

1 Il livello Data Link e le LAN

1.1 Il data link layer

I dispositivi (host, router, server ecc.) che supportano il data link layer sono detti **nodi**. La comunicazione avviene tramite **canali** che connettono nodi adiacenti, questi canali vengono chiamati **link** (via cavo, wireless), un'intera rete LAN viene vista come un unico link. I pacchetti che vengono trasmessi a livello data link si chiamano **frame**.

Richiami esame di reti

A livello IP il pacchetto trasmesso è chiamato **datagramma**, a livello trasporto TCP il pacchetto è chiamato **segmento**.

1.1.1 I servizi forniti

Incapsulamento (framing) Nella maggior parte delle volte la parte contenente i dati del frame contiene il datagramma IP. Il livello data link fornisce come servizio quello di trasferire un datagramma da un nodo ad un altro attraverso i link.

Accesso al link Un protocollo di medium access control (MAC) specifica come il frame deve essere trasmesso sul link.

Trasporto affidabile Un protocollo di trasferimento affidabile garantisce che ogni frame raggiunga la sua destinazione senza errori. Assicura che il trasferimento sia affidabile tra ogni nodo in modo da non dover rimandare tutto il pacchetto TCP ma solo i frame (datagrammi) danneggiati.

Individuazione e correzione degli errori Il nodo mittente fornisce un meccanismo per individuare gli errori, che verranno poi corretti dal destinatario.

Controllo di flusso La velocità con cui il nodo mittente trasmette e il nodo ricevente riesce a ricevere.

Half-duplex e full-duplex Con half-duplex in una comunicazione tra due nodi solo uno alla volta può trasmettere mentre l'altro resta in attesa e viceversa. Con full-duplex entrambi i nodi possono trasmettere e ricevere contemporaneamente.

1.1.2 Implementazioni

Il data link layer deve essere implementato in ogni nodo connesso alla rete. Le funzionalità Ethernet sono integrate nella scheda madre o in un chip Ethernet. Il livello collegamento è implementato su un chip detto network adapter o NIC.

1.2 Individuazione e correzione degli errori

L'individuazione degli errori prevede aggiungere ai dati trasmessi dei bit/byte aggiuntivi che servono per permettere di rilevare o correggere l'errore. Potrebbero presentarsi degli errori sia sulla parte dei dati che sui bit aggiuntivi perché viaggiano su un canale non affidabile.

1.2.1 Controllo di parità

Ad una certa sequenza di bit aggiunto un singolo bit, il bit di parità, che viene aggiunto dal mittente seguendo certe regole che dovranno essere condivise con il ricevente. Se il numero di bit corrotti è pari il test viene passato con successo anche se è presente un errore.

Esempio

Aggiungo un 1 se il numero di 1 è dispari 0 se il numero di 1 è pari. Attraverso le matrici di parità calcolo i bit di parità sia per le righe che per le colonne e confronto se il numero di bit coincide con il bit di parità. Posso correggere l'errore solo se è presente **1 errore** ma segnalare l'errore su **2 bit**.



1.2.2 CRC (Cyclic Redundancy Check)

È uno dei più potenti rilevatori di errori ma non può correggerli.

I codici CRC sono anche detti codici polinomiali in quanto 'è possibile considerare la stringa da inviare come un polinomio i quali coefficienti sono i valori 0 e 1 della stringa di bit. I CRC possono individuare *resto ≤ bit errati consecutivi*.

Svolgimento (1)

D = numero di bit per i dati;

G = sequenza di bit (generatore) che deve soddisfare determinate caratteristiche. In questo caso $r + 1$ bit.

Aggiungo in coda una sequenza di CRC lunga r bit (un bit in meno rispetto al generatore).

$$\langle D, R \rangle = D \cdot 2^r \text{ XOR } R$$

$D \cdot 2^r$ shifto il dato di r bit

$\text{XOR } R$ metto come ultimi bit significativi quelli di R

Il mittente prende gli R bit dei CRC e prende $\langle D, R \rangle$ e lo divide per G , fa il modulo 2, **se il resto della divisione è 0 il test è passato** altrimenti segnalo che è presente un errore.

Posso rilevare un numero di bit errati $< G$. Inoltre posso rilevare un numero di errori dispari.

Svolgimento (2) Come calcolare R

R deve essere tale per cui:

$$D \cdot 2^r \text{ XOR } R = nG$$

AB	A XOR B
00	0
01	1
10	1
11	0

$$R = \text{resto}\left[\frac{D \cdot 2^r}{G}\right]$$

Esempio

$D=101110$ $G=1001$ $r=3$

```
10111000 :1001
1001 XOR  101011
  101
 000 XOR
 1010
1001 XOR
  110
 000 XOR
 1100
1001 XOR
  1010
 1001 XOR
   011 = R
```

Alla fine il **mittente** trasmetterà come sequenza 101110**011** con gli ultimi 3 bit uguali a quelli calcolati.

Il **ricevente** prende la sequenza e la divide per G , se il resto è 0 il test è passato altrimenti segnalo la presenza di un errore.

Generatori Esistono dei generatori G standard da 8, 12, 16 o 32 bit prestabiliti.

1.3 Protocolli ad accesso multiplo

Esistono due tipologie di link:

1. **Punto-Punto** = un nodo è collegato direttamente ad un altro attraverso un link (canale) dedicato;
 - Ex. collegamento ethernet tra un host e uno switch.
2. **Broadcast** = il link è condiviso, potenzialmente diversi nodi possono trasmettere e/o ricevere contemporaneamente;
 - Ex. prime versioni di ethernet (tipologia a bus), comunicazione wireless.

I protocolli ad accesso multiplo stabiliscono un insieme di regole che ogni nodo deve seguire affinché la comunicazione avvenga con successo. In assenza di protocolli avremmo il problema delle **collisioni**, cioè due o più nodi che trasmettono contemporaneamente.

Consideriamo algoritmi di tipo distribuito che determinano come il canale viene condiviso, ad esempio determinano quando un determinato nodo deve trasmettere oppure cosa succede in caso di collisione. La comunicazione avverrà, sia per trasmettere dati che per regolare la trasmissione stessa, sullo stesso canale.

Protocollo ideale Consideriamo un canale ad accesso condiviso che è in grado di trasmettere ad un rate di R bps (bit per secondo).

Si desidera che:

1. quando un singolo nodo vuole trasmettere, può trasmettere ad una velocità di trasmissione pari ad R ;
2. quando ci sono M nodi che voglio trasmettere contemporaneamente, ogni nodo può trasmettere ad una velocità media di $\frac{R}{M}$;

3. completamente decentralizzato:

- non voglio nodi specializzati che coordinino la trasmissione (single point of failure);
- non voglio la sincronizzazione dei clock dei diversi nodi o degli slot.

4. semplice.

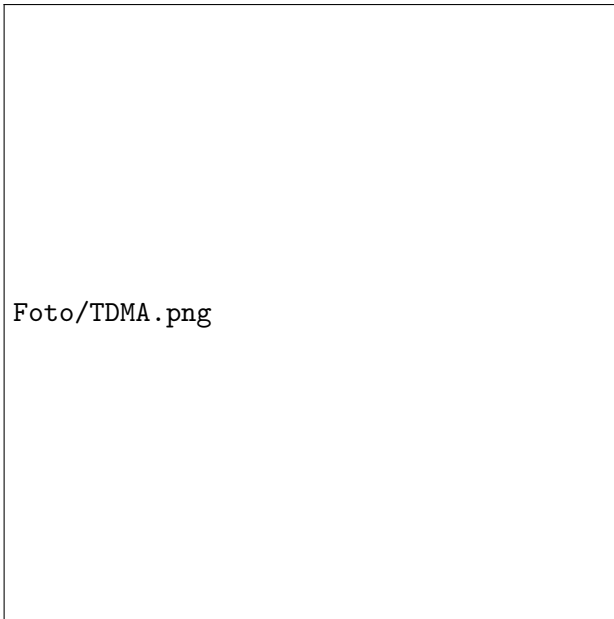
1.3.1 Protocolli a partizionamento del canale

Divido la capacità complessiva del canale in diverse parti che assegno ai diversi nodi.

TDMA Ogni nodo può accedere al canale solo in determinati intervalli di tempo. Ad ogni stazione viene assegnato un intervallo di tempo fisso in cui può trasmettere. Gli slot inutilizzati vengono persi, NON riutilizzati da altri nodi.

Esempio

Il time-frame viene suddiviso in un numero di slot pari al numero di nodi complessivi della rete.



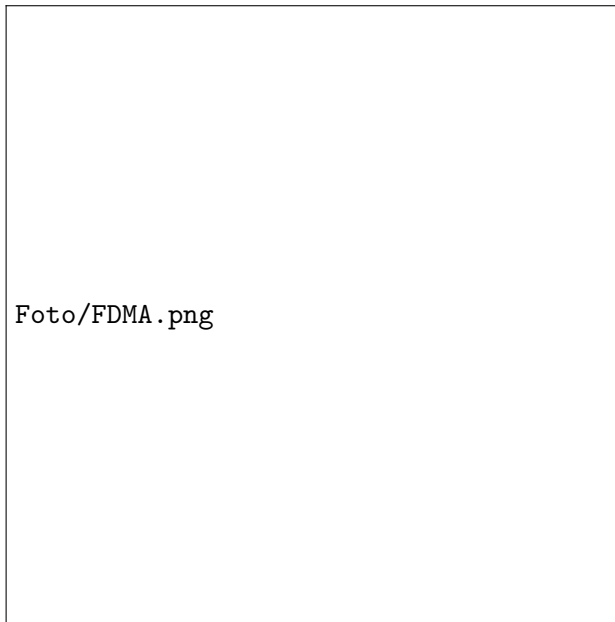
Proprietà ideali:

1. Soddisfatto;
2. Soddisfatto;
3. Non soddisfatto, è necessario mantenere gli stessi clock per sapere quando inizia uno slot;
4. Soddisfatto.

FDMA Viene partizionato il dominio delle frequenze. Ad ogni stazione viene assegnata una frequenza stabilita. Il tempo non utilizzato in frequenza viene sprecato. Posso non usare certe frequenze tra un intervallo ed un altro per evitare le sovrapposizioni di banda.

Esempio

Vengono partizionate le frequenze in base al numero di nodi presenti.



Proprietà ideali:

1. Non soddisfatto;
2. Soddisfatto;
3. Soddisfatto;
4. Soddisfatto;

1.3.2 Protocolli ad accesso casuale

Il canale non è diviso e si permette l'occorrenza di collisioni, nel caso di collisioni devo fare in modo di risolverla. Quando un nodo ha un pacchetto da inviare lo trasmette alla massima velocità, a priori non c'è coordinamento tra i nodi.

Slotted ALOHA Assumiamo che tutti i frame abbiano la stessa dimensione, il tempo è diviso in slot di uguale dimensione necessaria per trasmettere un intero frame. I nodi iniziano a trasmettere solo all'inizio dello slot. I nodi sono sincronizzati. Se 2 o più nodi trasmettono simultaneamente tutti i nodi della rete sono in grado di rilevare la collisione.

Quando un nodo ottiene un nuovo frame da inviare lo trasmette nel successivo slot, dopo aver trasmetto si mette in ascolto, se c'è un picco di energia viene a conoscenza della presenza di una collisione in questo caso ritrasmetto nel successivo slot con una probabilità p .

Esempio

All'inizio tutti i nodi vogliono trasmettere e avviene una collisione. Successivamente, dopo uno slot vuoto, i nodi 1 e 2 ritrasmettono e vanno in collisione. Allo slot successivo solo 2 trasmette ed ha successo. E così via finché tutti non hanno trasmesso.



Proprietà ideali:

1. Soddisfatta;
2. Non soddisfatta;
3. Non soddisfatta, perché è richiesta sincronizzazione;
4. Soddisfatta.

Durante le collisioni vengono sprecati degli slot. Un nodo è in grado di accorgersi subito della collisione ma deve aspettare un nuovo slot per trasmettere.

Efficienza

Un indicatore dell'efficienza è la proporzione di slot che hanno una comunicazione con successo rispetto a tutti gli slot disponibili su un intervallo di tempo lungo.

Supponiamo che N nodi vogliono continuare a trasmettere dei frame ed ognuno trasmette con probabilità p .

- La probabilità che un dato nodo abbia successo nella trasmissione è $= p(1 - p)^{N-1}$
- La probabilità che un qualsiasi nodo abbia successo è $= Np(1 - p)^{N-1}$
- Per la massima efficienza devo trovare p^* che massimizzi : $Np(1 - p)^{N-1}$
- Per molti nodi prendo il limite di $Np^*(1 - p^*)^{N-1}$ quando N tende a infinito.
- Efficienza massima $= \frac{1}{e} = 0.37$

Nella migliore delle ipotesi utilizzerò il canale il 37% del tempo.

CSMA (Carrier Sense Multiple Access) Ascolto prima di trasmettere (*carrier sense*), se il canale è libero allora trasmetto, altrimenti aspetto.

Esempio

Possono capitare delle collisioni anche ascoltando prima di trasmettere sul canale perché il segnale non si propaga immediatamente ovunque (*propagation delay*).

Foto/CSMA.png

CSMA/CD Protocollo utilizzato in ethernet. CSMA con il rilevamento della collisione (*collision detection*). Mentre trasmetto riesco a rilevare che è presente una collisione, appena riesco a farlo interrompo la trasmissione per ridurre lo spreco del canale. Facile da implementare su canali via cavo, difficile in wireless.

Esempio

Per velocizzare il fatto che tutti rilevino la collisione, appena qualcuno la rileva manda un segnale agli altri nodi intorno in modo tale che tutti rilevino la collisione.

Foto/CSMACD.png

Algoritmo di Ethernet

1. Ethernet riceve il datagramma dal livello rete sovrastante e crea il frame;
2. La scheda di rete ascolta il canale:
 - se è libero: inizia la trasmissione del frame;
 - se è occupato: aspetta finché il canale non si libera e poi trasmette.
3. Se sono riuscito a trasmettere l'intero frame senza collisioni ho finito;
4. Se viene rilevata un'altra trasmissione mentre invio, smetto di trasmettere e mando un segnale di *abort* (*jamming signal*) agli altri nodi;
5. Dopo aver abortito devo ritrasmettere e utilizzo un algoritmo di **binary (exponential) backoff**:
 - dopo l' m -esima collisione scelto un numero casuale K nell'intervallo $\{0, 1, 2, \dots, 2^{m-1}\}$. Ethernet aspetta $k \cdot 512$ bit times (tempo necessario per trasmettere un bit) e poi ritorno al punto 2.
 - Più collisioni ci sono maggiore sarà l'attesa per ritrasmettere.

1.3.3 Protocolli a turni

Ogni nodo può trasmettere in maniera dedicata ma solo in determinati turni.

Polling Controllore centralizzato, ciclicamente richiede ad ogni nodo se ha bisogno di trasmettere se si dedica un intervallo di tempo per trasmettere altrimenti passa al successivo.

I contro di questo protocollo sono: l'overhead del polling (pochi byte), latenza perché devo aspettare che il controllore faccia il giro e il single point of failure del controllore.

Questo protocollo è utilizzato in Bluetooth.

Token-passing Invece di avere un master ho un *token*. I nodi sono disposti ad anello, in modo ciclico. Il token passa al successivo nodo che conosce il suo successore. Chi ha il token può trasmettere per un intervallo di tempo fisso, costante e limitato.

I contro di questo protocollo sono: l'overhead del token, latenza per l'attesa del token e single point of failure (ciascun nodo quando ha il nodo diventa un punto di fallimento).