

LE RETI LOCALI A BUS	1
Allocazione statica del bus	2
I protocolli con rilevamento di collisione.	2
Protocollo ALOHA puro	3
Protocollo ALOHA a slot temporali	3
Protocolli con rilevamento di portante	4
CSMA1	4
CSMA non persistente	4
Protocolli senza collisione	5
Protocollo a mappa di bit	5
Protocolli a contesa limitata	6
Standard IEEE 802.3 (rete Ethernet)	8
Standard IEEE 802.4 token bus	9

Le reti locali a bus

In una rete locale a bus tutti i computer sono collegati da un unico canale di trasmissione condiviso. Una rete locale di questo tipo è detta di tipo broadcast in quanto è possibile che più stazioni di lavoro cerchino di accedere

contemporaneamente al bus creando delle collisioni cioè una situazione di contesa per il possesso del bus stesso e la sovrapposizione di informazioni. Occorre allora stabilire delle regole di accesso al mezzo che compongono il sottostrato di accesso al mezzo o Medium Access Control (MAC).

Allocazione statica del bus

Una prima soluzione per il problema della condivisione del bus è quello di ripartire il suo uso fra le varie stazioni in intervalli di tempo (tecniche di multiplexing) o suddividendo la banda di frequenza in vari canali. La distribuzione della risorsa bus avviene in maniera statica nel senso che viene effettuata a priori in base a calcoli preventivi e non viene modificata dal variare delle condizioni di lavoro delle varie stazioni collegate. Questi metodi trovano il loro limite nel caso in cui il numero di utenti non sia limitato o comunque stabile nel tempo. Se il numero di utenti aumenta, esauriti i canali a disposizione alcuni di essi dovranno attendere per accedere al bus mentre se il traffico diminuisce alcuni canali rimarranno inutilizzati con una riduzione dell'efficienza della rete. Nel caso del multiplexing temporale gli slot assegnati ad una stazione, nel caso che questa non stia operando rimarrebbero anch'essi inutilizzati.

I protocolli con rilevamento di collisione.

Il concetto fondamentale che sta alla base di questi tipi di protocolli è che le varie stazioni inviano i propri messaggi sul bus liberamente. Non è previsto, cioè, alcun controllo sull'accesso al mezzo. Se più stazioni tentano di utilizzare

contemporaneamente il bus si ha una collisione. La collisione comporta il fatto che i segnali elettrici lungo il bus vengono alterati e non sono più corrispondenti ai codici rappresentanti una stringa di bit. Le stazioni che hanno inviato i messaggi restano in ascolto dei segnali che attraversano il bus e possono così rendersi conto che si è avuto una collisione. A questo punto il loro comportamento si differenzia a seconda del protocollo.

Protocollo ALOHA puro

Nel protocollo ALOHA puro le stazioni che hanno dato luogo alla collisione restano in attesa per un tempo di durata casuale e poi ritentano di inviare il messaggio nuovamente. Il fatto che i tempi di attesa siano casuali fa in modo che sia poco probabile che esse ritentino di utilizzare il bus contemporaneamente. Uno dei difetti di questo protocollo è che la singola stazione si può rendere conto del fatto che ci sia stata una collisione ma non del punto della trasmissione in cui tale collisione è avvenuta per cui essa è costretta a ritrasmettere di nuovo tutto il messaggi, oppure una collisione può avvenire anche quando una trasmissione sia iniziata senza problemi.

Protocollo ALOHA a slot temporali

Un miglioramento è costituito dal protocollo ALOHA a slot temporali. In sostanza con questo protocollo, il tempo è suddiviso in intervalli regolari o slot. Una stazione non può iniziare una trasmissione in un istante qualsiasi ma soltanto all'inizio di uno slot. Il vantaggio consiste nel fatto che se la trasmissione comincia senza collisioni,

esse non potranno avvenire per tutta la durata dello slot temporale, poiché un'altra stazione potrà iniziare a sua volta una trasmissione soltanto all'inizio del successivo slot temporale.

Protocolli con rilevamento di portante

In tali protocolli le stazioni collegate al bus sono dotate di dispositivi di hardware in grado di rilevare se il bus è libero o occupato. Ciò non basta a garantire che non avverranno collisioni. Infatti occorre ricordare che un segnale viaggia lungo un canale trasmissivo in un tempo non nullo per cui, a causa dei ritardi di propagazione, pur essendo iniziata una trasmissione da parte di un'altra stazione, gli altri utenti possono credere che il bus sia libero e provare a iniziare una trasmissione essi stessi.

CSMA1

Un esempio di questi tipi di protocolli è il CSMA1 (carrier sense multiple access 1). Il problema fondamentale è che, poiché le varie stazioni rilevano lo stato del bus contemporaneamente, esse contemporaneamente possono credere erroneamente che il bus sia libero e contemporaneamente possono tentare di iniziare una trasmissione.

CSMA non persistente

Nel CSMA non persistente ogni stazione fa passare un intervallo di tempo di durata casuale fra due istanti in cui controlla lo stato del bus. Si riduce così la probabilità che due stazioni contemporaneamente possano leggere lo stato del bus e provare a trasmettere il proprio messaggio.

Protocolli senza collisione

Questi protocolli fanno in modo che una collisione non possa mai verificarsi. Ciò avviene però a discapito della velocità della rete.

Protocollo a mappa di bit

Un esempio è il protocollo a mappa di bit. Esso prevede ciclicamente una fase di contesa seguito da una fase di trasmissione. La fase di contesa è suddivisa in tanti slot temporali quante sono le stazioni collegate al bus. Le stazioni sono numerate da 1 a N. Nel primo slot di tempo della fase di contesa, la stazione numero 1 invia sul bus un bit pari a 0 se non vuole trasmettere, un bit pari ad 1 nel caso voglia inviare un messaggio. La stessa cosa farà la stazione numero 2 nello slot 2 e così via.

Numero di Stazioni	10
--------------------	----

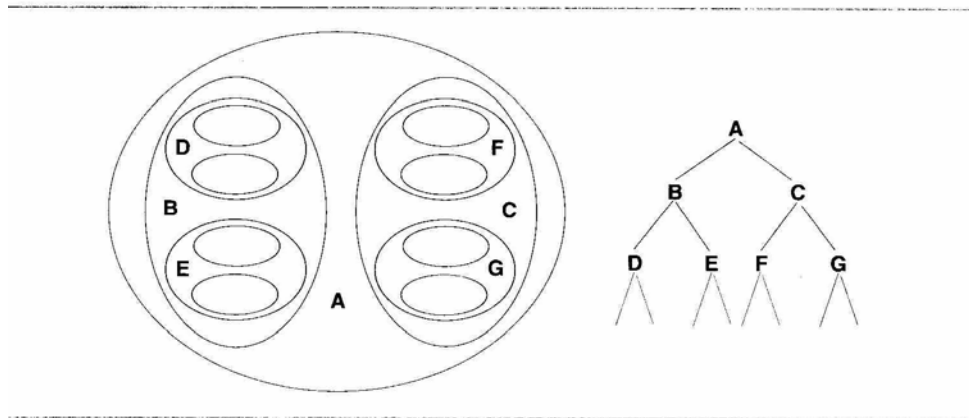
Periodo di Contesa					
Slot 1	1	LA STAZIONE VUOLE TRASMETTERE			
Slot 2	0				
Slot 3	1	LA STAZIONE VUOLE TRASMETTERE			
Slot 4	0				
Slot 5	0				
Slot 6	0				
Slot 7	0				
Slot 8	1	LA STAZIONE VUOLE TRASMETTERE			
Slot 9	0				
Slot 10	0				

Al termine della fase di contesa tutte le stazioni sanno quali sono quelle che vogliono trasmettere. Nell'esempio di figura vogliono trasmettere le stazioni 1, 3 e 8. Nella fase di trasmissione le stazioni che si sono prenotate invieranno i propri frame secondo l'ordine di prenotazione. Con questo metodo le collisioni sono impossibili. Chiaramente gli svantaggi sono relativi ai ritardi nella trasmissione. Infatti, se nella fase di trasmissione una sola stazione richiede di trasmettere, il suo frame non potrà essere inviato sul bus finché non sarà terminata la fase di contesa.

Protocolli a contesa limitata

I protocolli a contesa sono preferibili nel caso che il traffico sia ridotto. Infatti in tal caso le possibilità di collisione sono poche. Nel caso, invece, di traffico elevato sono preferibili i protocolli senza collisione. La scelta fra i vari tipi di protocolli è dunque delicata soprattutto per reti in cui le condizioni di funzionamento sono variabili. Per ovviare a queste difficoltà sono stati sviluppati i protocolli a contesa limitata che hanno un comportamento e prestazioni che si adattano alle condizioni di carico della rete.

In tali protocolli l'insieme A di tutte le stazioni collegate al bus può essere suddiviso in due sottoinsiemi B e C . Ognuno di questi sottoinsiemi viene suddiviso ulteriormente in altri sottoinsiemi e così via.



La fase di contesa avviene nel modo seguente. Durante il primo slot temporale tutte le stazioni possono richiedere l'uso del bus. Se una sola stazione richiede il bus non vi sarà collisione: la fase di contesa termina e si passa alla trasmissione. Se più stazioni richiedono il bus vi è collisione. Si passa allora ad un secondo slot della fase di contesa. In questo caso, però, viene consentito soltanto alle stazioni che ricadono nel sottogruppo B di richiedere l'uso del bus. Durante il secondo slot si possono verificare, allora, tre casi:

1. nessuna stazione del sottogruppo B richiede l'uso del bus: in tal caso nello slot successivo si solo i computer appartenenti al sottogruppo C possono tentare di comunicare
2. una sola stazione del gruppo B richiede di trasmettere: la fase di contesa termina e nella fase successiva avverrà la trasmissione
3. più stazioni del sottogruppo B cercano di ottenere il bus: in questo caso si ha collisione per cui, in uno slot successivo, si avrà una nuova contesa che verrà limitata al sottogruppo D e così via.

In ogni caso, alla fine della fase di contesa, tutte le stazioni che avranno tentato di ottenere il bus durante il primo slot temporale riusciranno a trasmettere.

Standard IEEE 802.3 (rete Ethernet)

Questo standard prevede l'uso di protocollo Carrier Sense Multiple Access with Collision detection CSMA/CD. Dispositivi detti transceiver che collegano le stazioni al bus, rilevano se la linea è libera o meno. Se, a causa di ritardi di propagazione del segnale lungo il bus, più stazioni credono che il bus sia libero e tentano di connettersi, si verifica comunque una collisione. La stazione che rileva la collisione, interrompe la trasmissione del proprio frame ed invia in linea un segnale di interferenza per avvisare le altre stazioni della presenza di un'anomalia. Essa tenterà successivamente di ricollegarsi al bus, aspettando per un tempo casuale secondo un algoritmo detto di backoff binario esponenziale. Tale algoritmo funziona nel modo seguente:

1. dopo la prima collisione, quindi al secondo tentativo, la stazione può scegliere di attendere un tempo zero per ritentare di connettersi al bus o un tempo pari ad uno slot di attesa che lo standard definisce di durata pari a 51,2 microsecondi.
2. se si ha di nuovo collisione, cioè al terzo tentativo, la stazione può scegliere di effettuare un nuovo tentativo dopo 0, 1, 2 o 3 slot temporali.
3. se c'è di nuovo collisione la stazione può scegliere fra 8 possibilità diverse e così via

4. in generale, al N-esimo tentativo, la stazione può scegliere fra 2^{N-1} possibilità diverse per scegliere il numero di slot temporali da attendere.

In questo protocollo le stazioni sono tutte della stessa priorità. Non vi è possibilità, quindi, di distinguere fra una trasmissione più importante ed una meno importante. Inoltre l'algoritmo prima descritto non è adatto nel caso che le collisioni siano molto probabili: una stazione potrebbe teoricamente attendere un tempo notevolissimo prima di poter accedere al bus per cui tale standard non è utilizzabile in applicazioni industriali

Standard IEEE 802.4 token bus

In questo standard ogni stazione ha una sua priorità logica determinata da un codice. Ad ogni istante non è detto che tutte le stazioni siano collegate alla rete. Si può avere ad esempio, la seguente situazione

Stazione con priorità 7	Non collegata
Stazione con priorità 6	Collegata
Stazione con priorità 5	Non collegata
Stazione con priorità 4	Collegata
Stazione con priorità 3	Collegata
Stazione con priorità 2	Collegata
Stazione con priorità 1	Collegata

La stazione con priorità maggiore al momento collegata alla rete è la numero 6 e può trasmettere i suoi dati sul bus. Quando essa avrà terminato invierà un particolare frame detto gettone o token alla stazione collegata alla rete che ha priorità immediatamente successiva. Poiché la numero 5 non è collegata, il token andrà alla numero 4. Per fare ciò la stazione numero 6 possiede una lista aggiornata dello stato della rete. Il token è un'autorizzazione alla stazione che lo riceve di trasmettere i suoi dati sul bus. Soltanto la stazione che in un dato momento possiede il token può utilizzare il bus. In tal modo non sono possibili collisioni. Se la stazione numero 5 non ha dati da trasmettere invia il token alla stazione che la segue immediatamente nella lista di priorità e che risulta collegata alla rete.

La lista posseduta dalla stazione che possiede il token viene aggiornata nella seguente maniera. Ad intervalli regolari la stazione invia sul bus un frame detto solicit successor che contiene il proprio indirizzo e quello del suo successore. Se una stazione che ha una priorità compresa fra quella della stazione che possiede il token e il suo successore intende essere inserita in lista, essa invia un messaggio alla stazione che possiede il token. Nell'esempio riportato nella tabella precedente la stazione 6 che possedeva il token inviava sul bus un solicit successor contenente il proprio indirizzo e quello della stazione 4. se la stazione 5 voleva collegarsi avrebbe inviato alla numero 6 un messaggio per farsi inserire in lista.

Può accadere che fra la stazione che possiede il token e la successiva presente in lista esistano più stazioni che potrebbero voler inserirsi e che queste, in seguito al solicit successor, possano richiedere contemporaneamente l'immissione in lista. L'invio

contemporaneo sul bus dei rispettivi frame di richiesta provocherebbe una collisione. In tal caso la stazione che possiede il token, rilevata la collisione, invia sul bus un frame di controllo denominato resolve contention. Le stazioni che hanno causato la collisione tenteranno di nuovo di inviare il messaggio attendendo un tempo di durata casuale (possono scegliere casualmente un tempo di durata pari a 0, 1, 2 o 3 slot temporali).

Nella fase iniziale di attività della rete naturalmente nessuna stazione possiede il token. In questo caso la prima stazione che vuole connettersi invia sul bus un frame di controllo che si chiama claim token. Poiché non ci sono altri nodi attivi tale stazione creerà una lista composta soltanto da se stessa. Tale lista si allungherà via via che la stazione che possiede il token emetterà frame solicit successor e vi saranno risposte.

Una stazione, per uscire dall'elenco dei nodi attivi in rete, quando le arriva il token, comunica alla stazione precedente chi è il suo successore e passa ad esso il suo token.

Consideriamo, ad esempio, la seguente situazione

Stazione con priorità 7	Non collegata
Stazione con priorità 6	Collegata
Stazione con priorità 5	Non collegata
Stazione con priorità 4	Collegata
Stazione con priorità 3	Non Collegata
Stazione con priorità 2	Collegata

Stazione con priorità 1	Collegata
-------------------------	-----------

La stazione 4 possiede il token e lo passa alla 2 che è la stazione presente nella sua lista perché la 3 non è attiva. La stazione 2 vuole uscire dalla lista dei nodi attivi per cui comunica alla 4 che il suo nuovo successore è la stazione 1 ed invia il token a quest'ultima.

Le liste sono naturalmente circolari per cui quando il token arriva alla stazione di priorità più bassa, da questa verrà poi inviato di nuovo alla stazione di priorità più elevata presente in lista. Nell'esempio della tabella precedente, visto che la stazione di priorità massima, la numero 7, è non attiva, il successore della numero 1 è la stazione numero 6.

Lo standard 802.4 prevede anche particolari modalità di comportamento per gestire casi di comportamento anomalo della rete. Ogni stazione, ad esempio, dopo aver passato il token al suo successore, resta in ascolto sulla linea per vedere se esso comincia a trasmettere frame di dati o il token alla stazione successiva. Se questo non avviene la stazione invia di nuovo il token al suo successore. Se ancora una volta il successore non dà segno di risposta la stazione di partenza invia in linea un frame detto *who follows* (chi segue?) che contiene l'indirizzo del suo successore in apparente avaria. La stazione collegata che segue immediatamente il successore in avaria risponde autonominandosi successore con un frame detto *set successor*. Se anche il frame *who follows* non ha risposta la stazione di partenza emette un frame *solicit successor 2* che ha lo scopo di richiedere se vi sono stazioni attive. In caso

negativo la stazione di partenza rigenera la sua lista facendo in modo che contenga soltanto il proprio nome.

Una possibile anomalia è ancora la perdita del token per disturbi sulla linea. La perdita del token porterebbe, naturalmente, al blocco della rete. E' per questo che le stazioni hanno un timer interno che riparte ogni volta che circola un frame sul bus (indice che tutto procede regolarmente e c'è una stazione che detiene il token). Se una stazione rileva che il token è andato perduto, perché il suo timer si è esaurito senza che alcuna stazione iniziata a trasmettere, invia sul bus un frame di claim token e reinizializza la rete.

B

backoff binario esponenziale; 8

C

claim token; 11; 13

CSMA non persistente; 4

R

resolve contention; 11

S

set successor; 12

solicit successor; 10; 11; 12

solicit successor 2; 12

standard 802.4; 12

T

token; 9; 10; 11; 12; 13

W

who follows; 12