

Capitolo 1

L' ACCESSO MULTIPLO

1.1 Il problema

Il problema dell'accesso multiplo è un caso particolare del problema di condivisione **dinamica** delle risorse fra più utenti e trova collocazione nel sottolivello MAC del livello 2 dell'OSI. In particolare, la risorsa da condividere è un canale di banda (velocità) W bit/s e deve essere condiviso fra M stazioni, dove M può anche essere infinito. La peculiarità nasce dal fatto che le stazioni non sono localizzate nello stesso punto e dunque non possono scambiarsi informazioni di coordinamento se non utilizzando il canale stesso.

In questo capitolo affronteremo l'analisi teorica delle problematiche relative all'accesso multiplo e ai protocolli utilizzati o proposti.

Senza perdita di generalità, ci limiteremo a trattare il caso di sistemi a commutazione di pacchetto in cui i pacchetti che devono essere trasmessi sul canale arrivano alle stazioni secondo un processo stazionario, normalmente assunto come Poisson, con frequenza media collettiva λ . I pacchetti attendono di essere trasmessi sul canale in una *Coda Locale*. Ove non altrimenti specificato assumeremo anche che i pacchetti siano di lunghezza costante e indicheremo con T il loro tempo di trasmissione sull'intero canale.

La peculiarità sopra evidenziata è fondamentale, altrimenti l'accesso alla risorsa potrebbe essere facilmente regolato da una fila d'attesa, chiamata *Coda Globale* in cui le richieste di accesso (pacchetti) troverebbero posto e verrebbero gestiti secondo opportune discipline.

Ovviamente, una Coda Globale può sempre essere costituita mediante un opportuno scambio di segnalazione fra le stazioni, che però in generale usa il canale e riduce la banda disponibile. In altri casi ci si limita a gestire le Code Locali senza la pretesa di un controllo generale.

In generale ogni livello L_i introdotto ha un costo traducibile nel fatto che la banda utile W_i , a disposizione del livello superiore, risulta minore di quella disponibile a livello inferiore W_{i-1} .

Definiamo *Efficienza* del livello i il rapporto

$$\eta_i = \frac{W_i}{W_{i-1}}$$

L'efficienza totale è poi il prodotto dell'efficienza dei vari livelli.

Nel seguito chiameremo *Accesso Multiplo Ideale*, quello che riesce a suddividere il canale fra le stazioni come se queste fossero collocate nella stessa località in modo che possano coordinarsi senza limitazioni nella loro attività di accesso con efficienza unitaria.

1.1.1 Notazioni e figure di merito

Lo stato del sistema è descritto dal vettore

$$\mathbf{n} = (n_1, n_2, \dots, n_M)$$

rappresentante la distribuzione dei pacchetti in attesa di trasmissione alle varie stazioni al tempo t . Su questo spazio è definibile il processo $\mathbf{N}(t)$.

Il sistema può avere capacità finita se le file d'attesa alle varie stazioni hanno capacità finita. In caso contrario, il numero di pacchetti in attesa può crescere indefinitamente.

Definiamo *throughput* trasportato dal sistema la quantità d'informazione trasmessa con successo in media in condizioni di stazionarietà. Le condizioni di stazionarietà sopra citate riguardano però il solo canale. Più in particolare si definisce throughput il throughput trasportato in condizioni di stabilità dei ritardi subiti dai pacchetti trasmessi. In presenza di code, ciò richiede che per ogni condizione iniziale $\mathbf{N}(0)$ si abbia

$$\lim_{t \rightarrow \infty} \sum_i N_i(t) < \infty \tag{1.1}$$

Ovviamente, in condizioni di stabilità il throughput coincide con il traffico offerto λ (pacchetti/s). E' invalso l'uso di misurare il throughput come il numero di pacchetti trasmessi con successo in un tempo pari a T (slot) e indicheremo questa misura indifferentemente con $S = \lambda T$ (pkt/slot) o con ρ . Evidentemente il throughput trova un limite massimo nella capacità del canale che in questo caso risulta essere 1 pkt/slot.

Fra le altre cifre di merito che interessano il problema possiamo individuare il *Ritardo di coda locale* W , definito come il tempo che intercorre fra l'arrivo del pacchetto alla stazione e la sua corretta trasmissione. Questo parametro, oltre che del protocollo, è tipicamente funzione del carico.

Un altro parametro, tipico del protocollo, è il *Ritardo d'accesso* D_a , definito come il precedente ritardo nel caso però che il traffico locale sia nullo. Questo parametro, e in particolare il suo andamento in funzione del numero di stazioni nel sistema M , mette in luce il peso del coordinamento fra le stazioni.

1.2 Classificazione dei protocolli di accesso multiplo

I protocolli di accesso multiplo sono costituiti dall'insieme di regole che ciascuna stazione deve seguire per ottenere una *corretta* trasmissione sul canale, ossia una trasmissione che non interferisca con le trasmissioni delle altre stazioni. In sostanza, mediante il Protocollo di accesso, il canale viene dinamicamente suddiviso fra le stazioni che ne hanno necessità.

Diversi tipi di classificazioni sono possibili. Una prima importante suddivisione riguarda il modo di addivenire alle decisioni, che opera la distinzione fra Protocolli **centralizzati** e Protocolli **distribuiti**.

Nei primi esiste un'entità esplicita, che può coincidere con una stazione, che ha il compito di raccogliere le segnalazioni, o richieste, delle varie stazioni, gestirle con procedure di accordamento e restituire permessi di accesso o trasmissione. Lo scambio di segnalazioni fra le stazioni e la stazione centrale può avvenire su canali riservati oppure facendo uso dello stesso canale da condividere. Questi sono protocolli molto flessibili, perchè realizzano praticamente la Coda Globale e permettono una vasta gamma di tipi di gestione delle richieste. Per contro richiedono un'esplicita segnalazione verso e dalla stazione centrale. Altri problemi posti da questa tecnica derivano dal presentare ritardo d'accesso non nullo, specialmente se il ritardo di propagazione non è trascurabile come accade con tratte via satellite, e dal garantire l'affidabilità della stazione centrale, che spesso richiede soluzioni ridondanti.

I Protocolli distribuiti sono intrinsecamente più robusti, ma sono realizzabili semplicemente solo se si rinuncia alla conoscenza completa della situazione e si accettano gestioni che possono non essere le migliori. La segnalazione può essere più schematica e l'efficienza del protocollo più alta.

Un'altro tipo di suddivisione riguarda la specificità o meno del protocollo nei riguardi del canale da utilizzare. Distinguiamo così i protocolli *generali* dai protocolli *specifici*. I protocolli specifici possono sfruttare la peculiarità del canale che utilizzano per semplificare al massimo il tipo di segnalazione e di coordinamento necessario. Possono così essere molto efficienti. Ovviamente il loro campo di applicabilità risulta essere limitato.

Ancora, si possono distinguere Protocolli ad *accesso ordinato*, in cui la sequenza d'accesso al canale da parte delle stazioni avviene seguendo regole deterministiche e prevedibili. Questi, in generale, offrono un ritardo d'accesso crescente col numero di stazioni coinvolte. Per ovviare a questo fatto si sono introdotti i Protocolli ad *accesso casuale* che determinano i diritti d'accesso in base alla scoperta di conflitti che vengono risolti ricorrendo ad esperimenti di natura casuale.

Possiamo poi distinguere i protocolli in base alle tre famiglie di canali tipici offerti dallo strato sottostante e già visti in quanto precede: i *canali sequenziali*, i *canali broadcast* e i *canali broadcast centrali*.

1.3 Polling

Col nome di Polling si indica un insieme di protocolli ad accesso ordinato in cui le stazioni ricevono, in successione ciclica, un messaggio *di polling* che rappresenta l'abilitazione a trasmettere sul canale.

Alla fine della trasmissione la stazione segnala il rilascio del canale di modo che il messaggio di polling possa essere inviato ad altra stazione.

La procedura di polling costituisce un prezzo che riduce la banda disponibile per i messaggi e che varia con le diverse possibili tipologie.

Il Polling può essere centralizzato ed è chiamato *Roll- Call Polling*. In questo caso il messaggio di polling è sempre rilasciato da una stazione centrale che sovrintende all'accesso. In generale, il messaggio è un vero e proprio pacchetto esplicito, con indirizzo della stazione di destinazione. Se la stazione non risponde, il messaggio viene mandato alla successiva e così di seguito.

Parzialmente centralizzato è anche lo *Hub Polling*, in cui la stazione centrale (Hub) ha il compito di iniziare un ciclo rilasciando il messaggio di polling alla prima stazione. Da questa, il messaggio di polling viene passato in sequenza alle altre stazioni e alla fine viene ritornato alla stazione centrale. Rispetto alla tecnica precedente si risparmia il passaggio del controllo da una stazione a quella centrale e poi ancora alla stazione successiva. Inoltre, se il canale è sequenziale, ossia prevede già un'ordinamento fisico fra le stazioni, allora il passaggio del controllo può fare a meno di essere esplicitamente indirizzato. Anzi basta un solo simbolo logico, spesso chiamato *control token*, o più semplicemente token, che venga fermato dalla stazione che ne ha bisogno e poi rilasciato verso la successiva.

Se il sistema non prevede una stazione master che gestisca il token, allora il sistema diventa del tutto distribuito, e prende il nome di sistema a token. Tuttavia, se questo migliora le caratteristiche e le prestazioni del protocollo, complica le procedure di gestione del token, quali la partenza del token, il controllo della sua integrità eccetera. Occorre che vengano distribuite anche queste funzioni, cosa che rende il meccanismo non semplice.

Fra le caratteristiche logiche dei sistemi di polling una molto importante riguarda il tempo di servizio ossia il tempo consecutivo per cui una stazione può tenere il token prima di rilasciarlo. Se non esistono limiti a tale tempo allora si parla di *servizio esaustivo*. Altrimenti si parla di *servizio limitato* e *k-limitato* se si consente la trasmissione consecutiva di k pacchetti consecutivi prima di rilasciare il controllo. Ancora si dice che il servizio è *gated*, se possono essere trasmessi i soli pacchetti già presenti alla stazione prima dell'arrivo del token.

Nelle prossime sezioni indagheremo le prestazioni dei servizi di polling in alcune situazioni simmetriche in cui tutte le stazioni offrono lo stesso traffico λ/M e il tempo di propagazione del token da una stazione all'altra sia uguale per tutte le stazioni. Indicheremo poi con h l'overhead dovuto alla somma dei tempi di propagazione del token da una stazione all'altra e del tempo di trasmissione del token stesso (noto anche come *walking time*).

1.3.1 Polling esaustivo

In Figura (1.1) è riportata l'attività del servente (canale). Tale attività è una successione di treni di trasmissioni da ciascuna stazione (servente busy) separati da intervalli di silenzio (servente idle) di durata multipla di h che rappresentano il walking time del passaggio di controllo. Una grossa complicazione è rappresentata dal fatto che i vari cicli di trasmissione in cui il processo può essere suddiviso, assumendo come punti di suddivisione l'arrivo (o la partenza) del token a una data stazione, non sono indipendenti (i punti indicati non sono di rigenerazione). Certo

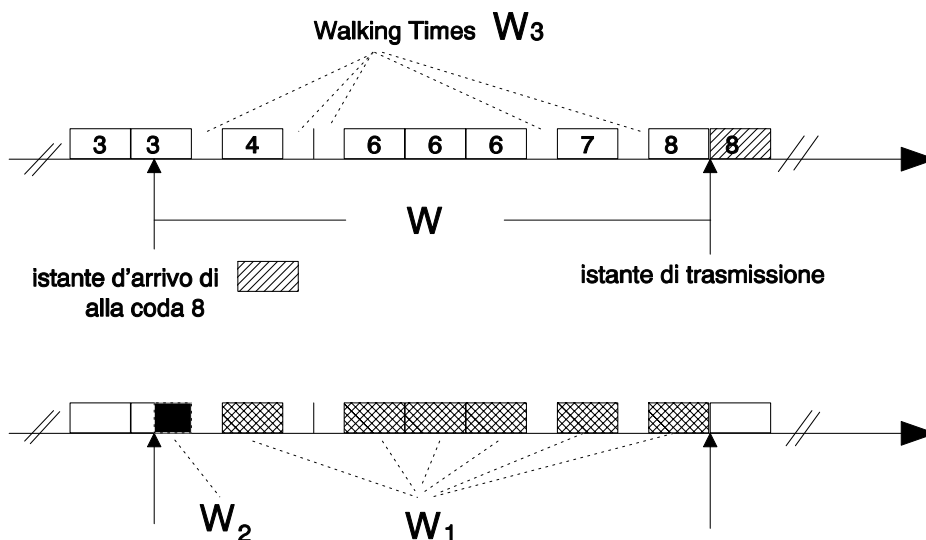


Figura 1.1: Esempio di occupazione del canale (servente) nel polling esaustivo. Sono evidenziate le trasmissioni delle stazioni che si succedono dalla 3 all 8.

il processo è rigenerativo, essendo istanti di rigenerazione gli istanti che iniziano il busy period globale, intendendo per tale periodo il periodo in cui o il servente è occupato in trasmissione o deve raggiungere code non vuote.

La probabilità di trovare il servente occupato in trasmissione (busy) è pari a $\rho = \lambda m_x$, come se gli arrivi fossero tutti a un'unica coda. La probabilità di cadere in un busy period globale non è però ρ perchè tale periodo contiene anche i walking times ed è più complicata da calcolare perchè non conosciamo come si inserisce e ripartisce il walking time fra i vari messaggi trasmessi in tale periodo.

Il tempo medio d'attesa in coda $E[W]$ può essere facilmente ricavato in un modo analogo a quanto fatto per la formula di Pollaczec-Khinchin. I termini sono però aumentati. Il primo termine è il tempo $E[N_c]m_x$ per trasmettere tutti i messaggi nelle code (in numero di $E[N_c]$) prima del messaggio considerato ed è rappresentato in Figura (1.1) dalla componente W_1 . Il secondo termine riguarda il tempo che si deve attendere prima che una coda venga esplorata di nuovo per prelevare il prossimo messaggio e vale $\rho E[Z] + (1 - \rho)h/2$ (nella Figura (1.1) si è evidenziato il caso in cui l'arrivo occorre durante una trasmissione ed è $W_2 = Z$). Il terzo (W_3) è il tempo medio di ritorno del controllo. Questo è zero se il controllo sta arrivando o è arrivato alla stazione sotto esame, ed è $(M - 1)h$ se il controllo è appena partito dalla stazione sotto esame (si noti che il walking time in arrivo è già stato tenuto in conto in W_2). Mediamente dunque si ha $W_3 = (M - 1)h/2$ e:

$$E[W] = E[N_c]m_x + \rho E[Z] + (1 - \rho)\frac{h}{2} + \frac{M - 1}{2}h \tag{1.2}$$

Utilizzando ancora il fatto che $E[N_c] = \lambda E[W]$ e risolvendo, si ottiene nel caso di tempo di trasmissione costante $m_x = T$

$$E[W] = \frac{\rho}{2(1 - \rho)}T + \frac{M - \rho}{2(1 - \rho)}h \tag{1.3}$$

Il primo termine della (1.3) è identico a quello che si ha nel caso di coda unica. Il secondo cresce al crescere di M e tien conto del fatto che, a parità di traffico globale, al crescere di M cresce la parte di traffico che arriva alle code precedenti quella in oggetto e che il token deve visitare ancora. Questo traffico è costituito da messaggi che arrivano dopo ma vengono trasmessi prima di quello in oggetto. Inoltre, per $\rho = 0$ la seconda parte non è nulla ma è uguale al minimo tempo d'attesa dovuto alla circolazione del token.

Dalla (1.3) si ottiene l'occupazione media di ogni coda locale come

$$E[N_l] = \frac{\lambda E[W]}{M}$$

E' interessante anche vedere cosa succede del tempo di ciclo C . La durata media di un ciclo oltre al tempo di trasmissione del token e di propagazione è formata anche dal tempo di trasmissione dei pacchetti arrivati a ciascuna stazione durante un ciclo stesso e che mediamente sono $\lambda E[C]$. In condizioni di equilibrio si ha così

$$E[C] = Mh + \lambda E[C]m_x \quad (1.4)$$

che fornisce

$$E[C] = \frac{Mh}{1 - \rho} \quad (1.5)$$

con, al solito, $\rho = \lambda m_x$. Si può poi mostrare, ma la dimostrazione viene omessa, che nel caso di tempo di servizio costante uguale a T , la varianza del tempo di ciclo vale

$$VAR[C] = \frac{M^2 \lambda h T}{(1 - \rho)(1 - \rho^2)} \quad (1.6)$$

La (1.5) può anche essere ricavata osservando che si può porre $E[C] = B_c + Mh$ dove con B_c si è indicato il tempo di trasmissione in un ciclo. Poichè l'ergodicità assicura che

$$\rho = \frac{B_c}{B_c + Mh}$$

risolvendo per B_c si ha

$$B_c = \frac{\rho}{1 - \rho} Mh \quad (1.7)$$

mentre il numero di messaggi trasmessi in ogni ciclo è

$$Q_c = B_c/T = \frac{\lambda}{1 - \rho} Mh \quad (1.8)$$

Si noti che tutte le grandezze sopra viste tendono a zero al tendere di h a zero. Ciò si spiega con l'operazione di media sul ciclo. Infatti, in un periodo di tempo finito e con $\rho < 1$ il numero medio di cicli tende all'infinito per $h \rightarrow 0$ e solo un numero finito ha lunghezza non nulla. Le relazioni sopra mostrano anche che per $\rho \rightarrow 1$ la lunghezza del ciclo e tutte le grandezze cicliche sopra divergono e il token tende a tornare dopo un tempo medio infinito.

In caso di traffico sbilanciato, per il quale si possono ancora scrivere le prestazioni in forma chiusa, se una sola stazione è attiva, questa può accaparrarsi tutta la banda, e questa è una proprietà positiva. Tuttavia, in presenza di traffico diverso in altre stazioni, si vede che il maggior traffico delle une influenza il tempo d'attesa delle altre, che può facilmente crescere senza limiti. Questa è una caratteristica negativa che fa preferire i sistemi con servizio limitato, in cui il controllo viene rilasciato forzatamente se i traffico di una stazione è troppo elevato.

Nel caso di servizio gated, in cui il pacchetto non viene trasmesso se arriva quando il token è presente alla stazione, e ciò capita con probabilità ρ/M , un termine pari a

$$\frac{\rho}{M}Mh = h\rho$$

va aggiunto alla (1.2) e il nuovo ritardo diventa:

$$E[W] = \frac{\rho}{(1-\rho)}T + \frac{M+\rho}{2(1-\rho)}h \quad (1.9)$$

1.3.2 Polling a servizio limitato

Questa tecnica, forza una stazione a cedere il controllo dopo aver trasmesso k pacchetti consecutivamente. E' generalmente considerata più onesta del servizio esaustivo perchè limita l'effetto, sul ritardo sperimentato in una stazione, del traffico delle altre. In particolare, noi consideriamo il caso in cui il servizio è limitato a un solo pacchetto. In questo caso la procedura di servizio è anche nota come ROUND ROBIN.

Il calcolo è leggermente più complicato che nel caso precedente perchè si complica il calcolo del termine relativo al passaggio del token perchè se ci sono più pacchetti in coda, il pacchetto considerato deve aspettare più di un ciclo di token e un altro ciclo deve aspettarlo se arriva durante un servizio della propria stazione. Poichè in media di pacchetti ce ne sono $E[N_c]/M = \lambda E[W]/M$, il pacchetto dovrà aspettare:

$$E[W] = \rho E[W] + \rho E[Z] + (1-\rho)\frac{h}{2} + \frac{M-1}{2}h + (\lambda E[W]/M + \rho/M)Mh \quad (1.10)$$

che porta a

$$E[W] = \frac{\rho}{2(1-\rho\frac{h+T}{T})}T + \frac{m+\rho}{2(1-\rho\frac{h+T}{T})}h \quad (1.11)$$

La (1.11) mostra che a parità di condizioni, il ritardo è peggiore che nel caso di servizio esaustivo o gated. Anzi, addirittura il throughput massimo è inferiore. Il ρ infatti non può crescere oltre $T/(h+T)$. Ciò è dovuto al fatto che il ciclo non può più crescere indefinitamente e il tempo perso nel walking time non può essere diluito su un ciclo infinito.

Tuttavia, la tecnica Round Robin riscuote generale successo perchè molto migliore nel caso di traffico sbilanciato. Si immagini infatti il caso che in tutti i sistemi visti si introduca una nuova stazione con un messaggio da trasmettere. In caso di utilizzo molto elevato il tempo all'arrivo del token può essere molto alto in caso di servizio esaustivo, mentre è comunque limitato al massimo a MT nel caso Round Robin. In generale questa tecnica tratta meglio, in termini di ritardo, stazioni che hanno minor traffico e ciò è considerato più opportuno.

Per quanto riguarda il tempo medio di ciclo, il ragionamento fatto per ottenere la (1.5) continua a valere e così pure la (1.5) stessa. Ovviamente però cambia la varianza.

In generale, se N fra le M stazioni hanno sempre pacchetti da trasmettere, il throughput massimo, ossia l'efficienza del sistema è data da

$$\eta(N) = \frac{NT}{NT + Mh} = \frac{1}{1 + \frac{Mh}{NT}} \quad (1.12)$$

ottenuta come rapporto fra il tempo utile e il tempo di ciclo. Dunque l'efficienza cambia col numero di utenti attivi e cresce con questi. Essa dipende fortemente da h che va contenuto il più possibile. Il fattore h/T è chiamato fattore di inefficienza. E' da notare che il massimo throughput η si riduce al decrescere di N a causa dell'overhead. Comunque, una stazione sola attiva può utilizzare molto di più di quando ne sono attive N , risultando il rapporto pari a

$$\frac{\eta(1)}{\eta(N)/N} = \frac{N + M(h/T)}{1 + M(h/T)}$$

1.4 Reti locali di tipo Token

In questa sezione illustriamo alcuni sistemi a token, esistenti o esistiti come sperimentazioni o proposte. Le valutazioni sono effettuate nell'ipotesi che la tecnica applicata sia di tipo Round Robin.

1.4.1 Token Bus

La rete locale IEEE 802.4 è un esempio di sistema cosiddetto a Token Bus. Come mezzo trasmissivo assume un bus a 10Mb/s di topologia alquanto libera perchè può essere formata da diversi tronchi uniti a formare una topologia ad albero.

Il Token è formato da un pacchetto con mittente e destinatario, oltre che il vero e proprio flag di token. Le stazioni sono inserite in un circuito logico che costituisce il percorso logico del token. Oltre

al protocollo vero e proprio, che consente la trasmissione di un messaggio di lunghezza variabile ma limitato superiormente, esistono delle procedure distribuite per inizializzare la rete, generare e controllare il token, inserire e disinserire stazioni nel circuito logico.

L'overhead h è costituito dal tempo di propagazione τ_b e dal tempo di trasmissione del token P_b . Nel caso conservativo in cui la rete sia di tipo stellare, con tempo di propagazione diametrale pari a τ_b l'efficienza con $N \leq M$ stazioni attive è

$$\eta(N) = \frac{1}{1 + \frac{M \tau_b + P_b}{N T}}$$

Si vede come questa tecnica poco si presti a cambiamenti di scala per cui il fattore τ_b/T diventi importante.

1.4.2 Single Token Ring

Nei canali di tipo ring, uno dei problemi è la rimozione dei segnali immessi e dei segnali che eventualmente si trovano a circolare per errore. Una delle procedure più sicure anche se meno efficienti, è quella di fare estrarre un segnale immesso alla stazione che l'ha generato.

Con il metodo detto *single token*, quando una stazione riceve il token da quella che la precede sul ring, apre l'anello, trasmette il suo messaggio, drenando nel frattempo tutto quello che riceve dal ring finchè riceve anche il proprio messaggio. Quando questo è stato estratto in modo totale dal ring, questo viene richiuso e il token trasmesso a valle.

La procedura di cui sopra è poco efficiente nel caso in cui il tempo di propagazione sul ring non è trascurabile rispetto al tempo di trasmissione dei messaggi, perchè, in questo caso, più messaggi potrebbero trovare posto contemporaneamente sullo stesso ring aumentando l'efficienza.

Comunque, questa tecnica consente di ritardare al minimo il segnale transitante e cioè di un tempo di bit, necessario a riconoscere lo stato del flag di token e ad alterarlo, (porlo occupato) in modo che la stazione a valle non lo possa usare. Il flag di token viene contenuto in un pacchetto (trama) particolare che viene riconosciuto dalle stazioni. Tale pacchetto viene drenato da una stazione che trasmette e viene ritrasmesso con le modalità più sopra specificate.

Il calcolo dell'efficienza differisce dal caso generale di polling perchè le stazioni che non usano il token non lo tolgono e non lo ritrasmettono, in quanto il token passa direttamente alla successiva senza bisogno di ritrasmissione. Questo offre ovviamente un vantaggio rispetto al caso precedente a parità degli altri parametri.

Detto τ_r il tempo di propagazione sul ring (praticamente il tempo di propagazione se si trascura il bit di ritardo nelle stazioni (latency time)), il tempo di passaggio del token, da una stazione che trasmette alla successiva, risulta essere pari a

$$h + T = \tau_r/M + P_r + \tau_r + T$$

dove i primi due termini rappresentano il tempo di propagazione e di trasmissione del token, mentre gli altri due rappresentano il tempo di propagazione e di trasmissione del messaggio. Il tempo di passaggio del token, da una stazione che non trasmette alla successiva è

$$h' = \tau_r/M$$

L'efficienza di questa tecnica quando $N \leq M$ stazioni trasmettono è dunque:

$$\eta(N) = \frac{1}{1 + \frac{\tau_r}{NT} + \frac{\tau_r + P_r}{T}}$$

Si vede che il termine dominante è ancora del tipo di quello che domina il token bus. Ciò a causa del giro completo che deve fare il messaggio prima che il token possa essere reinserito. Una variante migliore, nel caso sia $T \leq \tau_r$ consiste nel trasmettere il token appena si incomincia a ricevere il messaggio trasmesso. In questo caso si risparmia un tempo T nel reinviare il token e l'efficienza risulta

$$\eta(N) = \frac{1}{1 + \frac{\tau_r}{NT} + \frac{(\tau_r - T) + P_r}{T}} \quad T \leq \tau_r$$

un po' migliore dell'efficienza del token bus nelle medesime condizioni.

1.4.3 Multiple Token Ring

Come ulteriore variante del caso precedente, il messaggio contenente il token viene ritrasmesso, da una stazione che l'ha fermato, in coda al messaggio trasmesso. L'aggettivo *multiple* non si riferisce al token ma alla possibilità che possano esserci più messaggi transitanti in successione sul ring. Dunque risulta

$$h = \tau_r/M + P_r$$

e l'efficienza di questa tecnica risulta

$$\eta(N) = \frac{1}{1 + \frac{\tau_r}{NT} + \frac{P_r}{T}}$$

Considerando che può essere $P_r \ll P_b$ e che si può avere $\tau_r \simeq \tau_b$ si vede che la tecnica Token Ring può essere molto più efficiente sia del Token Bus (l'overhead dovuto al tempo di propagazione è ripartito fra le stazioni) che ovviamente del Single Token Ring. In ogni caso, anche questa tecnica paga il suo contributo al tempo di propagazione, che può diventare dominante in sistemi molto veloci e grandi.

1.4.4 Expressnet

Si tratta di un bus unidirezionale ripiegato, con una tratta di trasmissione e una di ricezione parallela alla prima. Nella tratta di trasmissione la stazione può anche sentire il canale onde evitare di trasmettere se c'è un segnale in arrivo.

La tecnica di accesso è nota come *token implicito* poichè le funzioni del token sono sostituite dal fronte di discesa della trasmissione precedente, sentito sul canale di trasmissione. In questo modo, una stazione si accorge della fine trasmissione della precedente con un ritardo trascurabile, dell'ordine di un bit, e può *appendere* il proprio pacchetto in coda al *treno* di pacchetti transiente. Può accadere che, ancora con un bit di ritardo, ci si accorga di un'altra trasmissione che arriva da monte. In questo caso la trasmissione viene abortita, non senza però aver provocato una sovrapposizione per un tempo di bit, che va previsto come tempo di guardia.

Il sistema è del tipo aperto. I pacchetti si esauriscono alla fine del bus senza il bisogno di essere tolti. Però, quando hanno trasmesso tutti, il ciclo va fatto ripartire. Questo può essere fatto in modo semplice e distribuito da ogni stazione quando si osserva, sul canale di ricezione, un silenzio di almeno due bit. Il parallelismo dei due canali assicura che sarà solo la stazione più a monte a trasmettere per prima. Per inizializzare il sistema basta trasmettere un segnale qualsivoglia per un tempo sufficiente a riempire tutto il bus. Il fronte di discesa fa ripartire tutto. Ad evitare inutili ripartenze, le stazioni che non hanno pacchetti tentano di trasmettere, per iniziare il ciclo, un piccolo segnale (Locomotiva). Solo la stazione accesa più a monte ci riesce e fa ripartire il ciclo.

Il meccanismo è del tipo multiple token con la differenza che il token non viene mai trasmesso e dunque l'efficienza di questa tecnica risulta

$$\eta(N) = \frac{1}{1 + \frac{\tau_r}{NT}}$$

Si è usato lo stesso tempo di propagazione τ_r del ring perchè la topologia necessaria può essere configurata in un ring. Si vede anche qui che quando il tempo di propagazione diventa dominante rispetto a MT allora il sistema perde in efficienza.