

Capitolo 3

Data Link: MAC

3.1 Protocolli ad accesso multiplo

3.1.1 ALOHA

ALOHA puro

Il protocollo ALOHA è un sistema a contesa ideato per trasmettere via radio i segnali in broadcast. Ogni stazione trasmette ogniqualvolta ha bisogno di farlo, senza verificare se il canale è libero.

Dopo aver trasmesso, resta in ascolto per notare se si sono verificate collisioni e, in caso, riprova dopo un tempo casuale. Solitamente si attua utilizzando un meccanismo di **back-off**, secondo il quale la ritrasmissione avviene dopo un ritardo selezionato casualmente compreso tra 0 e $(K - 1)T$, dove T è il tempo di trasmissione del frame e K può eventualmente dipendere dal numero di collisioni già avvenute.

Si possono trasmettere con successo Ge^{2G} frame per unità di tempo, dove G indica il numero di trasmissioni medie per unità di tempo, raggiungendo un **utilizzo della banda pari al 18,4 percento**.

Slotted ALOHA

Nel protocollo Slotted ALOHA, il tempo viene diviso in **intervalli discreti detti slot**, cadenzati da una stazione principale. Tutte le stazioni possono trasmettere solamente all'inizio dello slot, dimezzando così le possibilità di collisione e raddoppiando l'utilizzo della banda.

In altre parole, In ALOHA un frame poteva collidere anche solo parzialmente, con un frame che era già stato inviato, o viceversa, con un frame appena inviato da un'altra stazione. Di conseguenza il periodo in cui il frame poteva collidere era 2 volte il suo tempo di vita. Con la versione slotted il periodo critico viene dimezzato, visto che **una collisione può verificarsi solo quando due frame vengono inviati contemporaneamente nello stesso slot**.

3.1.2 Protocolli Carrier Sense

Sono protocolli multi-accesso in cui ogni stazione è consapevole di condividere il canale con altre stazioni.

CSMA 1-persistente

Quando una stazione deve trasmettere, prima di farlo, **ascolta il canale per sapere se è già occupato**. In questo caso la stazione resta in attesa finché non si libera e poi trasmette. Se è libero trasmette i propri dati *senza esitazione*. Se avviene una collisione la stazione aspetta un tempo casuale e poi ricontrolla il canale. Il protocollo è chiamato *1-persistente* perché la stazione trasmette con probabilità 1 quando trova il canale libero.

Con questo protocollo si ottiene un **data rate superiore al 50 per cento** anche se ci sono due problemi critici:

1. non viene considerato il tempo di propagazione del segnale: se due stazioni si trovano lontane tra loro e una inizia a trasmettere, prima che l'altra veda il canale occupato passa del tempo durante il quale vede il canale ancora libero.
2. se due stazioni che vogliono trasmettere trovano il canale occupato, appena si libera trasmettono contemporaneamente, generando una collisione (corretto con le versioni p-persistente e non persistente).

CSMA non persistente

In questo protocollo **si tenta consapevolmente di essere meno ingordi**. Prima di trasmettere, ogni stazione controlla il canale: se nessun altro sta trasmettendo inizi a inviare i dati ma, se il canale è occupato, **la stazione non esegue un controllo continuo per impossessarsene subito alla fine della trasmissione, bensì attende per un intervallo casuale** prima di ripetere l'algoritmo. Di conseguenza, questo meccanismo permette di utilizzare meglio il canale, ma allunga i ritardi rispetto a CSMA 1-persistente.

CSMA p-persistente

Si applica ai canali divisi in intervalli temporali. Quando è pronta a trasmettere, ogni stazione controlla il canale. Se lo trova libero, trasmette subito con probabilità p e rimanda la trasmissione all'intervallo successivo con probabilità $q = 1 - p$. Il processo si ripete fino a quando il frame non è stato trasmesso o un'altra stazione ha iniziato a trasmettere. In quest'ultimo caso, la stazione sfortunata si comporterebbe come se ci fosse stata una collisione. Se la stazione inizialmente avesse trovato il canale occupato, avrebbe atteso fino all'intervallo successivo per poi applicare l'algoritmo descritto in precedenza.

CSMA con rilevamento delle collisioni

Questo protocollo, CSMA/CD (*CSMA with collision detection*), è alla base delle classiche LAN Ethernet. Le stazioni **ascoltano il canale durante la trasmissione e, se rilevano una collisione, interrompono bruscamente l'operazione** dal momento che i frame inviati sono comunque rovinati. Questa strategia permette di **risparmiare tempo e banda**.

È importante capire che l'individuazione delle collisioni è un processo analogico. L'hardware della stazione deve ascoltare il canale durante la trasmissione; se il segnale letto è diverso da quello inviato, sa che sta avvenendo una collisione. Se una stazione rivela una collisione, interrompe la trasmissione, aspetta per una quantità di tempo casuale e quindi prova di nuovo.

3.1.3 Protocolli senza collisione

Basic Bitmap

Ogni periodo di contesa, *contention period*, è diviso in N intervalli (*slot*). Se la stazione 0 deve inviare un frame, trasmette un bit 1 durante l'intervallo 0. In generale, quindi, **la stazione i -esima, quando ha un frame da inviare, invia un 1 nell'intervallo i -esimo**. Una volta trascorsi gli N Intervalli, tutti sanno chi deve trasmettere e in ordine iniziano a farlo. Una volta finito il giro di trasmissioni, si ricomincia da capo.

Nonostante la mancanza intrinseca di collisioni, questo protocollo presenta comunque dei difetti. **Se ci sono troppe stazioni il contention period diventa troppo lungo** causando uno spreco di banda, specialmente se solo poche stazioni vogliono trasmettere. Inoltre, se una stazione ha bisogno di trasmettere, però il suo slot per prenotarsi è già passato, deve rimanere inattiva fino al prossimo giro di trasmissioni.

Binary countdown

Ogni stazione intenzionata a utilizzare il canale **comunica in broadcast il proprio indirizzo sotto forma di stringa binaria**, partendo dal bit più significativo. In questo modo nel canale è presente l'OR degli indirizzi di tutte le stazioni che vogliono trasmettere. Successivamente, per decidere chi può trasmettere, **si inizia a guardare la stringa presente nel canale a partire dal bit più significativo**: se è impostato a 1, vengono scartate tutte le stazioni con quel bit pari a 0 e si prosegue analizzando il secondo bit della stringa. **Vince la stazione che rimane, ossia quella che ha l'indirizzo più alto**. Esempio: se le stazioni 0010, 0100, 1001 e 1010 vogliono trasmettere, vincerà 1010. Al termine della trasmissione, ricomincia un nuovo turno.

È abbastanza evidente la proprietà per cui **le stazioni con il numero più alto hanno una priorità maggiore** rispetto alle stazioni con il numero più basso. Per evitarlo, si possono usare due strategie:

- una strategia in stile ALOHA, nella quale una stazione **dopo aver trasmesso aspetta un tempo casuale** prima di ritrasmettere. Comporta però un difetto: il canale potrebbe rimanere inattivo se ci sono poche stazioni che trasmettono.
- quando una stazione ha ottenuto il diritto di trasmettere **si porta a priorità (indirizzo) 0**, mentre tutte le altre che non hanno trasmesso aumentano la propria priorità di 1.

Con questo metodo l'efficienza del canale è pari a:

$$\frac{d}{d + \log_2 N}$$

ma se il formato del frame viene scelto con cura in modo che l'indirizzo del mittente costituisca il primo campo del frame, si recuperano anche questi $\log_2 N$ bit e l'efficienza raggiunge il 100%.

3.1.4 Protocolli a contesa limitata

Adaptive Tree Walk Protocol

Protocollo di trasmissione che combina la strategia Collision Detection con quella Collision Avoidance. Le stazioni vengono viste come le foglie di un albero binario. Ad ogni slot di trasmissione tutti quelle che devono trasmettere trasmettono. Se si verifica una collisione vengono creati due slot, uno per il sottoalbero destro e uno per quello sinistro. Se in uno dei due slot si verifica una collisione si ripete lo stesso procedimento finché non tutti non riescono a trasmettere.

Con questa tecnica si riescono ad individuare le stazioni che vogliono trasmettere con al più $\log_2 N$ tentativi. Nel lato pratico questa tecnica può essere ancora migliorata: è possibile infatti tenere traccia delle stazioni che hanno trasmesso recentemente ed iniziare ad esaminare l'albero a partire da una certa profondità. Se ci sono A stazioni attive si può iniziare ad esaminare l'albero dai nodi di profondità $P = \log_2 A$.

3.2 Protocolli per LAN Wireless

3.2.1 Stazione Nascosta

Si verifica quando **una stazione non riesce a rilevare, a causa della distanza, altri concorrenti per il mezzo trasmissivo**. Il problema della stazione nascosta si verifica in questa situazione:

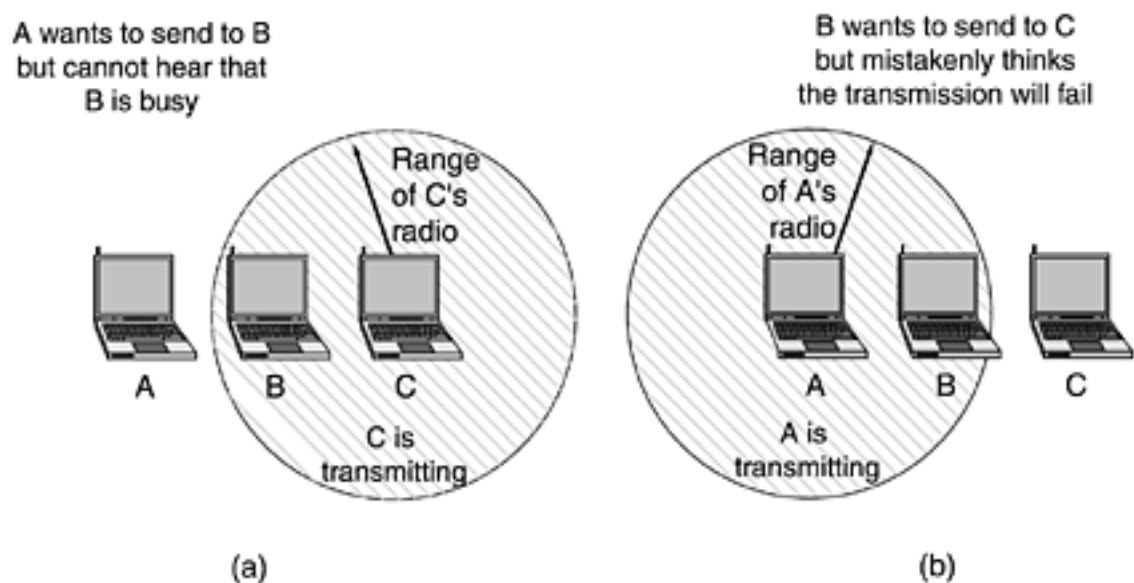


Figura 3.1: Esempio stazioni Wireless

1. A sta trasmettendo dei frame a B;
2. anche C vuole trasmettere a B;
3. C controlla il canale, non sente la trasmissione di A e quindi inizia a trasmettere a B.

Le due trasmissioni interferiscono tra loro portando alla perdita di entrambi i frame.

3.2.2 Stazione Esposta

Si verifica quando una stazione **non trasmette perché sente il canale occupato, anche se la sua trasmissione non interferirebbe** con la trasmissione già in corso.

1. B sta inviando dei frame ad A
2. C vuole trasmettere verso D, quindi ascolta il canale, lo trova occupato e non trasmette, anche se una trasmissione da C verso D non interferirebbe con la trasmissione di B.

3.2.3 MACA e MACAW

MACA è un protocollo di accesso al canale che si basa sull'uso di **due particolari frame**: RTS (*Request To Send*) e CTS (*Clear To Send*). Entrambi i frame contengono la dimensione del frame dati che sarà scambiato, in modo

che le altre stazioni possano stimare la durata della trasmissione. Quando una **stazione A vuole trasmettere ad una stazione B**, invia il frame RTS. Se B accetta la trasmissione, invia ad A il frame CTS, dopodiché la trasmissione può iniziare.

Le stazioni che sentono solo il frame RTS devono rimanere in silenzio per consentire ad A di sentire il CTS. Le stazioni che sentono solo il CTS o entrambi restano in silenzio per evitare interferenze.

La versione successiva (MACAW) introduce il meccanismo di ACK per il frame dati e un altro frame di controllo chiamato DS contenente la **dimensione del frame dati** che si sta per trasmettere. In questo modo se delle stazioni volevano trasmettere ad una stazione che sta già trasmettendo sanno tra quanto la stazione sarà libera.

3.3 Ethernet

3.3.1 Codifica di Manchester

Questo tipo di codifica nasce dalla **necessità di dover determinare senza ambiguità il punto iniziale, centrale e finale di ogni bit**, senza impulsi esterni. Si divide **il periodo di un bit in due intervalli uguali**: se si deve inviare 1 si tiene un **livello di tensione alto nel primo intervallo e basso nel secondo**. Viceversa se si deve inviare uno 0.

In questo modo **è più semplice la sincronizzazione** tra mittente e destinatario anche con lunghe sequenze di 0 o 1. Di contro, la **banda viene dimezzata**. Esiste una variante di questa codifica detta differenziale, che si basa sul cambio di transizione tra intervalli. Se è uno 0 avviene il cambio di transizione, altrimenti no. È più immune al rumore anche se più complessa.

3.3.2 Binary Exponential Backoff

È un algoritmo, usato in CSMA/CD e quindi in Ethernet, **per stabilire il tempo di attesa casuale una volta riscontrata una collisione**.

1. Si calcola T_p , il tempo di propagazione di andata e ritorno del frame nel caso peggiore (*slot time*);
2. si divide il tempo in intervalli P che durano T_p ciascuno;
3. stabilito un contatore $i = 1$, finché la trasmissione non avviene con successo:
 - (a) si sceglie casualmente un numero N tale che $0 \leq N < 2^i$;
 - (b) si attendono $N \times P$ intervalli prima di ritentare la trasmissione;
 - (c) si incrementa i .

Si noti che N **cresce esponenzialmente** con il numero di collisioni avvenute: questo garantisce che il tempo di attesa sia minimo se ci sono poche stazioni a collidere e che invece possa essere molto alto nel caso ci siano molte più stazioni. Esiste anche una versione *truncated*, in cui si fissa un **valore massimo per il contatore** i dopo il quale, in caso di ritrasmissione fallita, l'operazione viene annullata e ai livelli più alti viene restituito un errore. Ad esempio, se i può valere al massimo 10, il massimo tempo di attesa possibile equivale a 1023 slot time.

3.3.3 Frame Ethernet

Vengono di seguito illustrati i campi dati di un frame Ethernet secondo lo standard **DIX**, leggermente diverso dallo standard IEEE 802.3.

8 Byte	6 Byte	6 Byte	2 Byte	0 - 1500 Byte	0 - 46 Byte	4 Byte
Preambolo	Indirizzo Destinatario	Indirizzo Mittente	Type	Data	Pad	Checksum

Figura 3.2: Frame Ethernet

Preambolo Composto da 8 byte, ognuno dei quali contiene la sequenza di bit 10101010. Permette al clock del ricevente di sincronizzarsi con quello del trasmittente. In IEEE 802.3 i due bit meno significativi dell'ultimo byte formano un campo a sé stante: SoF, *Start Of Frame delimiter*, che avvisa il ricevente dell'immediata fine del campo Preambolo.

Indirizzo Destinatario/Mittente Indica gli indirizzi MAC di mittente e destinatario. Nel campo destinatario, se il bit più significativo è impostato a 1 significa che la trasmissione è diretta a un gruppo di stazioni (multicast). Se tutti i bit sono impostati 1, la trasmissione è di tipo broadcast.

Type Banalmente, indica al ricevente cosa fare con il frame arrivato, cioè a quale livello di rete passarlo. Nello standard IEEE 802.3 questo campo è detto *Length* e contiene la lunghezza del frame e per gestire il passaggio a livello di rete è stato creato un livello intermedio chiamato LLC (*Logical Link Control*).

Data Contiene i dati che devono essere trasmessi.

Pad E' un campo di riempimento usato per garantire una dimensione minima di 64 byte al frame. In questo modo la trasmissione del frame richiede almeno 2 volte il tempo di roundtrip, garantendo che il mittente possa accorgersi di una eventuale collisione.

Checksum CRC a 32 bit per il controllo su un eventuale errore di trasmissione.

3.3.4 Storia e versioni di Ethernet

10 Base 5 È la prima versione, veniva usato un grosso cavo coassiale che permetteva il collegamento di un transceiver ogni 2,5 metri, quest'ultimo si occupa del carrier detection e del collision detection. Il numero 10 indica la velocità (10Mbps) e 5 la lunghezza massima dei segmenti (500 metri).

10 Base 2 Viene usato un tipo di cavo coassiale più sottile, le giunzioni vengono effettuate mediante connettori BNC e il transceiver viene spostato all'interno del pc. E' più economico e affidabile però la lunghezza massima di un segmento cala (200 metri) così come il numero di computer per segmento (30).

10 Base T Il cavo condiviso viene sostituito da hub collegati mediante cavi UTP dedicati. I costi diminuiscono, l'affidabilità aumenta così come il numero di computer per segmento (1024). Diminuisce però la lunghezza massima di un segmento (100 metri).

10 Base F Si basa sulla fibra ottica, permette segmenti fino a 2km.

Fast Ethernet (100 Base T4-TX-FX) Il clock viene aumentato di un fattore 10 però, dato che con Ethernet $Efficienza = \frac{1}{Banda \cdot Lunghezza}$, è stato necessario utilizzare degli switch al posto degli hub. La versione T4 usa 4 cavi UTP (100 metri), la TX usa un cavo UTP5 (sempre 100 metri) mentre FX usa la fibra ottica e arriva fino a 2km

Gigabit Ethernet Oltre ad incrementare la velocità, introduce anche una modalità point-to-point. Per ottenere questa velocità viene perso il carrier sense. I frame vengono trasmessi a blocchi (frame bursting), riuscendo a tenere una copertura di 200 metri. Per evitare problemi di sincronizzazione, i dati vengono codificati in modo da evitare sequenze di 0 o 1 troppo lunghe. Inoltre, viene usato l'encoding 8B/10B in modo che non ci siano mai 4 bit consecutivi identici in una word e che non ci siano mai più di sei 0 o sei 1 nella stessa word. Viene introdotto anche un frame speciale per mandare in pausa la trasmissione, dato che se uno switch si blocca anche per 1ms si possono accumulare un elevato numero di frame.