storia del canale da quel punto.

Si noti che in questo caso gli istanti in cui tutte le collisioni sono state risolte sono istanti di rinnovo e pertanto il processo in gioco è rigenerativo e si ha

$$S = \frac{E[A]}{E[l]} \tag{5.32}$$

dove $E[A]$ numero medio di arrivi durante l (ciclo di rigenerazione non dipende più dal passato ma solo dal futuro, e così $E[l]$. Purtroppo ricavare i due valori medi non è semplice. C'è però un'altra strada.

Consideriamo subito il caso generale di suddivisioni multiple con probabilità non uniformi. Posto L_n la lunghezza del CRP che incomincia con n collisioni, si trova

$$L_n = 1 + \sum_{j=1}^{r} \left\{ \sum_{l=1}^{n} \left[\binom{n}{l} p_j^l (1-p_j)^{n-l} \sum_{k=0}^{\infty} L_{l+k} \frac{S^k}{k!} e^{-S} \right] \right\} \tag{5.33}$$

Purtroppo la (5.33) non è ricorsiva e può essere risolta solo con complesse procedure nelle trasformate. A differenza del caso BA, la (5.33) dipende esplicitamente anche dal traffico offerto e potrebbe non esistere se il traffico offerto fosse troppo alto. Infatti in questo caso la soluzione delle collisioni non riesce a tener dietro ai nuovi arrivi e collisioni con accumulo delle collisioni irrisolte per cui il ciclo non finisce più (il processo rigenerativo diventa "non ricorrente") e il throughput tende a zero. Dunque il massimo throughput del sistema è determinato dal massimo S per il quale esistono tutte le (5.33). Questo, nel caso di suddivisioni equiprobabili, si dimostra essere determinato dall'equazione

$$\frac{1}{1-\mu(S)} e^{-\mu(S)} \sum_{m=0}^{\infty} r^m \left\{ \left[1 - \mu(S) r^{-m} + \mu(S)^2 r^{-2m} (1 - r^{-1}) \right] \times \right.$$

$$\left. \times e^{\mu(S) r^{-m}} - (1 - \mu(S) r^{-m-1}) e^{\mu(S) r^{-m-1}} \right\} = \frac{1}{r} \tag{5.34}$$

dove

$$\mu(S) = \frac{rS}{r-1}$$

e $S_{\max}$ deve soddisfare il vincolo

$$S < \frac{r-1}{r}.$$

Si trova che, con suddivisioni uguali, per $r = 2$ si ha $S_{\max} = 0.360177$ mentre per $r = 3$ si ha $S_{\max} = 0.401599$, che costituisce il massimo valore su ogni r. Valutazioni con suddivisioni non uniformi non permettono di migliorare i risultati.

Si noti che questo protocollo funziona meglio che il corrispondente col BA.

5.3.5 CRA a numero medio di pacchetti costante [Massey]

Si è visto che il processo di ottimizzazione del precedente algoritmo si basa sulla conoscenza della frequenza media λ degli arrivi. Una procedura ancora migliore è allora quella di processare ad ogni CRP un numero medio di pacchetti non superiore a una data costante da ottimizzare.

Al fine di spiegare il concetto, supponiamo di operare conoscendo gli istanti di arrivo passati e anche futuri, $t_1, t_2, ...t_n, ...$ L'asse dei tempi viene suddiviso in *epoche* di lunghezza costante ξ e, di volta in volta, vengono trasmessi dall'algoritmo ad ogni CRP i soli pacchetti arrivati durante una sola epoca. Poichè questi sono in numero medio costante $\gamma = \lambda\xi$, la distribuzione della lunghezza di un CRP è sempre la stessa e gli istanti di inizio di un CRP sono *istanti di rigenerazione*.

Il throughput S di un tale processo trasmissivo può essere espresso in base ai teoremi sui processi rigenerativi:

$$S_{\max} = \frac{\gamma}{L} \tag{5.35}$$

In un processo di trasmissione *in tempo reale*, gli istanti di arrivo t_i che si conoscono sono solo quelli occorsi prima dell'istante di tempo t corrente. Se i pacchetti di un' epoca che si conclude in t_e sono stati tutti trasmessi con successo, una nuova epoca può essere definita come l'intervallo $[t_e, t_e + \xi]$ solo se $t_e + \xi \leq t$. altrimenti la nuova epoca è costituita dall'intervallo $[t_e, t]$, la cui durata è minore di ξ e dunque con un numero medio di pacchetti minore di γ. In un tale processo i CRP che si susseguono non costituiscono più cicli di rigenerazione. Tuttavia, dal punto di vista del massimo throughput, ciò che conta è come si comporta il sistema con una coda infinita, che coincide con il caso del processo in tempo differito.

La valutazione di L può essere effettuata sulla base di quanto già visto nei precedenti casi. Infatti L può essere espressa come

$$L = \sum_{n=0}^{\infty} a_n L_n = \sum_{n=0}^{\infty} \frac{\gamma^n}{n!} e^{-\gamma} L_n \tag{5.36}$$

dove a_n è la distribuzione degli arrivi in un epoca, esplicitata nella seconda parte con una distribuzione di Poisson. Naturalmente l'espressione di L dipende dal CRA usato e si può usare l'espressione generale (5.30). Purtroppo, non è possibile trovare un'espressione in forma chiusa della (5.36) nel caso generale. Inoltre essa mal si presta a una valutazione numerica.

Il seguente metodo si presta a una facile valutazione numerica in forma ricorsiva nel caso di suddivisioni equivalenti. Prima però è utile osservare che una suddivisione in gruppi, può essere anche fatta suddividendo l'epoca considerata in intervalli proporzionali alla probabilità dei gruppi stessi. Per esempio, una suddivisione binaria al 50% può essere ottenuta dividendo in due intevalli uguali l'epoca. Le stazioni con pacchetti arrivati nel primo intervallo appartengono al primo gruppo e sono quelle che trasmettono subito dopo la prima collisione. Una procedura di questo tipo è del tutto equivalente al lancio della moneta, ma permette di vedere l'algoritmo come una diretta esplorazione dell'asse dei tempi anzichè dell'albero delle collisioni. Inoltre, l'uso della suddivisione in intervalli ha altri due vantaggi. Primo, riesce a trasmettere i pacchetti secondo la procedura Primo Arrivato Primo Servito (FCFS). Secondo, permette di constatare come i pacchetti che corrispondono a biforcazioni nell'albero sono ancora distribuiti secondo Poisson, ovviamente con la condizione osservata durante lo slot.

Definiamo con l'indice j la profondità dell'albero (o dello stack) a partire dal livello 0 rappresentante il nodo radice. Sia poi

D_j il numero medio di slot necessari a risolvere le collisioni presenti in un nodo di livello j posto che nell'intervallo ci sia stata una collisione;

X_j il numero di sottointervalli dell'intervallo di livello j che contengono almeno due pacchetti, posto che nell'intervallo ci sia stata una collisione;

E' facile vedere che

$$D_j = r + \sum_{i=0}^{r} i D_{j+1} P(X_j = i) \tag{5.37}$$

Come già visto, in un intervallo di livello j sono presenti dei pacchetti il cui numero è distribuito secondo Poisson con valor medio γ_j, con

$$\gamma_j = \frac{\gamma}{r^j}.$$

Dunque si ha

$$P(X_j = i) = \begin{cases} \binom{r}{i} \dfrac{(e^{-\gamma_{j+1}} + \gamma_{j+1}e^{-\gamma_{j+1}})^{r-i}(1 - e^{-\gamma_{j+1}} - \gamma_{j+1}e^{-\gamma_{j+1}})^{i}}{1 - e^{-\gamma_j} - \gamma_j e^{-\gamma_j}} & 1 \le i \le r \\ \displaystyle\sum_{k=2}^{r} \binom{r}{k} \frac{e^{-\gamma_{j+1}(r-k)}(\gamma_{j+1}e^{-\gamma_{j+1}})^{k}}{1 - e^{-\gamma_j} - \gamma_j e^{-\gamma_j}} & i = 0 \end{cases} \tag{5.38}$$

La relazione sopra è, per $i > 0$ una binomiale fra gli eventi: *almeno due pacchetti nel sottointervallo* e il suo complementare; è però condizionata dal fatto che c' è stata una collisione, e dunque ci sono almeno due pacchetti nell'intervallo. Per $i = 0$, cade l'indipendenza fra i pacchetti nei sottointervalli perchè, stabilito che non ce ne sono con due o più pacchetti, allora, per la condizione sopra citata, il numero di sottointervalli con un pacchetto non può essere minore di due.

Occorre notare che se la profondità del livello è sufficientemente elevata, allora se si ha una collisione è praticamente certo che il numero di pacchetti coinvolti è due. La (5.37) diventa:

$$D_j = r + \frac{1}{r} D_{j+1}$$

e, nel limite, si ha:

$$\lim_{j \to \infty} D_j = \frac{r^2}{r - 1} \tag{5.39}$$

Il valore sopra può essere usato come valore di partenza a ritroso per calcolare ricursivamente le (5.37) fino a D_0. Il massimo throughput è allora espresso da

$$S_{\max} = \frac{\gamma}{1 + (1 - e^{-\gamma} - \gamma e^{-\gamma}) D_0} \tag{5.40}$$

Si trova che la suddivisione migliore è per $r = 2$. In questo caso il miglior γ è $\gamma_o = 1.148$ che fornisce un $S_{\max} = 0.4295$. Non si ottiene risultato migliore nemmeno con suddivisioni diverse fra i livelli. Dunque il valore di ξ che permette di smaltire il traffico massimo è $\xi_o = 1.148/0.4295 = 2.673$ slots.

Vediamo ora il caso di suddivisione binaria non uniforme. In questo caso le suddivisioni ai vari livelli sono più complicate. L'indice di livello j è sostituito dall'indice binario i, j che denota i suddivisioni di proporzione p_1 e j suddivisioni di proporzione p_2. Tale intervallo si trova a livello $i + j$.

$$\gamma_{ij} = \gamma p_1^i p_2^j.$$

Introduciamo ancora la variabile:

$Z_{ji}(k), k = 1, 2$ il numero medio di pacchetti presenti nel sottointervallo di indice k dell'intervallo di livello ij.

Abbiamo:

$$D_{ij} = 2 + D_{i+1,j} P(Z_{ij}(1) \geq 2) + D_{i,j+1} P(Z_{ij}(2) \geq 2) \tag{5.41}$$

con

$$\begin{cases} P(Z_{ij}(1) \geq 2) = \dfrac{(1 - e^{-p_1 \gamma_{ij}} - p_1 \gamma_{ij} e^{-p_1 \gamma_{ij}})}{1 - e^{-\gamma_{ij}} - \gamma_{ij} e^{-\gamma_{ij}}} \\ P(Z_{ij}(2) \geq 2) = \dfrac{(1 - e^{-p_2 \gamma_{ij}} - p_2 \gamma_{ij} e^{-p_2 \gamma_{ij}})}{1 - e^{-\gamma_{ij}} - \gamma_{ij} e^{-\gamma_{ij}}} \end{cases} \tag{5.42}$$

Inoltre quando uno dei due indici è sufficientemente grande si ha

$$D_{ij} = 2 + p_1^2 D_{i+1,j} + p_2^2 D_{i,j+1}$$

che fornisce

$$\lim_{(i+j) \to \infty} D_{ij} = \frac{2}{1 - p_1^2 - p_2^2} \tag{5.43}$$

Si noti che la (5.43) è minimizzata per $p_1 = p_2 = 0.5$. Assumendo $D_{i,j+1} = D_{i+1,i}$, caso uniforme, nella (5.41), si vede che ancora la scelta più conveniente è l'uniforme a tutti i livelli e il valore più sopra trovato costituisce il massimo anche sulle suddivisioni non uniformi.

5.3.6 CRA a numero medio di pacchetti ottimale [Gallager-Tsybakov]

Questo algoritmo è una variante del precedente che consiste nell'arrestare l'esplorazione dell'asse dei tempi quando non si hanno più informazioni (o meglio, si ha solo l'informazione a priori) riguardo al numero di pacchetti ivi presenti. Questo capita quando, essendoci stata una collisione in un intervallo, che implica almeno la presenza di almeno due pacchetti nell'intervallo stesso, nella successiva esplorazione dei (due o più) sottointervalli che ne derivano, si trovano almeno due pacchetti prima di aver esplorato tutti i sottointervalli stessi. Specificamente, ci si ferma dopo due successi, oppure dopo un successo e una collisione oppure dopo nessun successo e una collisione. Allora l'informazione relativa al numero di pacchetti presenti nei restanti sottointervalli dello stesso livello è solo quella a priori. Risulta in questo caso conveniente rimandare l'esplorazione di tale

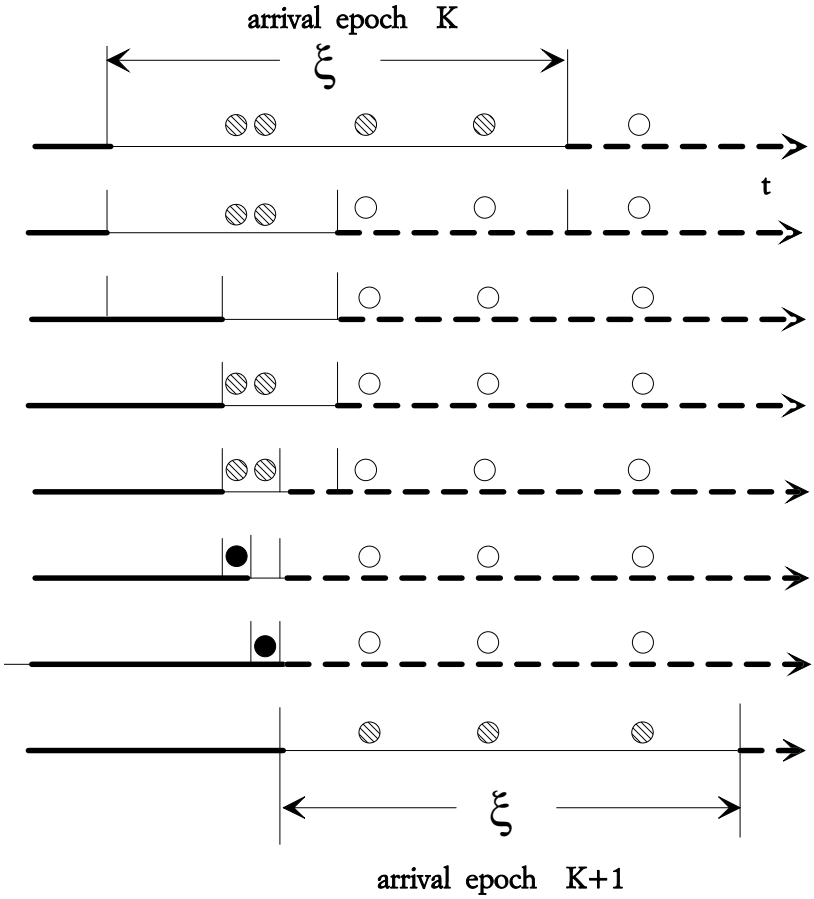

Figura 5.5 *Esempio di funzionamento dell'algoritmo Gallager-Tsybakov. In grassetto è la zona esplorata e in grassetto tratteggiato la zona ancora da esplorare.*

parte al successivo CRP. Se l'esplorazione dei sottointervalli avviene in senso crescente del tempo, arrestare l'esplorazione porta ad un epoca che è ancora un intervallo benchè più corto del valore ξ da cui si è partiti. La parte non esplorata viene conglobata nella successiva epoca che viene ancora presa di lunghezza ξ. In Figura 5.5 è riportato l'esempio di soluzione delle collisioni fra i pacchetti presenti nel periodo iniziale (epoca) K. In questa figura si vede che inizialmente l'asse dei tempi è suddiviso in tre zone: una zona già esplorata a sinistra, la K-esima epoca da esplorare, e la zona da esplorare in futuro a destra. Al procedere dell'algoritmo, il confine della zona esplorata si sposta verso destra mentre quello della zona da esplorare si sposta verso sinistra finchè non si congiungono alla fine del CRP, quando una nuova epoca viene scelta.

Con questa variante, il numero medio di pacchetti trasmesso ad ogni CRP non è più $\gamma = \lambda\xi$ ma va calcolato espressamente, come il numero di slot. Accanto alle definizioni già viste aggiungiamo:

N_j il numero medio di pacchetti che vengono trasmessi con successo a partire da un nodo di livello j posto che nell'intervallo ci sia stata una collisione;

F_j^{nc} il numero d'ordine del sottointervallo in cui si arresta l'esplorazione, senza che ci sia stata collisione;

F_j^c il numero d'ordine del sottointervallo in cui si arresta l'esplorazione, essendoci stata in quel sottointervallo una collisione;

$Y_j^{nc}(i)$ il numero medio di pacchetti trasmessi con successo quando si arresta l'esplorazione al sottointervallo i a livello j senza collisioni;

$Y_j^c(i)$ il numero medio di pacchetti trasmessi con successo quando si arresta l'esplorazione al sottointervallo i a livello j con una collisione;

Il massimo throughput è ancora espresso dalla (5.35) dove γ va sostituito dal valor medio dei messaggi effettivamente trasmessi.

$$S_{\max} = \frac{\gamma e^{-\gamma} + (1 - e^{-\gamma} - \gamma e^{-\gamma}) N_0}{1 + (1 - e^{-\gamma} - \gamma e^{-\gamma}) D_0} \tag{5.44}$$

Per il calcolo di N_0 e D_0, si vede facilmente che valgono le seguenti ricorsioni:

$$D_j = \sum_{i=1}^{r} [iP(F_j^{nc} = i) + (i + D_{j+1}) P(F_j^c = i)] \tag{5.45}$$

$$N_j = \sum_{i=1}^{r} [Y_j^{nc}(i) P(F_j^{nc} = i) + (Y_j^c(i) + N_{j+1}) P(F_j^c = i)] \tag{5.46}$$

Si ha

$$P(F_j^{nc} = i) = \frac{\gamma_{j+1}(i-1) e^{-\gamma_{j+1}(i-1)} \gamma_{j+1} e^{-\gamma_{j+1}}}{1 - e^{-\gamma_j} - \gamma_j e^{-\gamma_j}} \qquad i \geq 2 \tag{5.47}$$

$$P(F_j^c = i) = \frac{(e^{-\gamma_{j+1}(i-1)} + \gamma_{j+1}(i-1) e^{-\gamma_{j+1}(i-1)})(1 - e^{-\gamma_{j+1}} - \gamma_{j+1} e^{-\gamma_{j+1}})}{1 - e^{-\gamma_j} - \gamma_j e^{-\gamma_j}} \tag{5.48}$$

$$Y_j^{nc}(i) = 2 \qquad i \geq 2 \tag{5.49}$$

$$Y_j^c(i) = \frac{\gamma_{j+1}(i-1) e^{-\gamma_{j+1}(i-1)}}{e^{-\gamma_{j+1}(i-1)} + \gamma_{j+1}(i-1) e^{-\gamma_{j+1}(i-1)}} \tag{5.50}$$

La relazione (5.47) si basa sul fatto che ci deve essere un solo pacchetto nei primi $i-1$ sottointervalli e uno solo nel sottointervallo i. Per la relazione (5.48) non ci deve essere più di un pacchetto nei primi $i-1$ sottointervalli e almeno due nel sottointervallo i. La (5.49) si spiega da sola. La (5.50) è il numero medio di pacchetti che si trova nei primi $i-1$ sottointervalli (mai più di uno).

Ancora, se la profondità del livello è sufficientemente elevata, allora se si ha una collisione è praticamente certo che il numero di pacchetti coinvolti è due. La (5.45) diventa:

$$D_j = \frac{1}{r}\left(\frac{r+1}{2} + D_{j+1}\right) + \frac{r-1}{r}\,\frac{2}{3}(r+1) \tag{5.51}$$

Dunque si ha:

$$\lim_{j\to\infty} D_j = \frac{r+1}{2(r-1)} + \frac{2}{3}(r+1) \tag{5.52}$$

$$\lim_{j\to\infty} N_j = 2 \tag{5.53}$$

I valori sopra possono essere usati come valore di partenza a ritroso per calcolare ricursivamente le (5.45) e (5.46). Si trova che la suddivisione migliore è per $r = 3$. In questo caso il miglior γ è $\gamma_o = 1.23$ che fornisce un $S_{\max} = 0.4496$. Il caso $r = 2$ è però molto vicino con $S_{\max} = 0.4493$ e $\gamma_o = 1.16$. Se si indagano suddivisioni variabili fra i livelli e si pone $r_0 = 3$ e $r_i = 2, i \geq 1$, allora si trova $S_{\max} = 0.4532$ con $\gamma_o = 1.23$.

Vediamo anche qui il caso di suddivisione binaria non uniforme, come nel paragrafo precedente. Le (5.45) (5.46) diventano

$$D_{ij} = (1 + D_{i+1,j})P(F_{ij}^c = 1) + 2P(F_{ij}^{nc} = 2) + (2 + D_{i,j+1})P(F_{ij}^c = 2) \tag{5.54}$$

$$N_{ij} = N_{i+1,j}P(F_{ij}^c = 1) + 2P(F_{ij}^{nc} = 2) + (Y_{ij}^c(2) + N_{i,j+1})P(F_{ij}^c = 2) \tag{5.55}$$

con

$$P(F_{ij}^{nc} = 2) = \frac{\gamma_{ij}p_1 e^{-\gamma_{ij}p_1}\gamma_{ij}p_2 e^{-\gamma_{ij}p_2}}{1 - e^{-\gamma_{ij}} - \gamma_{ij}e^{-\gamma_{ij}}} \tag{5.56}$$

$$P(F_{ij}^c = 1) = \frac{1 - e^{-\gamma_{ij}p_1} - \gamma_{ij}p_1 e^{-\gamma_{ij}p_1}}{1 - e^{-\gamma_{ij}} - \gamma_{ij}e^{-\gamma_{ij}}} \tag{5.57}$$

$$P(F_{ij}^c = 2) = \frac{(e^{-\gamma_{ij}p_1} + \gamma_{ij}p_1 e^{-\gamma_{ij}p_1})(1 - e^{-\gamma_{ij}p_2} - \gamma_{ij}p_2 e^{-\gamma_{ij}p_2})}{1 - e^{-\gamma_{ij}} - \gamma_{ij}e^{-\gamma_{ij}}} \tag{5.58}$$

$$Y_{ij}^{nc}(2) = 2 \tag{5.59}$$

$$Y_{ij}^{c}(2) = \frac{\gamma_{ij} p_1 e^{-\gamma_{ij} p_1}}{e^{-\gamma_{ij} p_1} + \gamma_{ij} p_1 e^{-\gamma_{ij} p_1}} \tag{5.60}$$

e $Y_{ij}^{nc}(1) = Y_{ij}^{c}(1) = 0$.

Quando uno dei due indici è sufficientemente grande si ha

$$D_{ij} = [1 + D_{i+1,j}]p_1^2 + 4p_1 p_2 + [2 + D_{i,j+1}]p_2^2$$

che fornisce

$$\lim_{(i+j)\to\infty} D_{ij} = \frac{1 + 2p_1 p_2 + p_2^2}{1 - p_1^2 - p_2^2} \tag{5.61}$$

La (5.55) è ancora minimizzata per $p_1 = p_2 = 0.5$. Tuttavia non si può fare un ragionamento come nel caso precedente perchè questa volta, nell'ottimizzazione è coinvolto anche il fattore N_{jk}

5.4 Protocolli con feedback ternario

In questa classe di protocolli si assume che ogni stazione possa osservare il canale e distinguere se ci sia stata un' assenza di trasmissione, una trasmissione corretta oppure una collisione (feedback 0,1,C). Anche in questo caso supporremo che tale informazione sia disponibile alla fine dello slot.

Tutti i protocolli ad albero visti nel caso con feedback binario, possono essere estesi sfruttando le potenzialità del caso ternario. Il CRA parte e procede allo stesso modo salvo quando, nei primi $r-1$ slots di una certa ripartizione non trasmette nessuno. In questo caso si ha la sicurezza che nel successivo slot si avrà una collisione. E' dunque inutile operare una trasmissione in queste condizioni. Le stazioni operano come se questa ci fosse stata ed operano una successiva ripartizione.

5.4.1 Il protocollo base BA

Le equazioni corrispondenti alle (5.29), (5.30) e (5.31) sono:

$$L_n = 1 + \sum_{j=1}^{r} \left\{ \sum_{l=1}^{n} \binom{n}{l} p_j^l (1-p_j)^{n-l} L_l \right\} - p_r^n \qquad n \geq 2 \tag{5.62}$$

$$L_n = 1 + \sum_{i=2}^{n} \binom{n}{i} (-1)^i \frac{r(i-1) - \left[ip_r + (1-p_r)^i - 1\right]}{1 - \sum_{j=1}^{r} p_j^i}. \tag{5.63}$$

$$\gamma_0 = \frac{r - [p_r + (1-p_r)\ln(1-p_r)]}{-\sum_j p_j \ln p_j} \tag{5.64}$$

Con suddivisioni uguali, il miglior risultato si ha con $r=2$ che offre $S_{\max} = 0.375369$, superiore al caso di feedback binario sia con $r=2$ che con l'ottimale $r=3$. Se si accetta una ripartizione non uniforme, allora il miglior risultato si ha ancora con $r=2$, ma, a differenza del caso binario, con $p_2 = 0.582492$ e vale $S_{\max} = 0.38126$.

5.4.2 Il protocollo base FA

L'analisi è ancor più complessa che nel corrispondente caso binario. Limitiamoci ad enunciare i risultati che indicano comunque come ottima nel caso di suddivisione uniforme la suddivisione $r=3$ con $S_{\max} = 0.406970$ mentre con suddivisione ottimale si ha $p_3 = 0.370911$ e vale $S_{\max} = 0.407614$.

5.4.3 CRA a numero medio di pacchetti costante

Rispetto al caso binario, con riferimento alla (5.37), il numero di slot testati diventa $r-1$ se i primi $r-1$ sono trovati vuoti. Ciò lega ciò che accade nei sottointervalli e complica un po' il metodo di

analisi. Ridefiniamo rispetto al caso binario,

X_j il numero di sottointervalli dell'intervallo di livello j che contengono almeno un pacchetto, posto che nell'intervallo ci sia stata una collisione;

Si ha:

$$P(X=i)=\begin{cases} r\ \dfrac{e^{-\gamma_{j+1}(r-1)}(1-e^{-\gamma_{j+1}}-\gamma_{j+1}e^{-\gamma_{j+1}})}{1-e^{-\gamma_j}-\gamma_j e^{-\gamma_j}} & i=1 \\ \\ \binom{r}{i}\dfrac{e^{-\gamma_{j+1}(r-i)}(1-e^{-\gamma_{j+1}})^i}{1-e^{-\gamma_j}-\gamma_j e^{-\gamma_j}} & 2\le i\le r \end{cases} \tag{5.65}$$

In ciascuno dei sottointervalli, la probabilità di avere esattamente un pacchetto è

$$\alpha_j=\frac{\gamma_{j+1}e^{-\gamma_{j+1}}}{1-e^{-\gamma_{j+1}}}.$$

La nuova relazione ricorsiva diventa:

$$D_j=r+[D_{j+1}-\frac{1}{r}]P(X_j=1)+\sum_{i=2}^{r} iD_{j+1}P(X_j=i)(1-\alpha_j) \tag{5.66}$$

e per j sufficientemente elevato

$$D_j=(r-\frac{1}{r^2})+\frac{1}{r}D_{j+1}$$

e, nel limite, si ha:

$$\lim_{j\to\infty} D_j=\frac{r^2-1/r}{r-1} \tag{5.67}$$

Il calcolo mostra che con suddivisioni equivalenti la miglior scelta è sempre $r=2$ anche scegliendo suddivisioni diverse per il primo livello. I valori ottimali sono $S_{\max}=0.4622$ per $\gamma_o=1.25$.

5.4.4 CRA a numero medio di pacchetti ottimale

La variante rispetto al caso di feedback binario comporta che venga modificata la sola (5.45) nel seguente modo:

$$D_j = \sum_{i=1}^{r-1} [iP(F_j^{nc} = i) + (i + D_{j+1})P(F_j^c = i)] + (i - \alpha_j + D_{j+1})P(F_j^c = r)] \tag{5.68}$$

dove

$$\alpha_j = \frac{e^{-\gamma_{j+1}(r-1)}}{e^{-\gamma_{j+1}(r-1)} + \gamma_{j+1}(r-1)e^{-\gamma_{j+1}(r-1)}}$$

indica la probabilità che nelle prime $r-1$ suddivisioni non ci sia nessun pacchetto.

La (5.51) diventa

$$D_j = \frac{r-1}{r}\,\frac{2}{3}(r+1) + \frac{r-1}{r^2}\left(\frac{r}{2} + D_{j+1}\right) + \frac{1}{r^2}\left((r-1) + D_{j+1}\right) \tag{5.69}$$

con

$$\lim_{j\to\infty} D_j = \frac{1}{2} + \frac{1}{r} + \frac{2}{3}(r+1) \tag{5.70}$$

Il calcolo mostra che con suddivisioni equivalenti la miglior scelta è sempre $r = 2$ anche scegliendo suddivisioni diverse per il primo livello. I valori ottimali sono $S_{\max} = 0.4871$ per $\gamma_o = 1.27$.

In caso di suddivisione binaria non uniforme, la corrispondente della (5.54) diventa

$$D_{ij} = (1 + D_{i+1,j})P(F_{ij}^c = 1) + 2P(F_{ij}^{nc} = 2) + (2 - \alpha_{ij}D_{i,j+1})P(F_{ij}^c = 2) \tag{5.71}$$

dove, in modo simile al precedente caso, si ha

$$\alpha_{ij} = \frac{e^{-\gamma_{ij}p_1}}{e^{-\gamma_{ij}p_1} + \gamma_{ij}p_1 e^{-\gamma_{ij}p_1}}$$

E' poi:

$$\lim_{(i+j)\to\infty} D_{ij} = \frac{1 + 2p_1p_2}{1 - p_1^2 - p_2^2} \tag{5.72}$$

Il calcolo mostra che esiste una suddivisione ottimale dei livelli. La prima suddivisione conviene che sia a 0.458 e le altre a 0.485 con $\gamma_o = 1.275$. In questo caso si ha $S_{\max} = 0.4877...$

5.4.5 Problemi relativi al feedback ternario

Il feedback ternario è particolarmente vulnerabile all'errore di feedback che porta a interpretare uno slot vuoto come uno slot colliso. In questo caso l'algoritmo porta ad applicare una suddivisione che però è destinata a fornire uno slot vuoto anche nel successivo. L'esplorazione procede continuamente senza fermarsi mai perchè nella zona che si sta esplorando non esistono pacchetti.

Si può limitare l'effetto di un tale errore obbligando l'algoritmo a non operare più di n suddivisioni consecutive a seguito di slot vuoti, forzandolo, al tentativo $n+1$ a trasmettere i pacchetti nella zona non suddivisa. Questo porta ad un abbassamento dell'efficienza del protocollo, abbassamento che si riduce però in modo esponenziale al crescere di n. Una stazione che scopre allora $n+1$ slots vuoti consecutivamente sa di essere fuori passo e può passare alle procedure di riaggancio.

5.5 Limited Sensing

Gli algoritmi ottimali visti nella precedente sezioni hanno il difetto di richiedere il monitoraggio continuo del canale. Nasce allora il problema di come possa inserirsi una stazione nuova.

Le problematiche sono di due tipi. La prima riguarda il riuscire ad agganciarsi al CRA. Questo può essere fatto facilmente nel caso degli algoritmi che processano il numero di pacchetti ottimale, perchè in questi, la fine del CRA è segnalata da due consecutive trasmissioni senza collisione. Un' altra possibilità di sincronizzazione è fornita quando si osserva la successione di n pacchetti vuoti e uno colliso, tecnica vista nel precedente paragrafo per evitare *deadlocks* nell'algoritmo.

Una volta raggiunto l'aggancio col CRA non si sa però ancora come posizionare il proprio asse dei tempi per la scelta delle zone di asse da esplorare. Certo, si può far partire tale zona nell'istante di accensione. Questo viola la regola che fornisce un numero medio costante di pacchetti da processare, ed è sopportabile senza problemi solo se le stazioni che si accendono (e spengono) sono relativamente rare. Altrimenti questa tecnica inficia l'efficienza dell'algoritmo.

Ad evitare le conseguenze di cui sopra, si può utilizzare una variante Ultimo Arrivato Primo Servito (LCFS) che funziona nel seguente modo. Le stazioni che accendono aspettano di agganciarsi all'inizio di un nuovo CRA. La nuova epoca da esplorare ha, come estremo destro sull'asse dei tempi, proprio questo istante. L'estremo sinistro viene determinato in base alla lunghezza ottimale ξ della zona ancora da esplorare, escludendo intervalli già esplorati. Ciò non esclude che alla sinistra di questa zona ci siano porzioni di asse ancora inesplorate (FCFS). Anche la suddivisione delle zone avviene esplorando le suddivisioni partendo da destra (FCFS).

Poichè il cambiamento dell'ordine dell'esplorazione non altera l'efficienza dell'algoritmo le prestazioni sono le stesse del caso FCFS.

5.6 Bounds

Gli algoritmi visti nelle precedenti sezioni esemplificano lo sforzo per arrivare alla migliore efficienza date certe caratteristiche di canale. Tuttavia nessuno è ancora riuscito a trovare il miglior protocollo e i risultati trovati fin qui indicano solo dei possibili limiti inferiori *(lower bounds)*. Nel contempo, ricerche sono state fatte, intese a trovare limiti superiori *(upper bounds)* alle efficienze, in modo da delimitare la zona di possibile esistenza del miglior protocollo.

5.6.1 Kelly Bound

Si applica a protocolli che prevedono l'immediata prima trasmissione (protocolli Free Access) quando i nuovi arrivi siano di Poisson con frequenza media λ.

Siano α_0 e α_1 le probabilità che ad ogni slot si abbiano rispettivamente zero e una ritrasmissione. Allora per le ipotesi fatte si ha:

$$S = \lambda e^{-\lambda}\alpha_0 + e^{-\lambda}\alpha_1.$$

Poichè si ha $\alpha_1 \leq (1 - \alpha_0)$, si può anche scrivere

$$S \leq \lambda e^{-\lambda}\alpha_0 + e^{-\lambda}(1 - \alpha_0).$$

In condizioni di stabilità deve essere $S = \lambda$ e dunque la relazione sopra può essere scritta

$$\lambda \leq \lambda e^{-\lambda}\alpha_0 + e^{-\lambda}(1 - \alpha_0). \tag{5.73}$$

Il massimo valore di λ per cui la relazione sopra può essere soddisfatta si ha per $\alpha_0 = 0$ e in queste condizioni la (5.73) diventa

$$\lambda \leq e^{-\lambda} \tag{5.74}$$

che fornisce un $\lambda_M \leq .5671$.

5.6.2 Altri Bounds

Limiti superiori alla capacità massima di un canale con feedback ternario sono stati ottenuti a partire dal 1981 da Pippenger ($C \leq 0.744$) abbassati via via fino al più recente $C \leq 0.568$ ottenuto da Tsybakov e Likhanov (1988).

5.7 Feedbacks più generali

Le prestazioni viste nelle precedenti sezioni possono essere estese se si suppone che il canale possa fornire un feedback ancor più articolato. Sono stati studiati casi in cui si possa conoscere la molteplicità della collisione, almeno fino ad un certo numero.

In questa sezione studiamo il caso di interesse pratico in cui una stazione possa ascoltare la propria trasmissione oltre a quella degli altri e conoscere, oltre all'informazione $0, 1, C$, anche l'istante di occorrenza della ricezione della propria trasmissione. Per queste caratteristiche, che si possono trovare in un ripetitore, satellitare o non, il protocollo è stato battezzato ECHO.

L'informazione aggiunta viene utilizzata supponendo lo slot di lunghezza T', cioè maggiore del tempo di trasmissione del pacchetto T. In trasmissione, il punto di partenza x dall'inizio dello slot viene scelto uniformemente all'interno dei primi $T' - T$ secondi. In ricezione, se c'è una collisione, sfruttando la conoscenza di x si può decidere se c'è stato un pacchetto trasmesso prima, o dopo il proprio, e sfruttare tale informazione per risolvere la collisione.

Dal punto di vista pratico, l'istante di occorrenza del proprio eco può essere conosciuto a meno di un' incertezza Δ ineliminabile e può succedere che una collisione non possa venire rislota dal meccanismo di eco.

Supponendo l'intervallo $T' - T$ multiplo secondo r dell' intervallo di incertezza Δ, il risultato di una collisione può essere classificato da una stazione che ha colliso, nei seguenti tre modi:

F (First) se la stazione si ritiene la prima nel burst colliso. Ciò capita se l'eco di inizio trasmissione viene ricevuto nel Δ in cui lo si aspetta;

L (Last) se la stazione si ritiene l' ultima nel burst colliso. Ciò capita se l'eco di fine trasmissione viene ricevuto nel Δ in cui lo si aspetta;

M (Middle) negli altri casi.

L'algoritmo si basa ancora sulle epoche di arrivo di lunghezza ξ che vengono esplorate suddividendole, in seguito a collisioni, in r zone uguali.

Il CRA opera seguendo un albero binario in cui nel prossimo slot vengono trasmessi i pacchetti delle stazioni F; quando eventuali collisioni fra questi sono risolte vengono trasmessi i pacchetti delle stazioni L. I pacchetti delle stazioni M non vengono trasmessi e vengono rimandati alla successiva epoca.

Le zone di asse esplorate non costituiscono più, in generale, un intervallo ma sono formate da intervalli disgiunti. Ciò non impedisce di vedere l'epoca come un unico intervallo e di procedere con l'algoritmo.

Il throughput deve ora tener conto del fatto che l'intervallo $(r-1)\Delta$ costituisce un overhead (un

Δ è sempre presente e non viene conteggiato) e viene espresso come:

$$S = \frac{N(\gamma, r)/D(\gamma, r)}{1 + (r-1)\Delta} \tag{5.75}$$

dove, al solito, $N(\gamma, r)$ è il numero medio di pacchetti trasmessi con successo in un CRP, $D(\gamma, r)$ è la lunghezza media del CRP stesso, T viene considerato unitario e γ è il numero medio di pacchetti in un'epoca.

Il rapporto N/D in (5.75) può essere espresso come:

$$\frac{N}{D} = \frac{\gamma e^{-\gamma} + (1 - e^{-\gamma} - \gamma e^{-\gamma}) N_0}{1 + (1 - e^{-\gamma} - \gamma e^{-\gamma}) D_0} \tag{5.76}$$

dove N_0 and D_0 hanno il solito significato. Si ha poi:

$$\gamma_j = \frac{\gamma}{r^j},$$

Nel calcolo del throughput si possono identificare, a seguito di una collisione a livello j, quattro eventi:

$\mathcal{A}_j$ = Classi F e L disgiunte con un solo pacchetto per ciascuna classe;

$\mathcal{B}_j$ = Classi F e L disgiunte con un solo pacchetto in classe F (o L) e più di uno in classe L (o F);

$\mathcal{C}_j$ = Classi F e L disgiunte con più di un pacchetto in ciascuna classe;

$\mathcal{D}_j$ = Classi F e L coincidenti (e ovviamente, con almeno due pacchetti).

Possiamo scrivere:

$$P(\mathcal{D}_j) = r \; \frac{e^{-\gamma_{j+1}(r-1)}(1 - e^{-\gamma_{j+1}} - \gamma_{j+1} e^{-\gamma_{j+1}})}{1 - e^{-\gamma_j} - \gamma_j e^{-\gamma_j}} \tag{5.77}$$

$$P(\mathcal{A}_j) = \left(\frac{\gamma_{j+1} e^{-\gamma_{j+1}}}{1 - e^{-\gamma_{j+1}}} \right)^2 (1 - P(D_j)) \tag{5.78}$$

$$P(\mathcal{B}_j) = 2 \frac{\gamma_{j+1} e^{-\gamma_{j+1}} (1 - e^{-\gamma_{j+1}} - \gamma_{j+1} e^{-\gamma_{j+1}})}{(1 - e^{-\gamma_{j+1}})^2} (1 - P(D_j)) \tag{5.79}$$

$$P(\mathcal{C}_j) = \left(\frac{1 - e^{-\gamma_{j+1}} - \gamma_{j+1} e^{-\gamma_{j+1}}}{1 - e^{-\gamma_{j+1}}} \right)^2 (1 - P(D_j)) \tag{5.80}$$

Per N_j e D_j si hanno poi le seguenti equazioni ricorsive:

$$N_j = 2P(A_j) + (1 + N_{j+1})P(B_j) + 2N_{j+1}P(C_j) + N_{j+1}P(D_j) \tag{5.81}$$

$$D_j = 2P(A_j) + (2 + D_{j+1})P(B_j) + (2 + 2D_{j+1})P(C_j) + (1 + D_{j+1})P(D_j) \tag{5.82}$$

Si ha poi facilmente:

$$\lim_{j \to \infty} N_j = 2$$

$$\lim_{j \to \infty} D_j = 2 + \frac{1}{r - 1}$$

Nel caso ideale in cui $\Delta = 0$ si può avere una forma chiusa per il throughput dalle (5.75) e (5.76) osservando che con un overhead arbitrariamente piccolo, ad ogni collisione si determinano esattamente un pacchetto in classe F e uno in classe L. Dunque $N_0 = 2$ e $D_0 = 2$ e :

$$S_0 = \frac{\gamma e^{-\gamma} + 2(1 - e^{-\gamma} - \gamma e^{-\gamma})}{1 + 2(1 - e^{-\gamma} - \gamma e^{-\gamma})} \tag{5.83}$$

che ha il suo massimo valore per $\gamma = 2.9$ dove vale $S_{\max} = 0.673$; In Figura 5.6 sono riportati i valori di massimo throughput, ottimizzati su r e γ, e i corrispondenti valori di r, in funzione di pratici valori di Δ.

Si noti che l'algoritmo può essere migliorato perchè quando si verifica l'evento sopra indicato con $\mathcal{D}_j$, nello slot successivo si ha sicuramente una nuova collisione. Ciò può essere evitato, come nel caso di feedback ternario, suddividendo il sottointervallo in due parti ed esplorando separatamente ciascuna parte. Tuttavia, per pratici valori di r, il vantaggio non è apprezzabile perchè la probabilità di tale evento è molto piccola.

5.8 Protocolli con lungo tempo di propagazione

Nelle sezioni precedenti abbiamo supposto che il feedback di canale fosse disponibile alla fine della trasmissione. In molti casi però, il feedback non è disponibile che dopo un tempo di propagazione τ ammontante a parecchi tempi di trasmissione. E' questo il caso delle trasmissioni via satellite o delle trasmissioni ad altissima velocità.

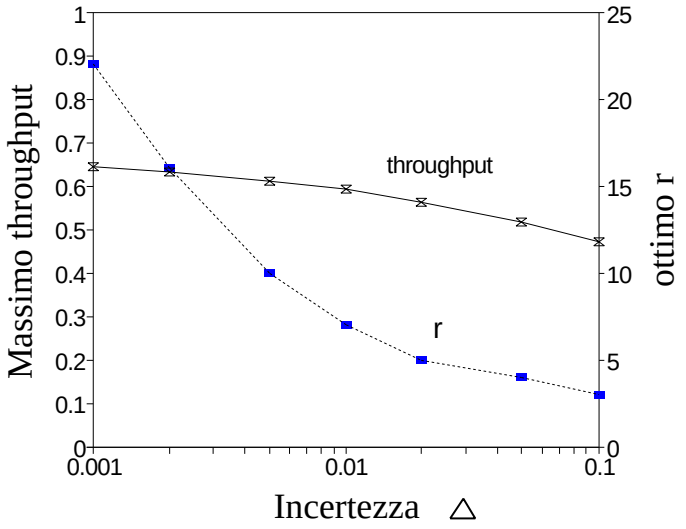

Figura 5.6 *Massimo throughput e miglior r nel protocollo ECO-CRA*

I protocolli visti nelle precedenti sezioni possono essere facilmente adattati a questo ambiente suddividendo il canale, che supponiamo suddiviso in slots, in un numero di sottocanali $r \geq \tau/T$, ciascun canale essendo composto da uno slot ogni r. I protocolli possono essere applicati separatamente a ciascun sottocanale. Esistono però due alternative principali, riguardo a come assegnare il traffico ai sottocanali.

La prima alternativa considera il traffico offerto suddiviso rigidamente fra i canali. Il traffico che arriva in periodi assegnati a un sottocanale, viene deterministicamente assegnato a quel sottocanale o a un determinato altro canale. In questo caso i protocolli agiscono indipendentemente e le prestazioni di throughput su tutto il canale sono perfettamente uguali al caso indiviso con $\tau = T$.

Nel secondo caso, le prestazioni dipendono dalla legge di assegnazione. Considerando protocolli Free Access, il primo slot in cui trasmettere diventa, anzichè il primo di un certo sottocanale, il primo in assoluto. In questo specifico caso si può mostrare che una tale variante aumenta il throughput globale. Tale aumento è però possibile solo perchè i protocolli FA sono subottimali.

5.9 Protocolli con limitato tempo di propagazione

All'altro estremo rispetto al caso considerato nella precedente sezione, c'è il caso in cui il feedback è disponibile *prima* della fine della trasmissione e questo fatto può essere sfruttato per aumentare l'efficienza dei protocolli. Basti pensare che in un sistema sincronizzabile gli slot possono essere fatti di durata uguale la tempo di feedback. In questo caso, se la durata della trasmissione è un multiplo di slot, il meccanismo di accesso casuale può essere visto, in caso di successo, come una prenotazione per la successiva trasmissione senza contesa negli slot successivi. Anche qui possiamo applicare i diversi meccanismi visti per accedere allo slot di "prenotazione". Il throughput corrispondente può essere calcolato facilmente (si veda più avanti) in base al throughput dei vari meccansimi di accesso.

Anche nel caso di canali a bus, non efficacemente slottizzabili, si possono ottenere grandissimi vantaggi se il ritardo di feedback è minore della durata della trasmissione.

5.9.1 CSMA

Il Carrier Sensing Multiple Access (CSMA) è il protocollo che direttamente deriva dall'ALOHA base, con l'aggiunta del feedback che riguarda l'occupazione del canale stesso. Lo strumento che rivela l'occupazione del canale viene chiamato *Carrier Sensing*, che dà il nome al protocollo.

L'operazione del CSMA consiste nel monitorare il canale e nell'astenersi da ogni trasmissione se il Carrier Sensing indica che il canale è occupato. Se il canale è libero il protocollo agisce come l'ALOHA e trasmette appena pronto il messaggio. In caso di collisione un nuovo tentativo viene effettuato dopo un ritardo casuale. Ulteriori precisazioni sulle diverse varianti possibili sono indicate più sotto.

Si noti che nonstante l'ascolto del canale le collisioni sono ancora possibili se il tempo di propagazione fra una stazione e l'altra è $\tau > 0$, come avviene in pratica. Ciò è mostrato in Figura 5.7a dove sono indicate le attività del canale come viste dal Carrier sense di due stazioni poste a distanza τ. Con riferimento alla trasmissione effettuata in una stazione, si vede che esiste un Periodo Vulnerabile nel quale le altre stazioni a distanza τ sono "cieche". Queste non possono aver trasmesso prima di questo periodo perchè altrimenti la prima stazione avrebbe visto il canale occupato e non avrebbe trasmesso. Non possono nemmeno trasmetter dopo questo periodo perchè la trasmissione della prima stazione arriva e alza il Carrier Sense. Ma il protocollo non può impedire trasmissioni durante il periodo vulnerabile, il che causa le collisioni. E' intuibile che le prestazioni dipendono dal valore di τ e migliorano al decrescere di questo (per $\tau = 0$ non si avrebbero collisioni e il troughput massimo potrebbe raggiungere il 100%.). Vedremo però che il parametro che influenza le prestazione è il rapporto $a = \tau/T$, con T tempo di trasmissione del pacchetto.

In Figura 5.7b è mostrato un caso estremo in cui $\tau = T/2$. Oltre questo valore di τ accade che l'informazione che si trae dal Carrier Sense può non essere più utile. Nel caso indicato infatti si avrebbe collisione sull'asse dei tempi inferiore, mentre la collisone sarebbe assente sull'asse superiore.

Riguardo al meccanismo base CSMA sono possibili diverse varianti:

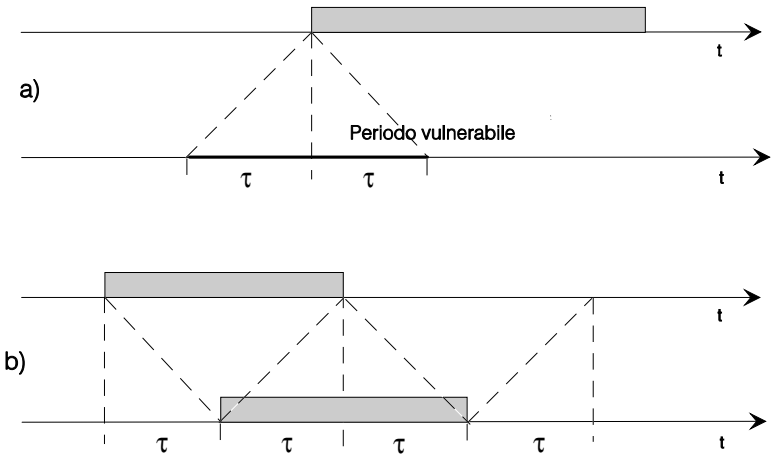

Figura 5.7 *Funzionamento del protocollo CSMA. a) periodo vulnerabile. b) relazioni temporali affinchè una collisione sia vista da tutte le stazioni sul bus.*

non-persistent CSMA. Quando, al momento individuato per la trasmissione o ritrasmissione, il canale è sentito già attivo, la trasmissione viene rimandata a un nuovo tempo scelto in modo casuale, come nel caso di una collisione.

persistent CSMA. Quando, al momento individuato per la trasmissione o ritrasmissione, il canale è sentito già attivo, la trasmissione viene trattenuta finchè il canale viene sentito libero e da questo momento viene rilasciata.

Esistono poi le varianti chiamate p**-persistent** in cui, quando al momento della trasmissione o ritrasmissione il canale è sentito già attivo, la modalità di funzionamento persistent viene applicata con probabilità p e con $1-p$ si usa la modalità non-persistent.

Le prestazioni di un tale sistema nel caso non-persistent può essere valutata utilizzando il modello di popolazione infinita già usato per l' ALOHA. Tuttavia le prestazioni dipendono dalla topologia delle varie stazioni connesse al bus. Un caso semplice, cui faremo riferimento in seguito per il calcolo delle prestazioni è quello di un sistema a topologia stellare (nei riguardi della propagazione del segnale) in cui tutte le stazioni distano egualmente dal centro stella in cui τ denota il tempo di propagazione da una stazione all'altra, uguale per tutte le coppie.

Utilizziamo al solito come unità il tempo T di trasmissione di un pacchetto e indichiamo con G il numero medio di trasmissioni e ritrasmissioni, supposte di Poisson, in un tale tempo unitario.

Il throughput può essere calcolato considerando il processo rigenerativo occupazione del canale. Detto B il tempo medio in cui il canale è continuamente occupato (Busy Period) e I quello per cui è continuamente libero (Idle Period), il throughput S può essere espresso come

$$S = \frac{\alpha}{B+I} \tag{5.84}$$

dove $\alpha = e^{-aG}$ è la probabilità che un busy period sia formato da una trasmissione non collisa.

L' idle period è il tempo d'attesa al prossimo arrivo, che, per la proprietà di non memoria, non risulta dipendere dall'istante in cui scatta l'attesa. La sua distribuzione di probabilità è perciò esponenziale negativa con valor medio

$$I = \frac{1}{G} \tag{5.85}$$

Detto $a = \tau/T$ il tempo di propagazione normalizzato, la lunghezza media del busy period è data da

$$B = 1 + a + Z(1 - e^{-aG}) \tag{5.86}$$

dove a è il tempo impiegato dalla fine della trasmissione a giungere a tutte le stazioni (si suppongono incluse anche quelle trasmittenti, ma la cosa non ha impatto col modello a popolazione infinita dove le stazioni trasmettono una sola volta) e Z rappresenta, in caso di collisione, il tempo medio fra la prima e l'ultima delle trasmissioni concorrenti.

Essendo i tentativi di trasmissione modellati secondo Poisson, Z viene calcolato come $a - E[X]$ essendo X la distanza dell'ultimo arrivo in a dall'estremo superiore dell'intervallo di lunghezza a iniziato dalla prima trasmissione. Per le proprietà di Poisson, X è distribuito come un esponenziale troncato in $[0, a]$ con media

$$E[X] = \frac{1}{G} - \frac{ae^{-aG}}{1 - e^{-aG}}$$

e il busy period medio, grazie al teorema della probabilità totale, ha espressione

$$B = 1 + a + (a - E[X])(1 - e^{-aG}) \tag{5.87}$$

Sostituendo in (5.84) i valori trovati si ottiene infine:

$$S = \frac{Ge^{-aG}}{G(1 + 2a) + e^{-aG}} \tag{5.88}$$

Le curve (5.88) per diversi valori di a sono riportate in Figura 5.8. Si vede che il massimo valore di throughput sale al decrescere di a, come ovvio, ma che in ogni caso,

$$\lim_{G \to \infty} S = 0.$$

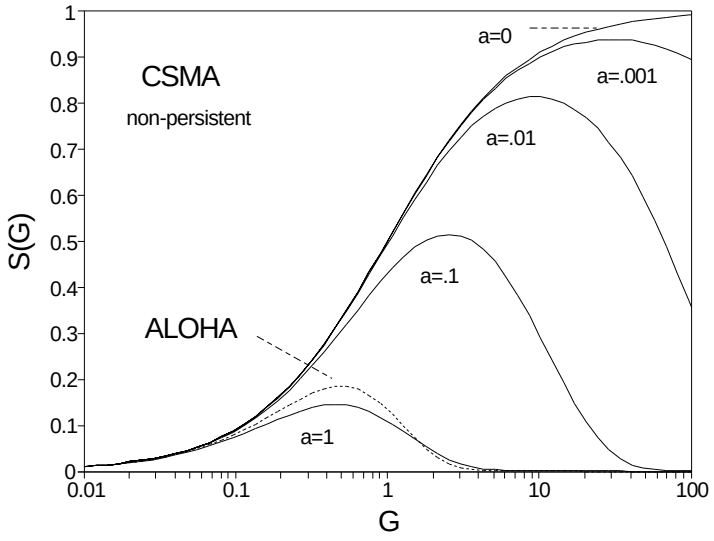

Figura 5.8 *Curve di throughput del protocollo CSMA non-persistent per vari valori del tempo di propagazione a*

Ciò suggerisce, come già nel caso ALOHA, che un protocollo di tal genere sia instabile, anche se l'instabilità ha effetti diversi al crescere di a.

Anche in questo caso si possono fare modelli più precisi con popolazione finita e si vede che la scala dei tempi su cui si manifesta tale instabilità, presente solo se il numero di stazioni è troppo grande, cresce al decrescere di a.

E' possibile anche considerare una versione slottizzata del protocollo CSMA in cui le trasmissioni siano sincronizzate all'inizio dello slot (cosa complessa nei canali a bus), con durata dello slot, normalizzata al tempo di trasmissione, pari ad a', con $a < a' < 1$. La condizione $a' > a$ permette di scoprire la collisione nello slot successivo alla trasmissione. Nell'ipotesi che a' sia un sottomultiplo esatto di 1, il ciclo vale $1 + Ka'$ dove 1 è il periodo di trasmissione, collisione o no, e $K \geq 1$ è una variabile geometrica con probabilità di successo (almeno un tentativo) pari a $1 - e^{-a'G}$. Dunque il valor medio del ciclo è

$$C = 1 + E[K]a' = 1 + a'\frac{1}{1 - e^{-a'G}} = \frac{a' + 1 - e^{-a'G}}{1 - e^{-a'G}} \tag{5.89}$$

Il ciclo contiene un pacchetto corretto solo se fra coloro che iniziano il nuovo ciclo ce n'è solo 1, e ciò capita con probabilità:

$$\alpha = \frac{a'Ge^{-a'G}}{1 - e^{-a'G}} \tag{5.90}$$

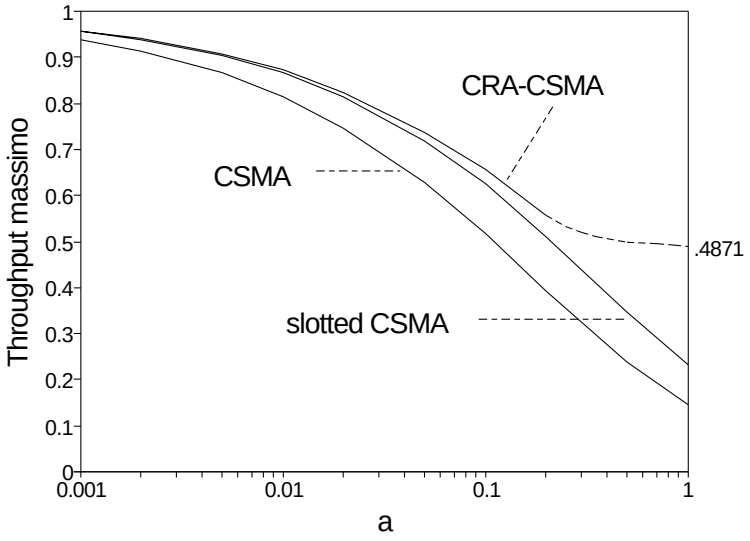

Figura 5.9 *Massimo throughput dei protocolli CSMA slotted, non slotted e CRA-CSMA*

Il throughput risulta allora:

$$S = \frac{a' G e^{-a'G}}{1 + a' - e^{-a'G}} \tag{5.91}$$

Per il tempo di slot normalizzato a' si è utilizzata una simbologia diversa rispetto al caso non slotted perchè nei canali più spesso utilizzati (radio o buses) lo slot deve essere almeno due volte il tempo di propagazione massimo. In questo caso $a' \geq 2a$. In Figura 5.9 sono riportati i valori di throughput massimo in funzione di a e a' delle due curve. Si vede che nel caso $a' = 2a$ non si ha praticamente differenza.

5.9.2 CRA-CSMA

Il CSMA, come visto nella precedente sezione, non è stabile con popolazione infinita. Si può però stabilizzarlo introducendo un algoritmo per la soluzione come quelli già visti nelle sezioni precedenti.

Consideriamo la versione migliore dei CRA visti, CRA a numero medio di pacchetti ottimale, con feedback ternario. In questo caso, considerando la versione slotted, uno slot vuoto è riconosciuto alla fine dello slot, quindi dopo τ secondi, mentre una trasmissione dura T e viene riconosciuta come corretta o collisa dopo $T + \tau$ secondi. Questa differenza complica le cose perchè la suddivisione ottimale dell'asse dei tempi esplorato non è più uniforme.

Detto ancora γ il valor medio di pacchetti nell' epoca considerata, la formula del massimo throu-

ghput è ancora data dalla (5.44) con la variante che qui riscriviamo,

$$S_{\max} = \frac{\gamma e^{-\gamma} + (1 - e^{-\gamma} - \gamma e^{-\gamma}) N_0}{a' e^{-\gamma} + (1 + a')\gamma e^{-\gamma} + (1 - e^{-\gamma} - \gamma e^{-\gamma})(1 + a' + D_0)} \tag{5.92}$$

essendo normalizzata a 1 la durata della trasmissione e il throughput in pacchetti/tempo di trasmissione. In questa normalizzazione lo slot ha lunghezza a'.

Poichè vedremo che, per controbilanciare il lungo tempo di trasmissione/collisione, l'epoca da esplorare contiene un numero medio di pacchetti molto piccolo, escluderemo la possibilità di avere più di due pacchetti in ogni collisione. Con ciò, si ha $N_0 = 2$ e

$$1 - e^{-\gamma} - \gamma e^{-\gamma} \simeq \frac{\gamma^2}{2} e^{-\gamma}$$

e la (5.92) si riduce a

$$S_{\max} = \frac{\gamma + \gamma^2}{a' + (1 + a')\gamma + (\gamma^2/2)(1 + a' + D_0)} \tag{5.93}$$

Grazie all'approssimazione introdotta, indicata al solito con p la ripartizione dell'epoca esplorata, si ha la seguente ricorsione in $D = D_0$:

$$D = (1 - p)^2 (a' + D) + 4p(1 - p)(1 + a') + p^2 (1 + a' + D) \tag{5.94}$$

che, risolta in D, fornisce la soluzione cercata. Poichè qui compare la sola dipendenza da p, si può cercare il valore di p che minimizza D. Questo risulta

$$p = \sqrt{a' + a'^2} - a'. \tag{5.95}$$

Si può poi cercare il miglior γ. I risultati di questa procedura sono esposti in Figura 5.9. I conti si arrestano al valore $a' = 0.2$ perchè oltre non valgono più le ipotesi fatte (γ ottimo in questo punto vale 0.53 e la probabilità che si abbiano più di due pacchetti risulta .0167). Si sa però che in $a' = 1$ il valore deve essere pari a .4871.

Si noti come nella zona d'interesse $a' << 1$, l'introduzione del miglior CRA non offra guadagni di throughput sensibili rispetto allo slotted CSMA. Ciò è dovuto al fatto che l'efficienza del CRA sta nel risolvere efficacemente le collisioni, che però in questo caso avvengono raramente. Sempre nella zona citata si può mostrare che una buona approssimazione al massimo throughput è

$$S_{\max} \simeq \frac{1}{1 + \sqrt{2a'}} \tag{5.96}$$

in $\gamma \simeq \sqrt{2a'/(1 + \sqrt{a'})}$.

5.9.3 CSMA-CD

Il protocollo CSMA ha lo svantaggio di sprecare tutto il tempo di trasmissione quando una collisione è in atto. In alcuni casi è possibile avere un rivelatore di collisioni quando la trasmissione è ancora in atto (Collision Detection (CD)). In questo caso, appena scoperta una collisione, la trasmissione viene abortita. Una variante di questo genere prende il nome di CSMA-CD.

Dal punto di vista pratico, per come è operativamente fatto il rivelatore di collisioni, può essere necessario continuare la trasmissione per un tempo $\delta < 1$ dopo la scoperta di una collisione. Ciò per permettere a tutte le stazioni di verificare con sicurezza una collisione. Evidentemente, se fosse $\delta = 1$ si ritorna all'efficienza del CSMA.

Anche con questo protocollo sono possibili le diverse versioni già viste per il CSMA. Qui calcoleremo le prestazioni del solo caso *non persistent* col modello di popolazione infinita e con la topologia stellare già indicata per il CSMA. L'unica variante rispetto al caso CSMA è che ora il busy period ha espressione

$$B = (1+a)e^{-aG} + (Z+\delta+a)(1-e^{-aG})$$

Utilizzando i risultati già noti per I e Z, e si ottiene:

$$S = \frac{Ge^{-aG}}{G(1+2a) + e^{-aG} - G(1-\delta)(1-e^{-aG})} \tag{5.97}$$

dove, a denominatore viene esplicitata la differenza, dovuta a δ, col denominatore della (5.88).

Le curve di throughput sono simili a quelle già viste per il CSMA, ma a pari a hanno il massimo più alto, più piatto e più spostato verso destra, mostrando una stabilità accresciuta. In Figura 5.10 è mostrato il throughput massimo in funzione di a nel caso più favorevole in cui $\delta = 0$.

5.9.4 CRA-CSMA-CD

Anche qui vale il discorso già fatto per il CRA-CSMA. Tuttavia, qui il caso è più semplice e generalizzabile ad altri ambienti. Il modello consiste nel considerare un canale slottizzato con slots di lunghezza normalizzata a' in cui le stazioni possono iniziare a trasmettere il pacchetto e ad ogni slot verificare se la trasmissione non ha colliso. In questo caso la stazione continua la trasmissione del pacchetto, alla fine del quale riprendono gli slots. In caso contrario la trasmissione riprende daccapo e si applica un algoritmo per la soluzione delle collisioni. In sostanza, gli slots servono a prenotare il canale e ci riferiremo alle trasmissione in questi slots come a *prenotazioni*.

Detto s il massimo throughput delle prenotazioni, che dipende dal CRA scelto, il massimo throu-

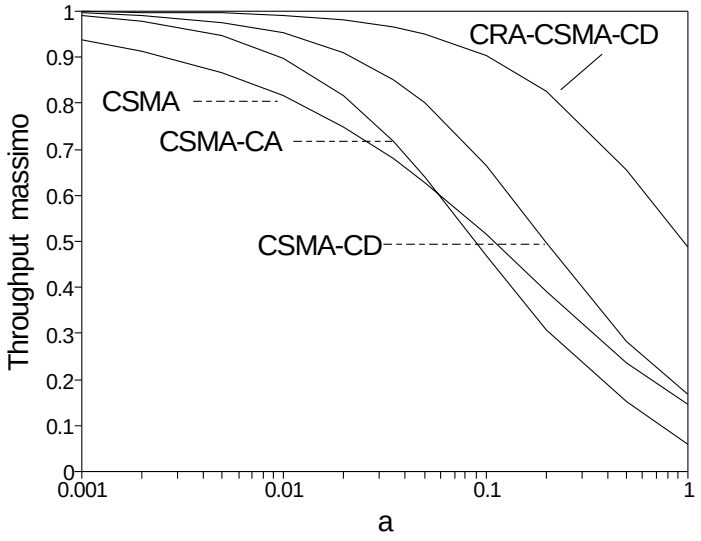

Figura 5.10 *Massimo throughput dei protocolli CSMA, CSMA-CD, CRA- CSMA-CD e CSMA-CA*

ghput di canale è dato da

$$S_{\max} = \frac{s}{(1-s)a' + s} \tag{5.98}$$

In Figura 5.10 è mostrato anche il throughput di tale protocollo supponendo di utilizzare come CRA l'algoritmo Gallager-Tsybakov. Si noti come nel caso $a' = 1$ il throughput si riduca a 0.4871. Naturalmente occorre tener conto delle perdite dovute alla slottizzazione del canale.

5.9.5 Meccanismi di priorità

Un meccanismo di priorità che funziona coi meccanismi Carrier Sense, con o senza Collisison Detect è quello di anteporre al pacchetto un preambolo di lunghezza $l_i = (i+1)2\tau$, dove i è un indice distribuito fra gli utenti (si veda la Figura 5.11). Gli utenti trasmettono secondo la regola del Carrier Sense e decidono se esiste collisione alla fine del preambolo. Se non esiste collisione la trasmissione prosegue, altrimenti viene interrotta. E' evidente che in questo caso vince la trasmissione con preambolo più lungo. In realtà, il meccanismo di cui sopra funziona anche con preambolo di lunghezza $2i\tau$.

Gli indici i possono essere fatti ruotare fra gli utenti nel seguente modo: ad ogni trasmissione corretta, che è sentita da tutti gli utenti, gli utenti che non hanno trasmesso passano dalla classe i alla classe $i+1$ mentre quello che ha trasmesso passa alla classe più bassa. Restano problemi di inizializzazione e il fatto che il throughput decresce con Ma. Si può rimediare in parte ammettendo che più utenti appartengano alla stessa classe, ma in questo caso non è garantito che ci sia un vincitore nella collisione.

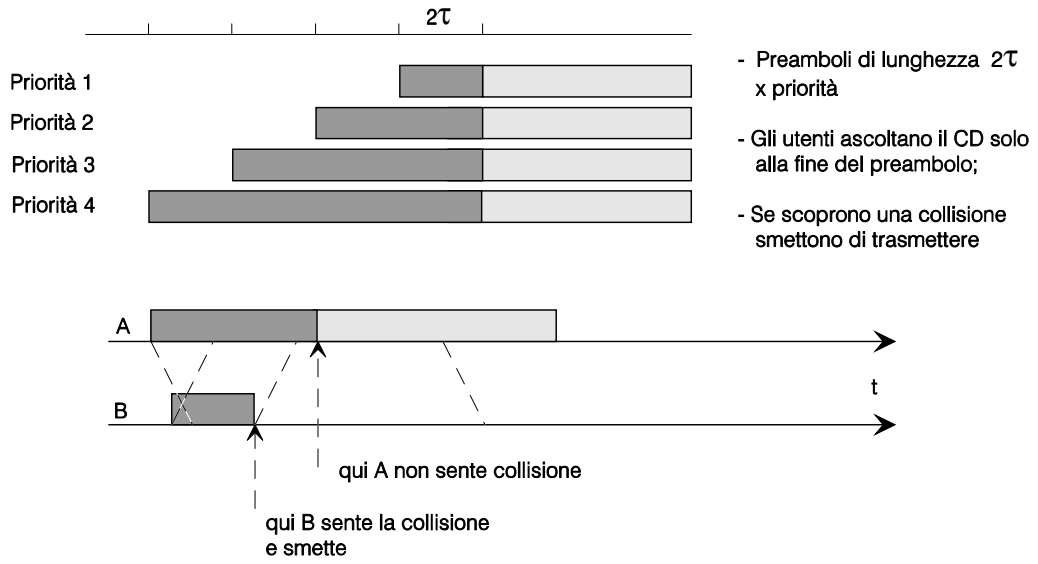

Figura 5.11 *Meccanismo di priorità con preambolo e suo funzionamento.*

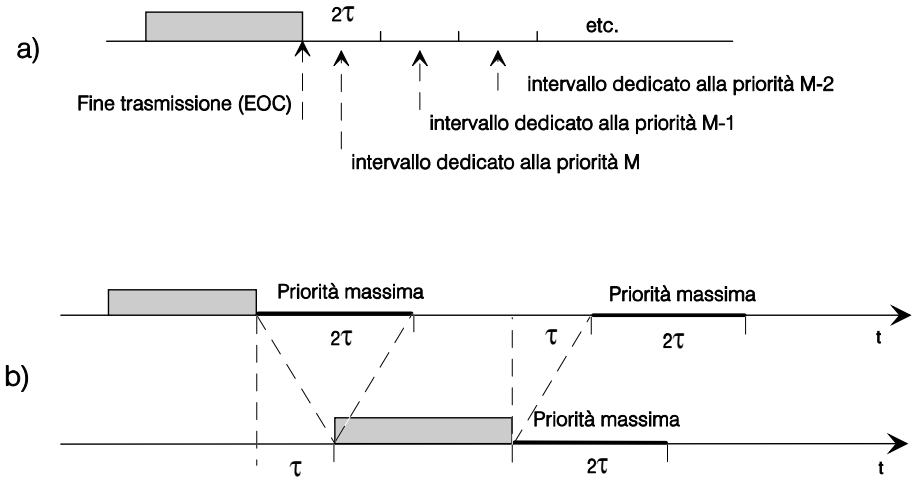

Figura 5.12 *Meccanismo di priorità. Nella parte a) è mostrato lo schema, mentre nella parte b) le relazioni temporali.*

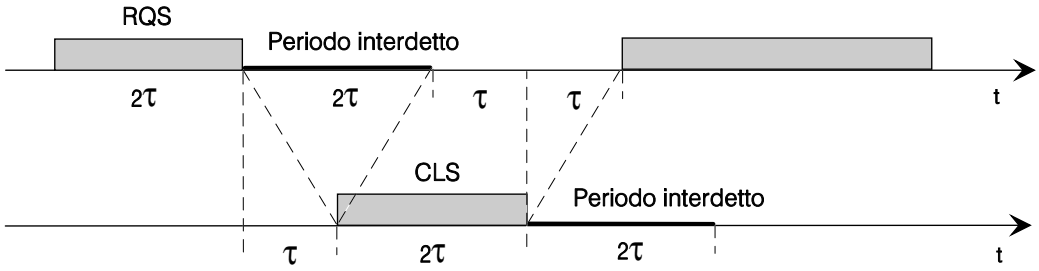

Figura 5.13 *Meccanismo CSMA Collision Avoidance*

Un altro meccanismo di priorità è quello che si applica al fronte di discesa del segnale sul canale, identificando, a partire da questo degli slot $i = 1, 2, ...M$ di lunghezza un poco maggiore di 2τ. Lo slot numero i è riservato per l'accesso alla classe i (si veda la Figura 5.12).

Evidentemente, se qualcuno di priorità più alta trasmette prima la sequenza degli slot si interrompe e riprende da capo a fine trasmissione, di modo che il canale potrebbe essere sempre monopolizzato, come giusto, dal traffico a priorità più alta.

A partire dallo slot M la sequenza può ripetersi, oppure l'accesso può essere lasciato libero, tipo CSMA o CSMA-CD.

Una priorità di questo tipo è utilizzata in alcuni casi per trasmettere gli ACK cui viene riservato il primo slot dopo il fronte di fine trasmissione di un burst.

5.9.6 CSMA-CA

Un ultimo cenno merita il meccanismo CA in assenza di meccanismo CD. Questo càpita negli accessi radio, dove è praticamente impossibile trasmettere ed ascoltare il canale sulla stessa frequenza, ed è dunque impossibile scoprire una collisione mentre si trasmette.

Il meccanismo usato in questo caso ha entrambi gli effetti CD e CA. Esso prevede l'uso del meccanismo "Request To Send" (RQS) - "Clear To Send" (CLS), il cui funzionamento è illustrato in Figura 5.13. Per accedere al canale il meccanismo prevede la trasmissione di un segnale, RQS appunto, che, ad evitare ambiguità, è codificato e in pratica contiene l'indirizzo del destinatario o di una stazione centrale preposta all'accesso e un CRC. Se RQS viene ricevuto dal destinatario senza collisione allora, in risposta, questo emette un segnale, ancora con l'indirizzo del richiedente e CRC, che opera come CLS. Solo dopo la ricezione del CLS, e dunque in assenza di collisioni, il richiedente trasmette infine il pacchetto.

L'accesso al canale è temporizzato dai vari fronti di discesa del segnale esattamente come indicato nella sezione precedente per le priorità. In pratica il primo slot dopo la discesa del carrier sense è dedicato alla trasmissione della segnalazione delle stazione coinvolte e dunque le altr stazioni si astengono dal trasmettere in questi periodo. Così, dopo la ricezione corretta di RQS la stazione

coinvolta trasmette CLS e alla ricezione corretta di questo il terminale originatore trasmette il messaggio. Naturalmente possono esserci collisioni che coinvolgono il segnale RQS. Affinchè una collisione sia vista da tutti occorre che la lunghezza di ciascun messaggio sia almeno uguale a 2τ e questa deve essere allora la minima lunghezza del segnale RQS. Se si verifica una collisione fra questi segnali il CLS non viene trasmesso e, trascorso il successivo periodo di deferenza 2τ, il meccanismo può essere ripreso. Se il segnale RQS non collide, allora la trasmissione del pacchetto è garantita. Dunque, questo meccanismo opera come un collision detect.

Il calcolo del throughput può essere effettuato come nel CSMA-CD, con la differenza che, detto p il ritardo di processing con cui si ha la risposta ai segnali RQS e CLS e supponendo che questi abbiano lunghezza $b \geq 2a$, normalizzati al tempo di trasmissione, si ha (in Figura 5.13 si è supposto b=2a e p=0):

$$B = (1 + 3a + 2b + 2p)e^{-aG} + (Z + a)(1 - e^{-aG})$$

Utilizzando i risultati già noti per B e Z, si ottiene:

$$S = \frac{Ge^{-aG}}{G(1 + 2a) + e^{-aG} - G(1 - e^{-aG}) + (2a + 2b + 2p)Ge^{-aG}} \tag{5.99}$$

dove, ancora, si è messa in luce la differenza del denominatore coi denominatori delle ((5.88) e ((5.97).

In Figura 5.10 è mostrato il throughput massimo di questo protocollo, indicato come CSMA-CA, in funzione di a nel caso più favorevole in cui $b = 2a$ e $p = 0$.

5.9.7 Il terminale nascosto

Nelle reti radio a bus, specialmente per terminali mobili, può succedere che qualche terminale venga nascosto alla "vista" di qualcun altro. Può succedere così che la coppia di terminali A e B si "veda" e così la coppia B e C, ma non la coppia A e C. Se A trasmette verso B il Carrier Sense di C è vanificato e se C trasmette, le due trasmissioni si distruggono.

Il meccanismo RQS e CLS del CSMA-CA elimina l'inconveniente del terminale nascosto. Infatti, C non sentirebbe il RQS di A, ma la collisione, rivelata dall'assenza del CLS, sarebbe limitata al RQS (o al più al CLS stesso).

5.9.8 DSMA-CD

Per evitare che si presenti il problema del terminale nascosto occorre ricorrere a una stazione centrale che veda tutti i terminali e faccia da ripetitore. Questo è il caso dei sistemi radio cellulari per dati,

che richiedono comunque una stazione base per accedere alla rete fissa e dove le comunicazioni da e per la base avvengono su due canali (o frequenze) diverse. In questo caso segnali equivalenti al Carrier Sense e al Collision Detect possono essere ritrasmessi ai terminali dalla stazione base, per esempio con opportuni "flags" numerici intercalati sul canale verso i terminali. Questa tecnica è usata nel sistema Cellular Digital Packet Data, uno standard americano per trasmissione dati cellulari a 19.4 Kb/s che è in grado di coesistere con la telefonia cellulare analogica americana AMPS. In questo standard l'acronimo DSMA sta per Digital Sense Multiple Access.

5.10 Protocolli d'accesso casuale standard

5.10.1 Protocollo 802.3

Il protocollo 802.3 è lo standard IEEE che deriva dalla rete Ethernet. Utilizza la trama MAC mostrata in Figura 5.14 e una velocità di trasmissione 10 Mb/s. Le primitive usate sono solo tre, ossia MA_UNIDATA.request per chiedere la trasmissione, MA_UNIDATA.confirm per segnalare la ricezione e MA_UNIDATA.confirm per segnalare l'avvenuta trasmissione o il fallimento della stessa.

Il protocollo d'accesso utilizzato è una variante del CSMA-CD 1-persistent, nota come "truncated binary exponential backoff" che funziona come segue. Viene definito uno *slot time* corrispondente a 512 bit, che è la base dei tempi del protocollo e che corrisponde anche al pacchetto più piccolo trasmissibile. Questo asse dei tempi discreto ha solo importanza locale e non richiede la sincronizzazione fra le stazioni. In breve, se al momento della richiesta di trasmissione il canale è libero si trasmette. Altrimenti, si attende la discesa del CS e quindi si trasmette. Se da questo istante il Collision Detect (CD) segnala collisione si abortisce la trasmissione, si trasmette una sequenza di jamming lunga 32 bit per rafforzare la collisione cosicchè tutti la sentano bene, e si inizia la procedura di ritrasmissione. Questa consiste nello scegliere un numero intero r di slots trascorso il quale la procedura viene ripetuta. Il numero r è scelto uniformemente nell'intervallo $0 \leq r \leq 2^k$, dove k è posto a zero all'inizio di ogni nuova procedura ed incrementato di uno ad ogni fallimento fino al raggiungimento di un massimo valore pari a 10. Sono ammessi 12 tentativi di trasmissione, dopodichè il MAC segnala al LLC il fallimento della trasmissione.

La filosofia che ispira questo protocollo è quella della rapidità, da qui la scelta dello 1-persistent. Il numero crescente di tentativi falliti indica pero' affollamento e l'aumento del campo di r è un tentativo di ridurlo. Si noti che il tipo di backoff esponenziale usato, che tende a diradare gli accessi al canale la crescere del numero delle collisioni, rappresenta un funzionamento diverso da quello che si ha con un backoff di tipo costante, come quello usato nei modelli ALOHA di cui abbiamo studiato la stabilità. Ciò riapre la questione se un protocollo ALOHA col presente tipo di backoff sia stabile o meno. E' stato mostrato che, anche qualora la legge esponenziale crescesse senza il vincolo dei 10 tentativi, si ha comunque instabilità.

La lunghezza dello slot coincide con il parametro 2τ tipico delle reti CSMA che vincola la massima distanza end-to-end fra due stazioni. Infatti la condizione affinchè pacchetti di lunghezza minima che collidono vengano visti collisi da ogni posizione in rete è che sia $T \leq 2\tau$, ossia $a \leq 0.5$. Alla velocità del segnali pari a 2×10^8 m/s e alla frequenza di 10 Mb/s la distanza 2τ percorsa in un

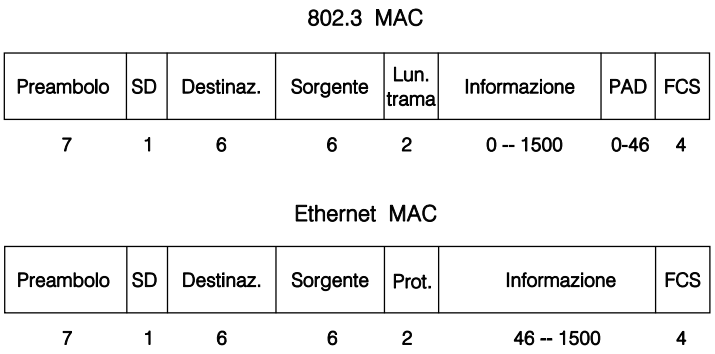

Figura 5.14 *Trama MAC del protocollo IEEE802.3 e del protocollo Ethernet. Le lunghezza dei campi sono espresse in byte.*

tempo di slot vale

$$d = \frac{2 \times 10^5 \times 512}{10^7} \simeq 10 \text{ km}$$

Poichè la durata dello slot è il massimo tempo di andata e ritorno ammesso, ne segue che il diametro massimo della rete è la metà, ossia 5 Km. In pratica il valore viene dimezzato dal momento che i ritardi che giocano non sono solo quelli di propagazione ma anche quelli di attraversamento delel varie unità hardware come i trasmettitori e i ripetitori.

Il formato della trama MAC è mostrato in Figura 5.14 insieme col formato della trama Ethernet da cui lo standard IEEE deriva. Quest'ultimo è tuttora utilizzato, nelle schede standard, protocollo di rete IP, come già spiegato trattando del protocollo LLC.

Il campo SD (Start Delimiter) è una particolare sequenza che viola il codice di linea (Manchester) per indicare in modo inequivocabile l'inizio della trama. Il campo di lunghezza specifica la lunghezza del campo dati mentre il campo PAD viene appeso solo quando occorre portare la lunghezza della trama alla lunghezza minima pari a 64 byte. La massima lunghezza ammessa è di 1518 byte (preambolo e SD esclusi).

Viene inoltre definito un tempo di guardia di 9.6 μs a partire dall'ultima trasmissione osservata, durante la quale il Carrier Sense (CS) continua a segnalare la presenza di portante, ciò per separare i pacchetti sul canale.

Il livello fisico ammette dei *repeaters* (relay di livello 1) che hanno lo scopo di connettere i cablaggi fisici in modo da estendere il raggio della rete fino al limite consentito dallo standard. Oltre a copiare le trame ricevute su una porta su ntutte le altre, i repeaters svolgono anche altre funzioni necessarie al funzionamento trasparente della LAN. Fra queste

- rigenera il preambolo, mangiato durante l'acquisizione del sincronismo, e lo SD. Ciò comporta

l'introduzione di un ritardo;

- propaga le collisioni scoperte su una porta inviando una sequenza di Jam su tutte le altre porte;
- riporta i segnali fisici che non rispettino i vincoli di lunghezza agli stessi vingoli;
- può partizionare porte in cui si presentino troppe collisioni.

Poichè i repeaters non fanno routing e non hanno bisogno di leggere i campi della trama, il ritardo che introducono è molto limitato, per lo più dovuto alla necessità di ricostruire il preambolo di sincronizzazione.

Particolarmente usata è la configurazione fisica su doppino di classe 3 (10BaseT) e configurazione ad albero, dove gli utenti costituiscono le foglie e i repeaters i nodi. I repeaters di livello più basso presentano una porta per ogni utente cui sono collegate. Possono però avere anche porte su coassiale o su fibra. La massima lunghezza del segmento è di 100 m e il numero di repeaters in cascata (livelli dell'albero) sono limitati a quattro. Il cavo tipico è costituito da un cavo UTP a 4 coppie di cui la LAN ne usa solo 2, una per trasmettere e una per ricevere. Con questa configurazione di cablaggio il funzionamento fisico potrebbe anche essere full duplex (trasmissione e ricezione contemporanea), ma il funzionamento del CSMA-CD lo impedisce perchè ci sarebbero comunque collisioni sulle altre porte dei repeater. Dunque il segnale di collisione è comunque attivato quando si riceve un segnale mentre si sta trasmettendo. In realtà, per facilitare l'analisi dei guasti, quando non transitano trame, sui doppini si trasmette comunque un segnale di *idle*. La codifica di linea utilizzata su rame è quella Manchester in cui un bit è rappresentato da una transizione.

Esiste anche la possibilità di usare fibre ottiche (10BaseF) in tre modalità. La prima, 10BaseFP (Fiber Passive), utilizza due collegamenti in fibra per trasmettere e ricevere da una stella ottica passiva. In questa stella convergono i canali di trasmissione e si suddividono quelli di ricezione. Il segmento fra stazione e stella può essere al massimo di 1000 m.

La modalità 10BaseFB (Fiber Backbone) è pensata per collegare repeater, in realtà due half-repeater, in modalità sincrona tramite la quale anche in assenza di dati si trasmette un segnale di idle per sincronizzazione continua e controllo funzionalità. La trasmissione su fibra avviene tramite l'impiego di LED a 850 nm, la fibra è del tipo 62.5/125 e il collegamento massimo è di 2 Km.

La modalità 10BaseFL (Fiber Link) è pensata per collegare indifferentemente stazioni repeater su una tratta di lunghezza massima pari a 2 Km. Il funzionamento logico è simile al caso 10BaseT mentre i segnali e la fibra sono quelli del 10BaseFB.

5.10.2 Fast Ethernet

Recentemente il protocollo 802.3 è stato esteso per ottenere velocità in linea di 100 Mb/s, con i vincoli però che la trama MAC e le regole che la determinano restassero le stesse in modo da facilitare l'interoperabilità fra spezzoni di rete a velocità diverse (802.3u).

Poichè come sappiamo il protocollo d'accesso CSMA-CD impone un rapporto costante fra il tempo di trasmissione della minima trama e il massimo ritardo di propagazione sulla rete, riducendosi il primo per via dell'aumento di velocità di linea occorre ridurre anche il secondo. Dunque, anche con solo uno spezzone di rete a 100 Mb/s, la rete totale non può superare un'estensione massima di circa 250 m, cosa accettabile visto che comunque anche la 10BaseT prevede prevede tratte d'utente non più lunghe di 100 m. La differenza di diametro fra LAN operanti a 10 Mb/s e quelle operanti a 100 Mb/s impedisce che i due tipi interoperino tramite semplici repeaters, che mantengono il *dominio di collisione*.

E' permesso l'uso di repeaters che però non distini più di 10 m.

Naturalmente, una tale variazione di velocità cambia pesantemente lo strato fisico. Questo standard ha due varianti, la prima 100Base4T prevista per funzionare con i vecchi cavi di categoria 3 adatti alla 10BaseT, e la seconda, 100BaseX prevista per cavi di categoria 5 e fibra (100BaseTX e 100BaseFX). Nello standard il livello fisico fisico viene suddiviso in due sottostrati in cui quello più basso, il Physical Medium Dependent (PMD) sublayer cambia con il mezzo usato ma presenta un interfaccia con lo strato superiore indipendente dal mezzo. Il sottostrato più alto utilizza questa interfaccia per trasformare i segnali come quelli che si aspetta il MAC CSMA-CD e viene chiamato Convergence Sublayer.

La variante 100Base4T utilizza tutti e quattro i doppini presenti nel cavo. I primi due sono usati per trasmissione dati in modo unidirezionale (simplex) e per collision detect come nello 10BaseT, mentre gli altri due, non usati nel precedente standard, sono usati insieme per trasmissione dati in modo half- duplex (come il coassiale). Dunque la trasmissione dati avviene su tre coppie in parallelo, riducendo la velocità su ciascuna coppia a 33.3 Mb/s. Tuttavia va cambiato anche lo schema di codifica di linea, dato che col Manchester encoding la velocità risulta ancora troppo alta per doppini di categoria 3. Si ricorre a una codifica di linea chiamata 8B6T, che traduce otto simboli binari in 6 simboli ternari, riducendo la velocità in linea a 25 Msimboli/s. Il meccanismo è alquanto complesso e non viene ulteriormente descritto qui.

Nel 100BaseX, dove il mezzo non è vincolante rispetto alla velocità, la codifica è di tipo 4B5B che trasforma gruppi di 4 bit in codifica a 5 bit.

Ulteriori variazioni introdotte con questo nuovo standard consistono nella possibilità data alle porte 100BaseX, ma anche alle porte 10BaseT, di funzionare in vera modalità full- duplex. naturalmente in questo caso il mezzo condiviso viene ridotto alle sole due porte connesse e la LAN può venire solo estesa con relay di livello 2 (bridge).

5.10.3 Gigabit Ethernet

Alla fine del 1998 è giunto a conclusione il protocollo 802.3z, noto come Gigabit Ethernet, in cui, come dice il nome, le funzionalità dello 802.3 sono state estese alla velocità di 1 Gb/s. Il nuovo standard prevede sia il funzionamento tipico CSMA-CD su mezzi di tipo shared basate su una configurazione a stella su repeater, sia un tipo di funzionamento punto-punto full duplex che non presenta problemi di accesso. Quest'ultimo tipo di funzionamento è previsto per connettere fra loro

entità o relay di livello 2 (bridge). Per ora lo standard prevede il funzionamento su un livello fisico che usa fibre ottiche, ultimodo e monomodo, e cavo schermato. In futuro è prevista l'estensione dello standard a cavo UTP di categoria 5.

Per evitare di ridurre ulteriormente il massimo diametro consentito del dominio di collisione, che mantenendo le regole della rete a 10 e 100 Mb/s si ridurrebbe a 20 m., volendo mantenere le dimensioni della rete a 100 Mb occorrerebbe aumentare di 10 volte la dimensione della trama di lunghezza minima. Ciò complicherebbe la connessione di spezzoni a velocità diversa a livello 2, in quanto i bridge dovrebbero riformattare completamente la trama. Ad evitare ciò si è trovata un'ingegnosa soluzione, ossia la tecnica nota come *carrier extension*.

La soluzione consiste nell'aumentare la dimensione dello slot, ossia la minima lunghezza in cui è presente attività di trasmissione sul canale, da 512 bit a 512 byte, ma mantenendo la minima lunghezza di trama di 512 bit. Se alla fine della trasmissione della trama questa risulta di lunghezza inferiore a 512 byte, si continua la trasmissione con una sequenza particolare di simboli di canale, noti come Carrier Extension (CE) fino a che la minima lunghezza dello slot viene raggiunta.

Naturalmente, ai fini del protocollo d'accesso, è come se la minima trama fosse di 512 byte e dunque se una collisione interviene dopo che la trama effettiva è stata trasmessa, nulla cambia rispetto al caso classico e la trasmissione viene considerata collisa (persa). Allo stesso modo il ricevitore scarta tutti i frammenti ricevuti di durata minore di 512 byte.

Naturalmente, la soluzione trovata non può evitare una forte perdita di efficienza quando le trame da trasmettere siano di lunghezza minima. In questo caso il tempo di trasmissione della trama si riduce di 10 volte, ma quello di trasmissione con la CE si alza di 8 volte, con un aumento di throughput di 1.25 anzichè 10 (riduzione di efficienza di 7/8). In pratica solo un afrazione di trame ha lunghezza minima.

Per aumentare l'efficienza si è pensato di consentire la trasmissione contigua di trame multiple da parte della stessa stazione per alzare per questa via l'efficienza del protocollo. Non è però pensabile di impaccare trame di lunghezza minore dello slot, perchè ciò renderebbe impraticabile il protocollo. Si è scelta allora la soluzione nota come *Frame Bursting* in cui la prima trama viene trasmessa con la carrier extension se inferiore allo slot, mentre le altre trame ammesse al possono essere di lunghezza qualsivoglia ma devono essere separate da un segnale di carrier extension di 96 bit (0.096 μs). In questo modo Le trame successive possono essere trasmesse senza pericolo di collisione e il ricevitore può tranquillamente ricevere trama per trama.

Inizialmente si pensava di ammettere il burst mode fino alla massima dimensione di trama ammessa, pari a 12000 bit (1500 byte). Successivamente, per trarre ulteriore vantaggio in efficienza si è alzato tale limite di 5 volte fino a a 65536 bit (8000 byte). Grazie all'aumento di velocità il massimo tempo di trasmissione del burst risulta comunque inferiore al tempo di trasmissione della massima trama nel caso a 10 Mb/s.

5.10.4 IEEE 802.11

Il comitato IEEE 802 ha anche definito uno standard 802.11 per reti locali di tipo wireless che trasmettono in una gamma attorno ai 2.4 GHz con modulazione di tipo spread spectrum, ma anche in infrarosso, con velocità dell'ordine di 1 e 2 Mb/s. In pratica vengono definiti diversi standard fisici e diversi starti di convergenza che adattano lo strato MAC ai diversi strati fisici.

Il MAC di questo standard è piuttosto complesso in quanto prevede la possibilità che si elegga una stazione come centrale per un meccanismo a polling, sia la possibilità di un meccanismo distribuito di accesso multiplo casuale che si basa sulla tecnica CSMA-CA.

5.10.5 Accesso base ISDN

Descriviamo qui il meccanismo di accesso al canale D dell'interfaccia base ISDN nel punto di riferimento S.

Il MAC utilizzato dai TE per accedere è del tipo CSMA-CD con un meccanismo di priorità che assomiglia a quello descritto nella sezione 5.9.5 ed ha la peculiarità di essere molto efficiente in quanto anche in presenza di conflitti, uno fra i TE contendenti riesce sempre a trasmettere con successo.

Poichè i terminali non si sentono direttamente è la NT a fornire l'eco del canale d'accesso sul canale di ritorno, come è già stato visto a proposito della Figura 1.28 riportanta la trama di multiplazione 2B+D. In questa trama i bit ricevuti dalla NT sul canale D sono trasmessi a distanza tale da consentire che il bit emesso da un terminale sia da questi ricevuto sul canale d'eco prima di emettere il bit successivo. Ciò tien conto del ritardo di due bit causato da ritardo di andata e ritorno (10μ s) alla distanza massima (1 Km).

Il canale d'eco fornisce il segnale equivalente al Carrier Sense. In pratica si è certi che non esiste attività sul canale quando si ricevono almeno 7 UNI consecutivi (che non possono esistere nella trama LAPD).

La somma dei segnali sul canale è bit sincrona e di tipo AND logico (si ricordi che UNO è trasmesso con livello nullo e ZERO con livelli alternati). I TE inattivi trasmettono sempre UNO. I TE attivi confrontano continuamente l'eco e il bit trasmesso. La scoperta di uno ZERO quando si è trasmesso UNO agisce da collision detect e interrompe la trasmissione.

La priorità viene formata dal fatto che un terminale per poter accedere (trasmettere) deve ascoltare un numero di UNI pari a $X > 7$, variabile col terminale secondo le regole spiegate sotto. Più basso è X più alta è la priorità e quando tentano di trasmettere TE con priorità diversa vince quello a priorità più alta. Infatti il primo bit del delimitatore di trama LAPD è uno ZERO (01111110), che, trasmesso dal terminale a priorità più alta viene visto immediatamente da tutti i terminali a priorità più bassa che dunque si astengono dal trasmettere.

Fra terminali con la stessa priorità possono nascere conflitti quando partono scrivendo lo stesso bit.

In base al canale d'eco il conflitto verrà rilevato quando si ha da parte di qualcuno dei contendenti la trasmissione di uno ZERO. In questo caso quelli che hanno trasmesso UNO rilevano la collisione e si astengono. Le trasmissioni continuano finchè rimane uno solo che non si è accorto di nulla e trasmette tutta la trama con successo.

Il vincitore di una contesa viene in pratica determinato dal contenuto del campo di indirizzo (SAPI e TEI). Ad evitare che il vincitore sia sempre lo stesso terminale si introduce un meccanismo di priorità rotante. Si offre però priorità assoluta ai messaggi che portano segnalazione telefonica (SAPI=0).

Ci sono due classi di priorità base e fisse, corrispondenti a $X = 8$ per la segnalazione (SAPI=0) e $X = 10$ per l'altro traffico (SAPI $\neq 0$). All'interno di ciascuna classe si ha una priorità di basso livello data rispettivamente da $X = 9$ e $X = 11$. Le priorità all'interno di ciascuna classe ruotano per assicurare equità nell'accesso all'interno alla stessa classe. Un TEI che ha trasmesso con successo abbassa la sua priorità, mentre la priorità torna normale quando il canale viene sentito libero $X + 1$ volte, ossia conta almeno $X + 1$ UNI. In questo modo, in caso di conflitti all'interno di una classe, tutti hanno la possibilità di trasmettere a turno.

FINE CAPITOLO