

Link di accesso multiplo

Esistono due tipi principali di collegamenti: **Punto-a-Punto (P2P)** e **Broadcast**. Nel primo caso, la comunicazione avviene tra due soli nodi, mentre nel secondo il mezzo è condiviso tra più dispositivi. Ci concentreremo sui link **broadcast**, che richiedono protocolli per gestire il traffico in modo efficiente.

Broadcast (Shared Wire or Medium)

Un mezzo di trasmissione condiviso da più dispositivi, dove i dati trasmessi sono visibili a tutti i dispositivi collegati. Richiede protocolli specifici per gestire la contesa (**collisioni**) e garantire la corretta trasmissione, ad esempio:

- **Ethernet condiviso** (cablaggio tradizionale).
- **HFC (Hybrid Fiber-Coaxial)**, utilizzato nelle reti via cavo.
- **802.11 LAN wireless** (Wi-Fi), reti mobili (4G/5G), comunicazioni satellitari.

Ogni dispositivo ascolta il mezzo e può decidere se i dati ricevuti sono destinati a lui. Può esserci **contesa** per l'accesso al mezzo, gestita da protocolli come CSMA/CD (Carrier Sense Multiple Access with Collision Detection) in Ethernet.

Caratteristiche ideali di protocolli ad accesso multiplo

Dato un **MAC** (Multiple Access Channel) di rate R (bit al secondo) vorremmo:

1. Quando **un nodo** trasmette deve poterlo fare a un rate R .
2. Quando **M nodi** trasmettono devono poterlo fare a un rate $\frac{R}{M}$.
3. Completamente **decentralizzato**, senza esterni che coordinano o clock di sincronizzazione.
4. **Semplice**, non deve avere una gestione complicata.

Non potremo averli tutti assieme, immaginiamo una coperta che tirandola verso il viso scopre i piedi e viceversa.

Tassonomia MAC (Multiple Access Channel)

Una rappresentazione delle tipologie dei MAC sono i seguenti 3 macrogruppi.

Categoria	Pro	Contro
Partizionamento di canale	No collisioni, prevedibilità	Utilizzo inefficiente con traffico variabile
Accesso casuale	Semplice, adatto a traffico variabile	Collisioni, meno efficiente con molti nodi
Taking Turns	Controllo efficiente	Maggiore latenza con pochi dati

Partizionamento di canale

Il canale viene suddiviso in **porzioni più piccole**, assegnate a ciascun nodo per un uso esclusivo. Ad esempio **TDMA, FDMA**. È più adatto per scenari con traffico prevedibile, il punto 3 dei desiderati non è già più verificato perchè **avremo una figura che coordina l'invio**, e il punto 1 dato che **non è mai possibile inviare a rate R** .

Accesso casuale

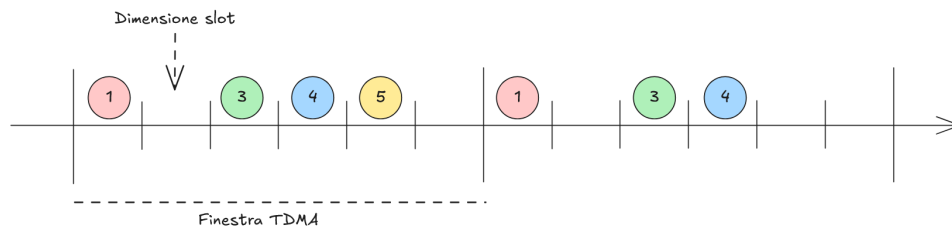
Il canale **non è suddiviso**, ed è condiviso tra tutti i nodi. Le collisioni sono permesse, ma i protocolli forniscono meccanismi per recuperare da esse ad esempio **ALOHA, CSMA** o **CSMA/CD**. Maggiore flessibilità, ma meno efficiente in caso di traffico elevato.

Taking Turns

I nodi trasmettono a turno. Quelli con più dati da inviare possono occupare il canale più a lungo ad esempio **Token Passing**. Riduce le collisioni. Anche in questo caso avremo un controllore e quindi il terzo punto non è più verificato.

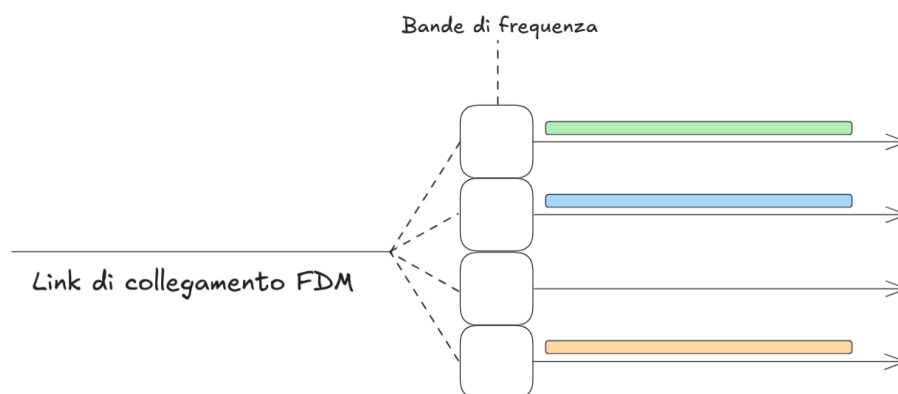
Partizionamento di canale, TDMA

L'accesso al canale è fatto in round ovvero attraverso slot temporali, ogni slot ha una dimensione fissa e quelli non usati sono detti **slot idle**. C'è un **clock** che gestisce l'**invio dei frame negli slot**, ogni host ha un codice e in base al tic di clock l'host attuale può **invadere in maniera esclusiva** il canale inviano il frame. Abbiamo uno spreco essendo che alcuni slot saranno in idle.



Partizionamento di canale, FDMA

Lo spettro del canale è diviso in frequenze di banda, ovvero ogni macchina ha una sua banda di frequenza assegnata con cui può trasmettere, a destinazione verranno divisi e filtrati secondo la frequenza. Avremo uno spreco essendo che alcune frequenze non saranno utilizzate.



Accesso casuale

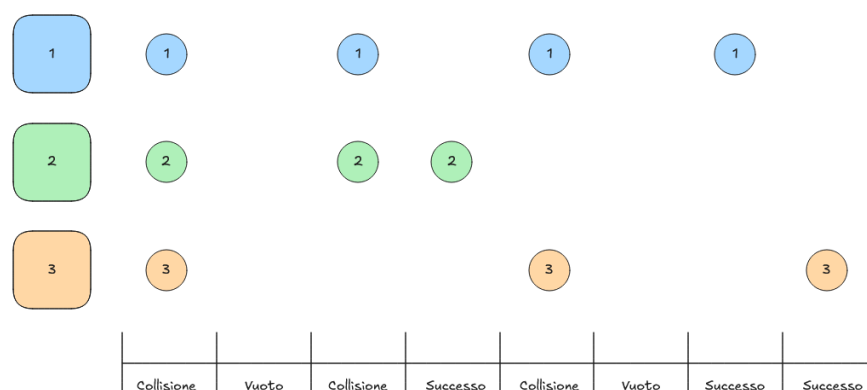
Per ogni host, quando deve inviare un frame invade il canale e trasmette a velocità R , se un altro host invade il canale avremo una collisione, i protocolli ad accesso casuale indicano come accorgersi della collisione e ripristinare l'invio.

Slotted ALOHA

Bisogna fare dei presupposti, ogni frame ha la **stessa dimensione**, il **tempo è diviso** in porzioni di **dimensioni fisse** e i **nodì sono sincronizzati** e iniziano a trasmettere appena il clock parte così da non perdere tempo, ad esempio se ho 3 frame da trasmettere occupo 3 slot temporali. Il problema è quando ci sono più host che devono trasmettere, entrambi trasmettono e si genera una collisione.

Quando un nodo trasmette un frame:

- Se **non c'è stata collisione**: il nodo può trasmettere un frame allo slot successivo
- Se **c'è stata una collisione**: il nodo ritrasmette il frame in uno slot successivo con probabilità p di trasmettere. Viene calcolata la probabilità di invio per non rendere deterministica la collisione.



- **Pro:** se c'è solo un nodo che trasmette lo può fare a un rate R , decentralizzato e semplice.
- **Contro:** se c'è collisione abbiamo uno spreco dello slot, slot inutilizzati e serve una sincronizzazione dei clock per l'invio.

Efficienza di Slotted ALOHA: supponiamo di avere N nodi che vogliono trasmettere, ognuno con molti frame e con una probabilità calcolata p di invio.

- Probabilità che un host trasmetta da solo in uno slot è $p(1 - p)^{N-1}$
- Probabilità che almeno un host trasmetta con successo in uno slot è $Np(1 - p)^{N-1}$
- Per la massima efficienza dobbiamo trovare la probabilità migliore ovvero $Np(1 - p)^{N-1}$ massimizzato come valore da cui otteniamo p^* .
- Lo utilizziamo nel calcolo $Np^*(1 - p^*)^{N-1}$

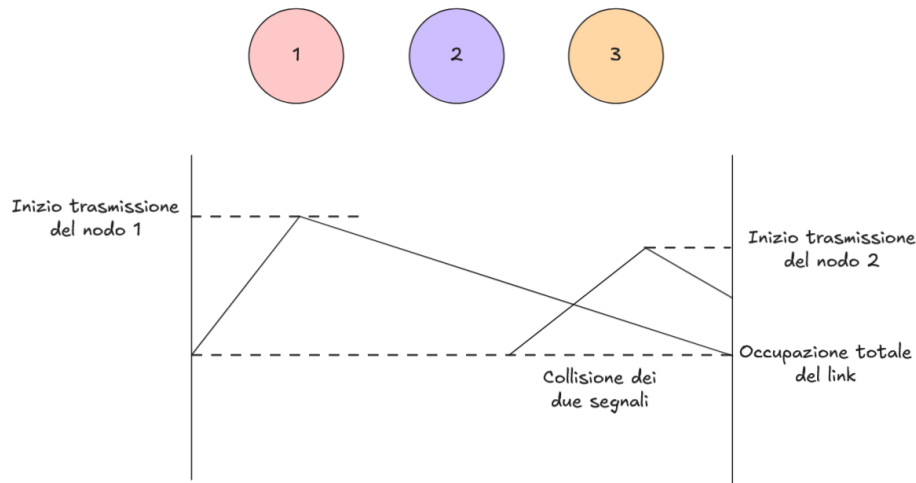
Con il numero di host N che tende a infinito troviamo che il canale nel migliore dei casi è usato al 37% del tempo perchè la funzione tende a $\frac{1}{e}$ ovvero 0,37.

Pure ALOHA

Il protocollo Pure ALOHA è **senza slot temporali**, molto più **semplice** e **non richiede sincronizzazione**. Il frame che arriva per primo **viene inviato immediatamente**, se invece troviamo **collisione** aspettiamo la fine del frame attuale nel collegamento e ritrasmettiamo alla fine di esso con probabilità p . Questo metodo ha una efficienza del minore ovvero che il canale viene occupato per il 18% del tempo perchè la collisione aumenta senza sincronizzazione.

CSMA

Il protocollo **CSMA** (Carrier Sense Multiple Access) in versione semplice prevede un controllo singolo: **se il canale è libero invio il frame, altrimenti aggiunto un delay alla trasmissione**. Abbiamo un problema di collisioni dovuto al ritardo di propagazione, quindi un host può vedere il canale libero non sapendo che l'altro ha già iniziato a trasmettere. **Avendo una collisione si perde l'intero tempo di trasmissione** essendo che l'intero frame andrà ritrasmesso.



CSMA/CD

L'estensione **CD** indica il **Collision Detection** integra un meccanismo, ovvero che ogni dispositivo **verifica** se il **segnale presente sul canale** è uguale a quello che deve trasmettere, se diverso abbiamo una interferenza e quindi una collisione, le **collisioni vengono individuate in breve** e i dispositivi coinvolti in essa cessano di trasmettere, dopo la interruzione **ritrasmettono** sulla base di un **delay casuale di tempo**. La collisione è semplice nelle reti cablate, in quelle wireless no perché i dispositivi non possono ascoltare il canale e trasmettere contemporaneamente per come sono implementate.

Algoritmo di funzionamento di CSMA/CD

1. La **NIC (Network Interface Card)** riceve un datagramma dal livello di rete e lo incapsula in un frame Ethernet.
2. La NIC verifica se il canale è libero o occupato:
 - **Se il canale è libero:** inizia immediatamente la trasmissione.
 - **Se il canale è occupato:** la NIC attende finché il canale non diventa libero.
3. Se la NIC trasmette l'intero frame senza rilevare collisioni, la trasmissione è completata con successo.
4. Durante la trasmissione, la NIC monitora il canale per verificare la presenza di segnali sovrapposti (indicativi di una collisione) e procede a:
 - Interrompere immediatamente la trasmissione con segnale **abort**.
 - Invia un segnale di **jam** per notificare agli altri dispositivi che si è verificata una collisione.
5. Dopo aver rilevato una collisione, la NIC attende un intervallo casuale di tempo prima di ritentare la trasmissione. Questo intervallo aumenta esponenzialmente dopo ogni collisione:
 - Dopo la **m-esima collisione**, la NIC sceglie un valore casuale K da $\{0, 1, \dots, 2^m - 1\}$.
 - L'attesa è calcolata come $K \cdot 512$ bit-time (bit-time è un'unità di tempo corrispondente al tempo necessario per trasmettere un bit sul canale).
 - Dopo l'attesa del backoff, la NIC torna al passo 2 (controllo del canale) e riprova a trasmettere.

Efficienza di CSMA/CD: Per trovarla dobbiamo calcolare il suo valore sulla base del ritardo di trasmissione e propagazione.

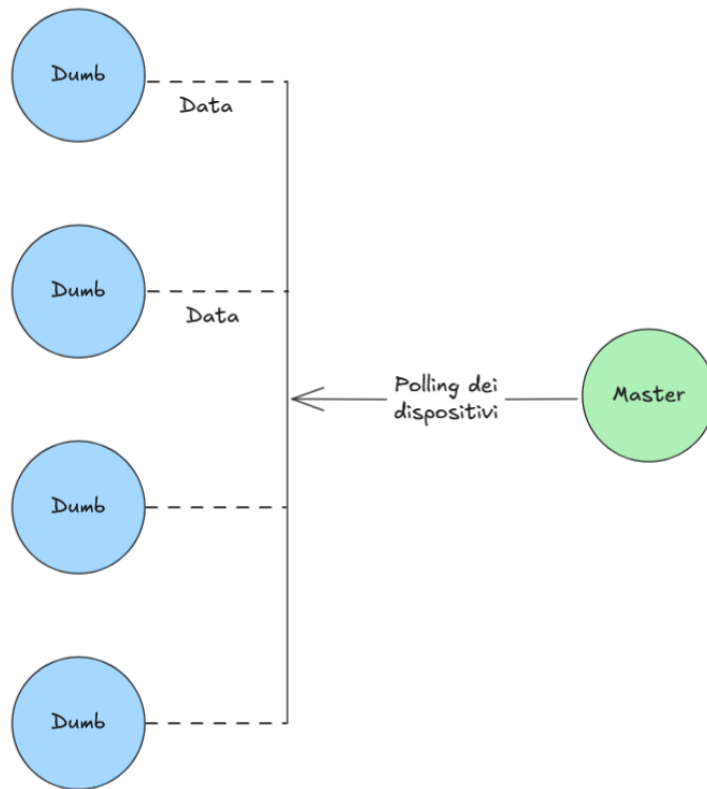
$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

L'efficienza tende a 1 quando t_{prop} tende a 0 e t_{trans} tende a infinito. Ha prestazioni migliori rispetto ad ALOHA ed è decentralizzato, meno costoso e semplice.

Taking Turns

Polling

Questo metodo prevede un **master** che interroga (polling) dei dispositivi dumb che hanno dei dati da fornire, è uno SPOF e causa overhead dato che il master dovrà interrogare tutti i dispositivi anche quelli senza dati da fornire.



Token Passing

Il metodo Token Passing prevede l'utilizzo di un token che permette l'invio di frame con mutua esclusione sul canale di comunicazione, viene passato sequenzialmente al nodo successivo, i problemi sono che abbiamo overhead per il token che finisce ad host che non hanno niente da trasmettere e SPOF dato che se viene corrotto nessuno può più inviare.

