**Data Link Layer LAN**

**Interferenze e Collisioni**

Connessione a un canale broadcast condiviso. Centinaia o anche migliaia di nodi possono comunicare direttamente su un canale broadcast:

– Si genera una *collisione* quando i nodi ricevono due o più frame contemporaneamente.

**Protocolli ad accesso multiplo**

Protocolli che fissano le modalità con cui i nodi regolano le loro trasmissioni sul canale condiviso.

La comunicazione relativa al canale condiviso deve utilizzare lo stesso canale!

1. **Protocolli a suddivisione del canale**

TDMA: accesso multiplo a divisione di tempo.

* Suddivide il canale condiviso in *intervalli di tempo.*
* Gli slot non usati rimangono inattivi

FDMA: accesso multiplo a divisione di frequenza.

* Suddivide il canale in bande di frequenza.
* A ciascuna stazione è assegnata una banda di frequenza prefissata.

1. **Protocolli ad accesso casuale**

Definisce:

* 1. Come rilevare un’eventuale collisione.
  2. Come ritrasmettere se si è verificata una collisione.

**Slotted Aloha**

Il tempo è suddiviso in slot. Quando un nodo vuole trasmettere attende l’inizio di uno slot. Se non ci sono collisioni allora trasmette, altrimenti ritrasmetterà con probabilità p in uno slot successivo.

**Pro**

• Consente a un singolo nodo di trasmettere continuamente pacchetti alla massima velocità del canale.

• È fortemente decentralizzato, ciascun nodo rileva le collisioni e decide indipendentemente quando ritrasmettere.

• È estremamente semplice.

**Contro**

• Una certa frazione degli slot presenterà collisioni e di conseguenza andrà “sprecata”.

• Un’alta frazione degli slot rimane vuota, quindi inattiva.

**Aloha puro**

Non sincronizzato.

Quando arriva il primo pacchetto: lo trasmette immediatamente e integralmente nel canale broadcast.

Elevate probabilità di collisione.

Maggiore è il traffico e peggiore è il protocollo in termini di throughput.

**CSMA**

Si pone in ascolto prima di trasmettere.

Se rileva che il canale è libero, trasmette l‘intero pacchetto.

Se il canale sta già trasmettendo, il nodo aspetta un altro intervallo di tempo.

Essendo a rilevazione della portante utilizza una codifica Manchester.

Per via dei ritardi di propagazione, possono esserci ancora collisioni; quindi quando un nodo rileva una collisone cessa subito la trasmissione.

**CSMA 1-persistente**

Quando un calcolatore ha dati da trasmettere, ascolta il segnale presente sul mezzo trasmissivo.

Se trova il canale libero, trasmette il frame altrimenti continua ad ascoltare fino a che il canale non si libera, e poi trasmette il frame.

In caso di collisione, la stazione aspetta un tempo casuale e ripete l’algoritmo.

Il protocollo si chiama 1-persistente perché quando trova il canale occupato, resta in ascolto continuamente, ed appena il canale si libera trasmette con probabilità 1 (sempre).

**CSMA p-persistente**

Il tempo è suddiviso in slot temporali.

Chi vuole trasmettere ascolta continuamente il canale e quando lo trova libero trasmette con probabilità p, altrimenti attende uno slot successivo con probabilità 1-p.

In caso di collisione, la stazione attende un tempo casuale e ripete l’algoritmo.

**CSMA non persistente**

Quando una stazione vuole trasmettere e trova il canale occupato, non attende in ascolto, ma riprova dopo un tempo casuale.

**CSMA/CD**

Lavora in tre fasi:

* **Carrier sense**: ogni stazione che deve trasmettere ascolta il bus e decide di trasmettere solo se questo è libero.
* **Multiple access**: per via dei ritardi di propagazione, due o più stazioni potrebbero trasmettere contemporaneamente.
* **Collision detection**: per rilevare una collisione, ogni stazione, mentre trasmette un pacchetto, ascolta i segnali sul mezzo trasmissivo, confrontandoli con quelli da lei generati.

Alla fine di una trasmissione si può avere un periodo di contesa se 2 o più stazioni iniziano a trasmettere.

In caso di avvenuta collisione, la stazione trasmittente sospende la trasmissione e trasmette una sequenza di **jamming** (interferenza trasmissiva) per comunicare a tutte le stazioni di rilevare l'avvenuta collisione.

Le stazioni in ascolto riconoscono il jamming e scartano i bit ricevuti. La stazione trasmittente ripete il tentativo di trasmissione dopo un tempo pseudo-casuale per un numero di volte non superiore a 16.

1. **Protocolli a rotazione**

**Protocollo a prenotazione**:

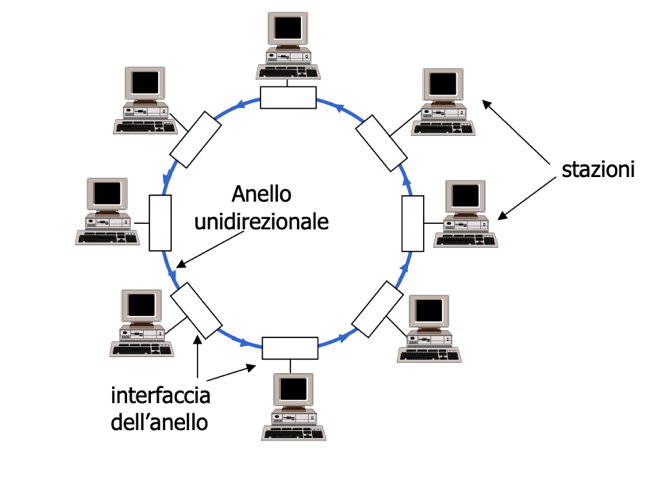
Protocollo a mappa di bit elementare:

* sulla rete ci sono N stazioni, numerate da 0 a N-1
* alla fine della trasmissione di un frame inizia un periodo di contesa, in cui ogni stazione, andando per ordine di indirizzo, trasmette un bit che vale 1 se la stazione deve trasmettere, 0 altrimenti.
* al termine del periodo di contesa (privo di collisioni in quanto ogni stazione aspetta il suo turno) tutti hanno appreso quali stazioni devono trasmettere, e le trasmissioni procedono un frame alla volta sempre andando per ordine
* se una stazione riceve dati da trasmettere quando la fase di prenotazione è terminata, deve attendere il successivo periodo di contesa per prenotare la propria trasmissione

L’efficienza di questo protocollo è bassa per grandi valori di N e basso carico trasmissivo;

**Token ring**

Non utilizza un mezzo broadcast ma un insieme di collegamenti punto-punto associati in successione per realizzare una topologia ad anello

****

* Questo protocollo prevede l’utilizzo della tipologia ad anello.
* Sull’anello circola un piccolo frame, detto token (gettone) che le stazioni ricevono da una parte e ritrasmettono dall’altra in continuazione.
* Una stazione è autorizzata a trasmettere dati solo quando è in possesso del token:
  + la stazione riceve il token, lo trattiene ed inizia a trasmettere dati;
  + terminata la trasmissione, ritrasmette il token in coda ai frame di dati.

**Efficienza**

Il protocollo token ring è poco efficiente in condizioni di basso carico.

La stazione che deve trasmettere deve attendere di ricevere il token (o in generale deve attendere il suo turno) prima di poterlo fare, anche se il canale non e’ occupato

* In condizioni di carico elevato, quando tutti vogliono trasmettere, l’efficienza del protocollo sfiora l’unità
  + il solo overhead è dovuto alla necessità che ha una stazione di identificare il token prima di poter trasmettere
  + in questi protocolli il token è scelto in modo opportuno per minimizzare l’overhead
* Una importante caratteristica di questo genere di protocolli è la possibilità di valutare un tempo massimo di ritardo per le trasmissioni
  + una stazione che desidera trasmettere dovrà attendere al più N tempi di trasmissione (uno per stazione, nel caso tutti debbano trasmettere) prima che tocchi nuovamente ad essa
  + questo permette l’utilizzo del protocollo in situazioni in cui i tempi di risposta possono essere determinanti (ad esempio una catena di montaggio)