

# 1 - Proprietà delle Reti

27 settembre 2021

## Le distanze

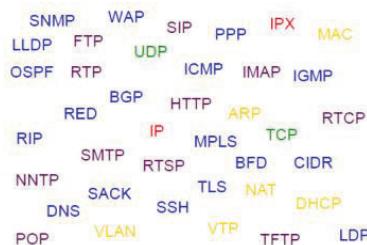
Le reti di calcolatori interessano tutte le interconnessioni tra circuiti e dispositivi nelle quali la **distanza** assume valori significativi. Non sono reti di calcolatori i circuiti integrati o le interconnessioni tra le parti di una stessa unità di elaborazione, in quanto le distanze sono nell'ordine dei millimetri o centimetri, mentre sono reti tutte quelle connessioni nell'ordine dei metri. Considerando 5 **kilometri** come distanza significativa di riferimento, possiamo distinguere due tipi fondamentali di rete:

- La prima (distanza sotto i 5 km) è l'interconnessione tra le risorse di calcolo in uno stesso edificio o edifici vicini, e prende il nome di **local area network (LAN)**;
- La seconda (distanza maggiore di 5 km) è l'interconnessione in una più vasta area geografica, chiamata **wide area network (WAN)**.

## Terminologia

La terminologia nelle reti di calcolatori è di fondamentale importanza. Molte tecnologie sono identificate da acronimi e può risultare difficile memorizzare la grande quantità di informazioni contenuta in essi.

L'obiettivo di questo corso è fornire una giusta chiave di lettura per interpretare tutti i termini tramite una comprensione basilare della struttura delle reti di calcolatori.



A proposito di terminologia, consideriamo le seguenti definizioni preliminari:

- Un **bridge** o **switch** è un dispositivo che connette tra loro più reti LAN, generando una rete più ampia; generalmente, le *lan* connesse tra loro appartengono alla stessa unità operativa;
- Un **gateway** è un dispositivo che connette tra loro più LAN, creando una WAN; la differenza rispetto ai *bridge* sta nel livello di astrazione a cui si lavora, che nel caso del gateway è maggiore;
- Una **sottoretete di comunicazione** in una WAN è ciò che collega tra loro le macchine della rete adibite all'esecuzione di programmi per gli utenti; queste macchine vengono dette **host**. Il compito della sottoretete è di trasportare messaggi da host a host, i quali non sono macchine utente ma bensì macchine che consentono la trasmissione dei dati realizzabile fisicamente.

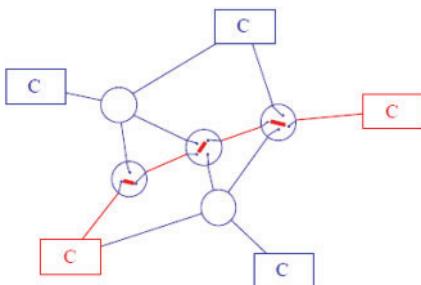
## Tipo di scambio di informazioni

L'ambito di interesse di questo corso riguarda lo **scambio di messaggi**; non siamo invece interessati allo scambio di memoria, condivisa e in parallelo. (dropbox). Il livello di comunicazione parallela nelle reti è infatti **bassissimo** (a differenza del funzionamento di qualsiasi calcolatore elettronico).

Ciascun tipo di comunicazione in una rete di calcolatori avviene per via **seriale**, sebbene sia possibile ottenere, a livelli di astrazione maggiori, le stesse funzioni delle memorie condivise in remoto (scambio di messaggi in serie).

## La commutazione (switching)

Da definizione, la commutazione è *l'insieme delle tecniche elettroniche e metodologie utilizzate per far dialogare tra loro calcolatori connessi in rete*. Distinguiamo due tipi di commutazione, a circuito

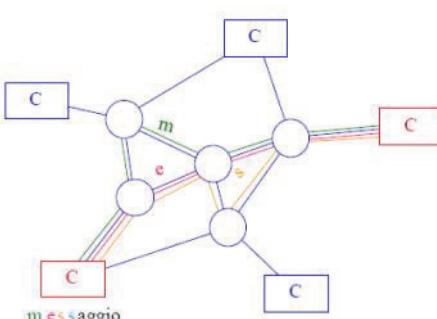


La **commutazione di circuito** funziona in tutto e per tutto emulando gli interruptori dei circuiti elettronici: per trasferire un messaggio, vengono realizzate delle connessioni *ad hoc* sfruttando i nodi della rete, attraverso i quali è possibile connettere tutti i terminali della rete. Si presuppone quindi l'esistenza di più "centraline" che vadano a formare un sistema intermedio detto **sottoretete di comunicazione**.

Il problema principale di questa tecnologia è proprio nel dover realizzare dei circuiti fisici tra due terminali qualsiasi: poiché nella realtà le connessioni avvengono nell'ordine di millisecondi e soprattutto più terminali richiedono la stessa connessione verso un singolo server, è fisicamente irrealizzabile una rete globale di calcolatori con commutazione reale a circuito.

## Mezzi trasmittivi

L'infrastruttura fisica si è evoluta nel corso degli anni, passando dai cavi coassiali e telefonici alla fibra ottica. Esiste persino uno standard definito dalla IEEE (*fc 1149, IP datagrams over avian carriers*) che definisce il trasporto di informazioni tramite piccioni viaggiatori! Questo aspetto scherzoso serve in realtà a far riflettere sulla semplicità concettuale di qualsiasi rete di calcolatori: lo scambio di messaggi avviene **facendo uso di scatole chiuse**, tanto che quella scatola potrebbe contenere (o rappresentare) qualsiasi cosa, l'importante è che funzioni!

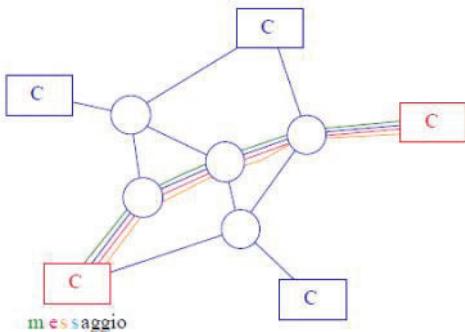


La **commutazione di pacchetto a datagramma** risolve le difficoltà realizzative di una rete a circuito applicando una delle procedure fondamentali della comunicazione nelle reti: la **separazione del messaggio**. Le "parti" del messaggio (definiti datagrammi o **pacchetti**) vengono spediti nella rete singolarmente, che si occuperà poi di instradare ciascun pacchetto a seconda della disponibilità della rete e delle tempistiche di comunicazione.

Non importa quale strada percorra ciascun pacchetto, poiché alla fine il destinatario riceverà sempre tutti i pacchetti, i quali, ricomposti dai sistemi di rete, riformeranno il messaggio spedito dal mittente. Una rete con commutazione a datagramma è strutturata per questo scopo e per garantire una comunicazione comprensibile.

La commutazione a circuito è invece adatta in altri ambiti, come quello telefonico, in cui il **canale** di comunicazione deve rimanere a disposizione **esclusiva** di tutti e due i terminali per tutta la durata dello scambio di messaggi.

**commutazione di circuito: adatta per ambiti in cui canale di comunicazione rimane in disposizione esclusiva**



La commutazione a datagramma ha uno svantaggio: i pacchetti, non seguendo tutti la stessa strada, complicano non poco la trasmissione di un singolo messaggio, che mette in gioco più parti della rete che non necessariamente andrebbero utilizzate, poiché non costituiscono la strada più breve. Nella **commutazione di pacchetto a circuito virtuale**, oltre a far percorrere a tutti i pacchetti la stessa strada, si elimina anche la necessità di dover risequenziare i pacchetti una volta ricevuti dal destinatario. Passano più pacchetti nello stesso filo.

### In conclusione, qual è il miglior tipo di commutazione?

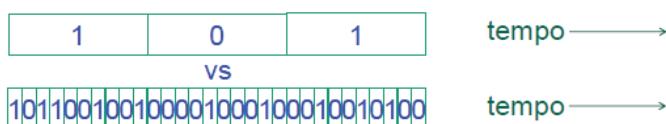
*Nel mondo è universalmente utilizzata la comunicazione di pacchetto a datagramma, di più facile realizzazione e di maggiore versatilità rispetto alla commutazione a circuito virtuale (seppur quest'ultima sia più vantaggiosa).*

### Velocità e banda di comunicazione

Quando si parla di velocità nelle reti non è possibile quantificare una vera e propria "rapidità" con cui i bit vengono effettivamente trasmessi; a prescindere dalla tecnologia della rete, infatti, è impossibile distinguere la velocità di trasmissione del singolo bit in base alla tecnologia utilizzata, anche a distanze di migliaia di chilometri.

*Se, ad esempio, trasmettessimo due bit trasmessi da Roma a Tokyo simultaneamente utilizzando però ciascuno due tecnologie diverse (ad esempio, una a 1 Gbit/sec e l'altra a 10 Gbit/sec), questi arriverebbero nello stesso istante.*

Il tempo di trasmissione dipende solo dalla distanza fisica e dalla velocità della luce ed è definito **delay o ritardo**. Parlare quindi di velocità di una data tecnologia, quindi, vuol dire in realtà parlare di **bandwidth o larghezza di banda**: due tecnologie a confronto potranno inviare simultaneamente un diverso numero di bit, in base alla loro larghezza di banda.



Nella pratica esiste una regola non scritta, secondo la quale *è più facile comprare più banda piuttosto che ottenere meno delay da una rete*. Quando aggiorniamo il nostro contratto con il provider di rete ad una rete su fibra ottica a 1 Gigabit al secondo, infatti, non stiamo ottenendo meno ritardo rispetto ad un utente di una vecchia rete telefonica a 56 kilobit al secondo (il cui limite è imposto dalla velocità della luce), ma stiamo comunque ottenendo una larghezza di banda infinitamente più grande, che, in un certo senso, dà la sensazione di avere una "connessione più veloce".

### Gestione delle risorse

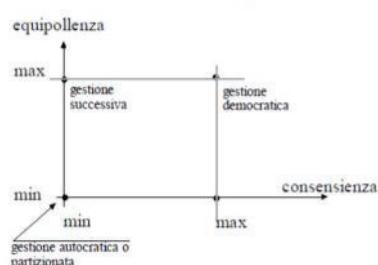
Una rete di calcolatori può essere vista come un insieme di risorse interconnesse, il cui controllo è articolato in varie attività, che vanno dalla **gestione** dei diritti di accesso all'esecuzione delle operazioni disponibili passando per il **segnalamento** delle stesse. Per ogni risorsa ci sono uno o più **gestori** ed è possibile identificare diversi tipi di gestione.

Una gestione di **una risorsa** da parte di un solo gestore è detta **autocratica**, mentre quella di più gestori è detta **multilaterale**:

- **Partizionata** (un processo diverso per ogni attività di gestione);
- **Successiva** (più processi eseguono a turno ogni attività);
- **Replicata** (tutti i gestori partecipano a ciascuna attività);
  - o È detta **democratica** quando ogni attività ha luogo tramite una cooperazione alla pari dei gestori (mutuo consenso).

Due qualità della gestione di una rete sono la **consenzienza** e l'**equipollenza**: la prima è alta se il numero dei gestori che partecipano ad ogni istanza di un'attività è alto, mentre la seconda quantifica il grado di uguaglianza nella responsabilità di gestione.

### Lo spazio equipollenza - consenzienza – analisi della tolleranza ai guasti



### velocità – banda di comunicazione

- **LAN** 10 Mb/sec - 100 Gb/sec
- **WAN** 64Kb/sec - 10 Gb/sec - 50 Gb/sec – 100 Gb/sec. – 200/400 Gb/sec.

Considerare in questa scelta la difficoltà di ripetere lo stesso percorso per tutti i pacchetti considerata la pluralità di soggetti che entrano in gioco in qualità di proprietari/gestori dei vari tratti di rete per i quali passano i pacchetti.

### gestione delle risorse e guasti

- **resistenza ai guasti:**
  - alta se è alto il numero di gestori che partecipano ad ogni istanza di una certa attività, con alto grado di uguaglianza nella responsabilità di gestione

### Il problema degli standard

*Problema degli standard: alcuni sono imposti dalle case costruttrici, altri da convenzioni internazionali; nonostante ciò, la standardizzazione nelle reti di calcolatori ha livelli molto più alti di molte altre tecnologie a livello mondiale (meccaniche, elettriche ed elettroniche).*

## 2 - Modello ISO-OSI a strati

30 settembre 2021

### Comunicazione multilivello

Le reti di calcolatori sono generalmente organizzate in livelli, ognuno costruito su quello inferiore. Questo modello è detto **pila protocolle a strati** o **modello OSI** (Open Systems Interconnection), e si tratta di uno standard definito dal consorzio **ISO** (International Standard Organization) che costituisce un modello primo passo verso la standardizzazione internazionale dei protocolli usati nei vari livelli.

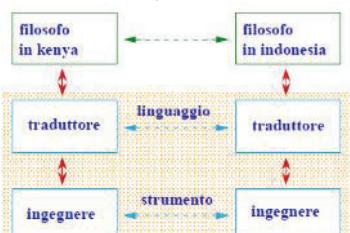
|                         |
|-------------------------|
| <b>7. Applicazione</b>  |
| <b>6. Presentazione</b> |
| <b>5. Sessione</b>      |
| <b>4. Trasporto</b>     |
| <b>3. Rete</b>          |
| <b>2. Data Link</b>     |
| <b>1. Fisico</b>        |

Ogni strato corrisponde ad un livello di astrazione e offre funzioni ben definite; lo scambio di dati tra livelli è minimizzato.

Tale modello contribuisce a rendere una maggiore astrazione della rete, così come avviene nelle interfacce utente dei sistemi operativi; in un sistema operativo, infatti, uno strato fornisce dei servizi allo strato soprastante, servizi vengono astratti gradualmente man mano che si passa da uno strato ad uno superiore. Con l'astrazione dei servizi, la macchina aumenta di usabilità e risulta più comprensibile all'utente finale.

### L'esempio dei due filosofi

Due filosofi si trovano in punti diversi del mondo, hanno entrambi la stessa corrente di pensiero ma parlano diverse lingue; non potendo comunicare direttamente, i due filosofi si rivolgono a due traduttori, che traducono i messaggi in una lingua comune per comunicare tra di loro. Essi a loro volta si rivolgono a due ingegneri che tra di loro riescono a comunicare attraverso la tecnologia.

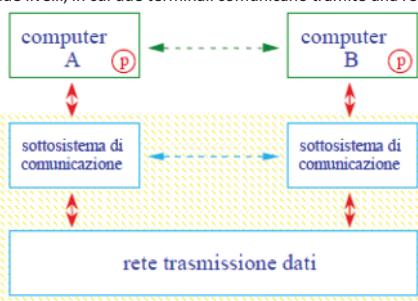


- I traduttori possono avere come lingua comune il francese, il tedesco o l'inglese;
- Gli ingegneri potrebbero usare l'alfabeto morse o il codice delle bandiere;
- Filosofi e traduttori devono essere d'accordo sulla modalità di scambio di informazioni, così come i traduttori e gli ingegneri.

Ciò che accade nel rettangolo punitato è l'equivalente di una rete di calcolatori. La comunicazione non dipende dalla *lingua* adottata dai due traduttori, l'importante è che sia comune e condivisa da entrambi; lo stesso discorso vale per gli ingegneri. I due filosofi e traduttori inoltre devono essere d'accordo sulle *modalità di scambio delle informazioni*, anche i traduttori e gli ingegneri.

Il protocollo è sostanzialmente un accordo fra i partecipanti di una conversazione su come la conversazione deve procedere.

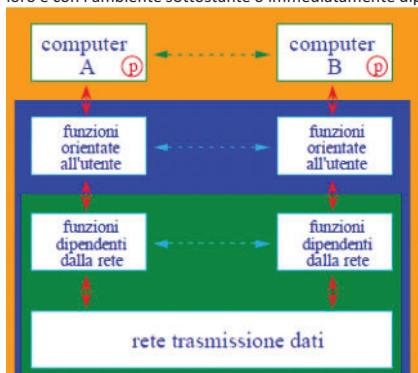
Partendo dall'esempio dei due filosofi, possiamo quindi modellare un'architettura di rete primitiva a due livelli, in cui due terminali comunicano tramite una rete a due strati:



Il traduttore è un sottosistema di comunicazione, mentre l'ingegnere è l'infrastruttura di trasmissione dati che si occupa di recapitare fisicamente il messaggio. Distinguiamo inoltre la comunicazione tra utenti (o processi) e la comunicazione tra computer.

I processi sono programmi in esecuzione nel computer

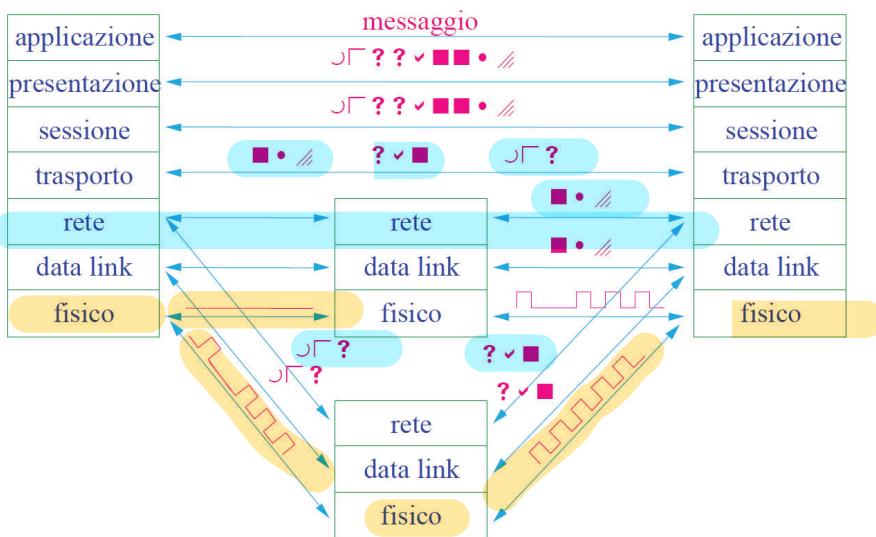
È possibile identificare quindi tre livelli o ambienti nella rete, che in un certo modo comunicano tra loro e con l'ambiente sottostante o immediatamente dipendente da esso:



- Nell'**ambiente globale**, il computer A riesce a "vedere" il computer B grazie agli ambienti sottostanti.
- Nell'**ambiente OSi** vengono offerte funzioni orientate all'utente e direttamente accessibili da questi;
- Nell'**ambiente di rete** vengono svolte funzioni dipendenti esclusivamente dalla rete, i cui parametri vengono comunicati dai livelli soprastanti.

## Gli strati ISO-OSI agli estremi e nei sistemi intermedi

Analizziamo ora i singoli livelli e come cambia il messaggio scendendo nei livelli della pila:

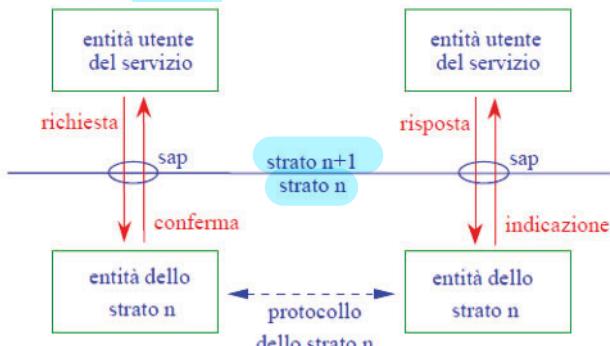


Dalla presenza di più di un livello di rete nel grafico possiamo vedere come la rete comunichi tramite la commutazione a datagramma.

In conclusione, i concetti fondamentali del modello sono tre: **strati, interfacce** tra strati consecutivi e **protocolli**, ovvero i linguaggi parlati su ciascun livello.

### Entità, sap e primitive di servizio

Ciascun livello di rete è composto da diverse entità software o hardware, ciascuna delle quali fornisce un servizio allo strato successivo. Considerando quindi un **insieme di entità**, ciascuna n-entità comunica con l'entità n+1 dello strato n+1, fornendo un n-servizio; non tutte le funzioni eseguite dall'n-strato si traducono in servizi per lo strato n+1, ma quelle che vengono fornite attraversano tutte un punto logico detto **service access point (sap)**. Il sap costituisce quindi la frontiera tra ciascuno strato della rete e mette in comunicazione due strati adiacenti. Indipendenza funzionale: ogni strato è definito in modo del tutto indipendente dallo strato inferiore, l'unico punto di contatto è l'interfaccia.



L'insieme delle n+1 entità che sono connesse allo stesso n-sap è detto **n-utente** dell'n-servizio, mentre l'insieme delle n+1 entità che cooperano per l'esecuzione delle funzioni interne all'n-strato è detto **n-fornitore**. Tutte le possibili interazioni tra utente e fornitore sono dette **primitive di servizio**, indicate nella figura dalle frecce rosse.

|                    |                    |
|--------------------|--------------------|
| <b>Richiesta</b>   | Utente > fornitore |
| <b>Indicazione</b> | Fornitore > utente |
| <b>Risposta</b>    | Utente > fornitore |
| <b>Conferma</b>    | Fornitore > utente |

### I pacchetti

Come abbiamo visto, la commutazione della rete è a **datagramma**, che vuol dire che i dati scambiati tra due terminali vengono suddivisi in pacchetti. Ciascun pacchetto dati viene arricchito da **preamboli** scendendo via via nei livelli della rete; tali preamboli sono chiamati **header**, e sono usati solitamente per comunicare con il corrispondente livello dell'architettura destinataria. Ad esempio, il **ph** (presentation header) riporta nel pacchetto il metodo di crittazione usato dall'applicazione.

Il pacchetto, dall'alto, può essere quindi visto come un insieme di buste, l'una dentro l'altra, e tale metodo viene applicato per trasmettere anche la più piccola delle informazioni; la quantità di informazioni addizionali (e della lunghezza totale del pacchetto, o **frame**) è determinata dal numero di livelli impiegati dal sistema.

#### Riepilogo terminologia

**sap: Service Access Point** (indirizzo)

**pdu: Protocol Data Unit** (pacchetto o trama o frame), composto da:

- **SDU: Service Data Unit** (talvolta chiamato **payload**)
- **PCI: Protocol Control Information** (header della busta)

 $(n\text{-pdu} = n\text{-PCI} + n\text{-SDU})$

7. Il **livello di applicazione** maschera tutta la complessità della rete. È il livello di astrazione più alto ed è ciò che l'utente visualizza sullo schermo, e permette di mettere in comunicazione macchine distanti che "non parlano la stessa lingua". Non potendo comunicare direttamente il messaggio dovrà viaggiare attraverso tutti i livelli della rete;

6. Il **livello di presentazione** si occupa di favorire una sintassi comune tra i due sistemi (**codifica**); una volta stabilito ciò, può occuparsi della sicurezza del messaggio e trasferisce il messaggio criptato al livello inferiore ("traduce" il messaggio);

5. Il **livello di sessione** consente di stabilire, chiudere, temporizzare o sincronizzare una sessione tra due sistemi (**sincronizzazione fra processi obbligativi**). Stabilita una sessione, il messaggio deve essere preparato al trasferimento;

4. Il **livello di trasporto** si occupa di prendere il messaggio e **frammstrarlo in pacchetti**. È il livello "del quale il programmatore si fida", infatti esso garantisce che il messaggio non venga modificato durante il trasporto. Tutti i pacchetti, infatti, da qui in poi non saranno ulteriormente modificati;

I livelli da 7 a 4 sono i livelli *End-to-end* detti **livelli degli host**, mentre i livelli da 3 a 1 si trovano su tutti i sistemi e sono i **livelli dei mezzi** (sistemi intermedi).

3. Il **livello di rete** è collegato agli altri livelli di rete dei vari sistemi e si occupa di "passare la palla" (ovvero i pacchetti) tra un sistema e l'altro, fino a raggiungere il sistema destinatario. Appaiono quindi dei **livelli intermedi**: ogni livello di rete ha accesso alla mappa di rete e indirizza i pacchetti secondo la strada più breve per ciascun pacchetto, ma non riesce a **comunicare** la strada al pacchetto. I pacchetti arrivano quindi al livello fisico, facendo un piccolo salto;

2. Il **livello di Data Link** nasce per fronteggiare i malfunzionamenti del **livello fisico**, che è molto sensibile ai disturbi esterni nell'atto di trasmissione del segnale. **Approfondiremo questo livello più avanti**;

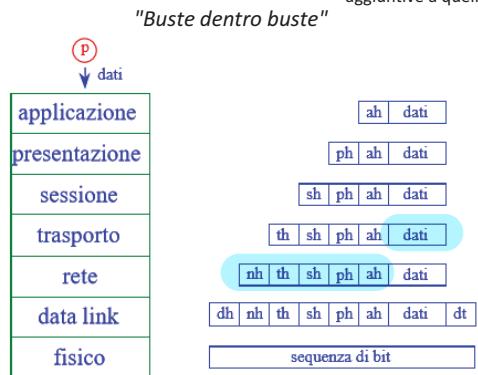
1. Il **livello fisico** si occupa finalmente di **spostare i pacchetti** da un sistema all'altro. Per rendere possibile fisicamente questa azione, i pacchetti vengono convertiti da sequenze di bit a forme d'onda e tali segnali vengono spediti nella rete. I **livelli fisici intermedi** si occupano semplicemente di reinidirizzare la forma d'onda verso il sistema di arrivo, mentre il livello fisico in ricezione si occupa di **ricostruire il segnale in bit** partendo dalla forma d'onda quadrata che viaggia sulla rete fisica, sia essa composta da cavi di rame, di fibra oppure sia essa una rete senza fili.

Entità n e n+1, dentro gli strati ci sono dei software. L'**interfaccia** fra strati adiacenti è il **sap**. Le entità si scambiano informazioni. Quando n+1 chiede al livello n di portare a destinazione un'informazione gli fa una **richiesta**.

Il passaggio inverso è quello di **indicazione**.

Si vuole che n+1 dia un riscontro di ricezione, arriva una **risposta** (non in tutti i protocolli) e quando viene fornita a chi ha mandato il messaggio diventa una **conferma**.

ah - application header, contengono dati di supporto  
ph - presentation header, informazioni sul metodo di encryption  
Ciascun livello aggiunge informazioni aggiuntive a quelle che deve trasportare.



## Approfondimento sui livelli 1-4

4 ottobre 2021

### • Strato fisico (1)

Lo strato fisico si interfaccia direttamente con i mezzi fisici di trasmissione, offrendo allo strato superiore una comunicazione indipendente dal particolare mezzo trasmisivo (ricordate lo standard dei piccioni viaggiatori?). I servizi che deve fornire allo strato di collegamento (sopra) sono:

- Gestione e identificazione della connessione fisica;
- Trasmissione delle unità dati (pdu);
- Consegnà in sequenza delle unità dati;
- Notifica di malfunzionamenti.

### • Strato di collegamento o Data Link (2)

Lo strato di Data Link fronteggia i malfunzionamenti dello strato fisico (rivelazione e correzione degli errori). I servizi che fornisce allo strato di rete sono:

- Trasferimento di (pdu) informazioni senza vincolo di formato, codice o contenuti;
  - Si occupa quindi di collegamenti brevi, da nodo a nodo;
- Selezione di una certa qualità di servizio (trasmissione veloce, ma instabile, oppure lenta, ma garantita);
- Utilizzo di due code nelle due direzioni, i pacchetti vengono messi in coda

### • Strato di rete (3)

Lo strato di rete conosce la topologia della rete e permette l'instradamento dei pacchetti. Se connesso, esso instaura, mantiene e rilascia le connessioni di rete. I servizi che fornisce sono:

- Trasferimento di informazioni da estremo a estremo;
  - Si occupa di collegamenti lunghi, dal sistema di partenza a quello di arrivo
- Selezione di una certa qualità (tempo di attraversamento, disponibilità) - non sempre disponibile;
- Scelta del tipo di commutazione: circuito, pacchetto datagramma, circuito virtuale

*La comunicazione in modalità connessa è diversa dalla semplice comunicazione di un messaggio, come potrebbe essere l'invio di una cartolina; qui infatti viene mantenuto un dialogo per tutta la durata del tempo di connessione.*

### • Strato di trasporto (4)

Lo strato di trasporto colma defezioni e fluttuazioni del grado di servizio delle connessioni di rete, e ottimizza l'uso della rete dal punto di vista dei costi. È il primo strato end-to-end (ovvero risiede solo nei nodi terminali). I servizi forniti allo strato di sessione sono:

- Instaurazione di una connessione;
- Trasferimento dati e gestione della connessione;
- Rilascio della connessione;
- Sincronizzazione tra i due sistemi per mezzo di conferma.



Sul livello 3 abbiamo messo in piedi una connessione con qualcun altro, che permette lo scambio di pacchetti. Il livello 4 può a sua volta essere connesso e può instaurare una o più connessioni tra noi e l'altro sistema. A livello applicativo, dal livello 4 in su, possiamo svolgere più attività e ognuna di queste avrà la sua connessione di trasporto.

Una connessione di trasporto ha bisogno di una connessione di rete per funzionare. A tal punto, nulla vieta ad una singola connessione di trasporto di appoggiarsi a più di una connessione di rete, trasmettendo i pacchetti nella rete in parallelo.

Un po' di banda da una parte e un po' di banda dall'altra. Poder utilizzare varie connessioni di livello più basso per una stessa connessione di livello alto.

N.B.: per tutti i livelli superiori al primo sono pensabili due modalità operative, che danno luogo a due tipi di servizi: **connessi** o non **connessi**.

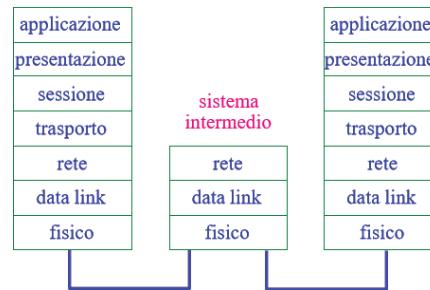
- Un esempio di servizio connesso è una telefonata, mail;
- Un esempio di servizio non connesso è una cartolina;

In generale, un protocollo connesso è più affidabile, mentre un protocollo non connesso è più efficiente.

| Protocollo   | Affidabilità | Efficienza |
|--------------|--------------|------------|
| Connesso     | +            | -          |
| Non connesso | -            | +          |

Connesso: quando ti rivolgi al livello inferiore, quando si stabilisce un dialogo, quando si dice esplicitamente che è finito.

Non connesso: è molto veloce ma poco affidabile, non ci sono riscontri esplicativi



### Tecniche usate nei protocolli di livello 2, 3, 4

- Meccanismo dei riscontri (**acknowledgement, ack**);
- **Piggybacking** (tempo massimo di attesa);
- Tecniche di controllo a finestra (numerazione dei pacchetti);

### Protocolli connessi e non

| Livello | Lan           | Wan     |
|---------|---------------|---------|
| 2       | No            | Si e no |
| 3       | Sempre più no | No      |
| 4       | Si            | Si      |

vuole un servizio affidabile

Può accadere che, in un collegamento in cui è presente un sistema intermedio, il primo data link e quello intermedio siano connessi e quello intermedio con l'ultimo non lo siano? Si (attenzione, dal punto di vista fisico sono comunque connessi tutti e tre)

### Svantaggi della non-connesione

Quando due protocolli di livello 3 non sono connessi, l'indirizzo del destinatario deve essere specificato in ogni pacchetto che viene scambiato tra i due sistemi, e l'ordine di arrivo non è garantito.

### connessioni di trasporto e di sessione

#### connessione di sessione

#### connessioni di trasporto

#### connessioni di sessione

#### connessione di trasporto

tempo

Servizio di posta elettronica attivo sul computer. Quello che percepiamo è un servizio sempre attivo. Cambiamo luogo e cambiamo wifi (accesso del provider).

Connessione server di google, prima lo uso per posta elettronica, poi per un'altra cosa e poi posta elettronica di nuovo.

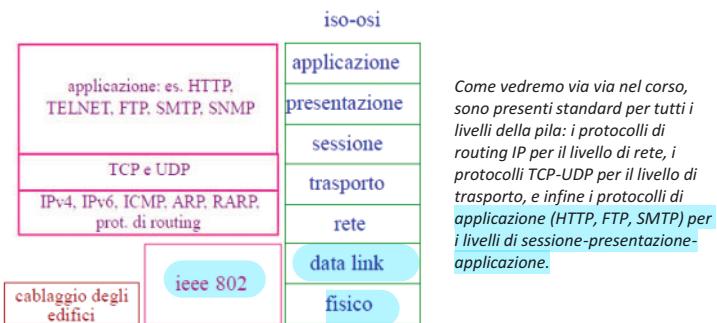
## Gli strati vicini alle applicazioni (5-7)

Mentre gli strati finora visti seguono rigidamente la filosofia della pila a strati secondo cui ciascun livello è in serie con il prossimo, gli strati più alti della pila solitamente possono essere considerati anche paralleli tra loro. Gli strati vicini alle applicazioni si occupano principalmente dell'accesso alla rete da parte dell'utente e dell'applicazione e della gestione delle interfacce di comunicazione.

- Lo **strato di sessione (5)** permette la **sincronizzazione del dialogo** tra due processi presenti su due calcolatori della rete;
- Lo **strato di presentazione (6)** permette lo scambio di messaggi tra calcolatori indipendentemente dalla sintassi della trasmissione, o dai metodi di accesso alla rete;
- Lo **strato di applicazione (7)** è l'interfaccia tramite cui l'utente **accede alla rete**, l'effettiva applicazione di rete, sia essa un client di posta, un browser web, un client p2p, eccetera..

## Gli standard per le reti locali

Sono stati definiti diversi standard da una commissione internazionale detta **IEEE (Institute of Electrical and Electronic Engineers)**, tra cui uno standard dedicato alla definizione dei protocolli di rete per le reti locali, detto IEEE 802. Uno standard, nelle reti di calcolatori, può essere comparato agli standard utilizzati nelle infrastrutture fisiche di rete, come ad esempio il cablaggio degli edifici.



*Come vedremo via via nel corso, sono presenti standard per tutti i livelli della pila: i protocolli di routing IP per il livello di rete, i protocolli TCP-UDP per il livello di trasporto, e infine i protocolli di applicazione (HTTP, FTP, SMTP) per i livelli di sessione-presentazione-applicazione.*

IEEE 802 riguarda il livello 1 e 2 della pila iso-osi.

## 3 - IEEE 802.2 (Livello Data Link)

7 ottobre 2021

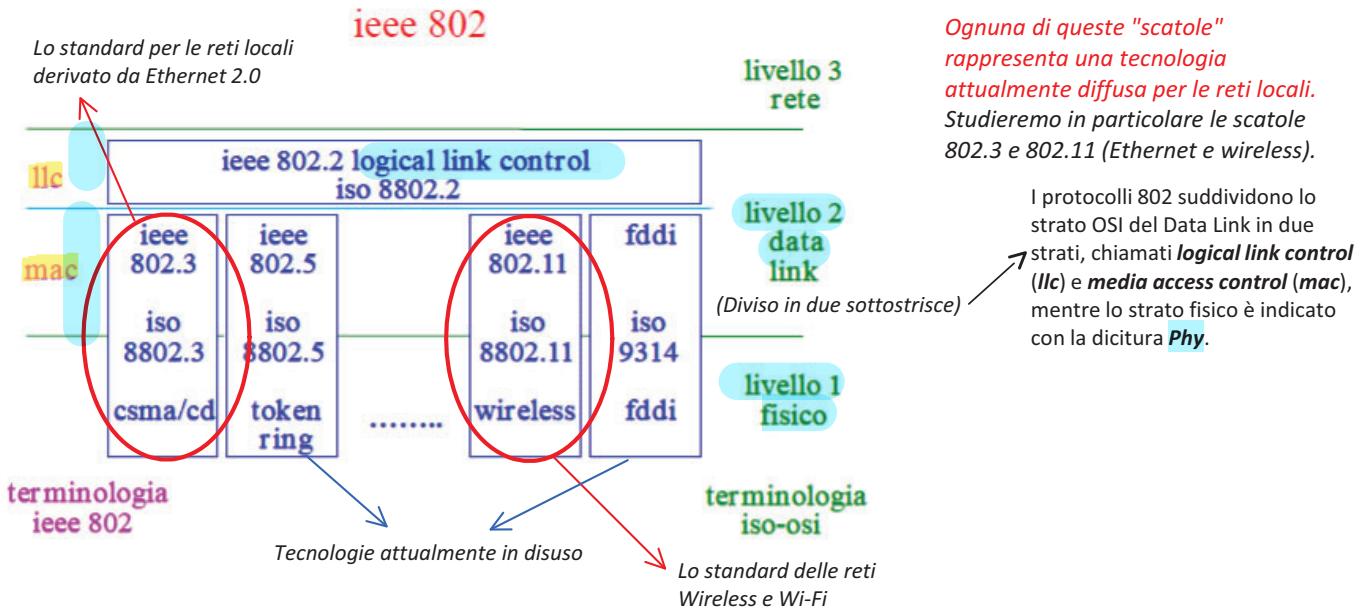
### Il progetto IEEE 802 per le reti locali

Come abbiamo detto, per ciascuno strato della rete sono stati proposti e adottati, negli anni, diversi standard. L'IEEE 802 LAN/MAN è una commissione dell'IEEE che è stata preposta per sviluppare standard per le reti locali (**LAN**), per le reti personali (**PAN**) e per le reti metropolitane (**MAN**); più precisamente gli standard 802 sono quelli dedicati a reti con pacchetti di lunghezza variabile o fissa e alle reti isocrone, nelle quali i pacchetti sono spediti su base temporale periodica. Nel nostro studio, lo standard 802 definisce i livelli 1 e 2 della rete locale, il livello fisico e il livello Data Link.

Alcune componenti del progetto IEEE 802:

|  |  |
|--|--|
| 802.1  | Higher layer and management  |
| 802.2  | Logical link control ( <b>llc</b> )  |
| <u>802.3</u><br>- 802.3u<br>- 802.3z<br>- 802.3ae, 802.3ak | Carrier sense, multiple access, collision detection ( <b>CSMA-CD</b> )<br>Fast ethernet<br>Gigabit ethernet<br>10 Gigabit ethernet |
| 802.5  | Token ring   |
| <b>802.11</b>  | <b>Wireless lan (wlan)</b>   |

I primi due livelli sono specifiche di tecnologie utili per tutte le tipologie di reti locali e si occupano dello strato alto del modello ISO-OSI.



### Il sottolivello mac (Media Access Control)

È il sottolivello inferiore del Data Link, ed è specifico per ogni tipo diverso di lan (a differenza dell'**llc**, che, come vedremo, è lo stesso per tutte le lan IEEE 802); lo scopo del mac è quello di disciplinare l'accesso multiplo di molteplici nodi ad un unico canale di comunicazione condiviso, evitando o gestendo l'occorrenza di collisioni. Assumendo infatti che esista un unico canale condiviso, la condivisione di questo da parte di più dispositivi implica la soluzione di due problemi:

1. In **trasmissione**, occorre verificare la disponibilità del canale e soluzione di eventuali conflitti (*a tal scopo si usano degli algoritmi distribuiti, eseguiti contemporaneamente su più sistemi*);
2. In **ricezione**, determinazione del destinatario e del mittente, tramite la presenza di **indirizzi mac** che consentano trasmissioni e che distinguono i computer in una rete:
  - a. **single** (punto-punto) da un computer a un computer;
  - b. **multicast** (punto-gruppo di punti), da un computer a un gruppo di computer (es: a tutte le stampanti);
  - c. **broadcast** (punto-tutti i punti), da un computer a tutti i computer.

Facendo parte del livello Data Link, anche nel livello mac transitano i pacchetti della rete (**pdu**) provenienti dal livello llc e contenenti indirizzi progettati per tutti e tre i tipi di trasmissione.

*Nel mondo IEEE-802, il mezzo trasmissivo è condiviso (così com'è condivisa l'aria in una stanza piena di portatili e cellulari connessi a internet).*

Vediamo la struttura di una le generica **mac pdu**, ossia i pacchetti che transitano a livello mac:



Ricorda che per ogni standard derivato da 802 esistono diverse definizioni del livello mac, e di conseguenza diversi formati delle mac pdu.

- Un indirizzo del destinatario, che specifica un *service access point* detto **mac-dsap**;
- Un indirizzo del mittente, che specifica anch'esso un sap detto **mac-ssap**;
- Una pdu proveniente dal livello llc, la **llc pdu** (il *logical link control* è quindi un sottopacchetto del pacchetto pdu del mac, una **busta dentro una busta**);
- Una sequenza dedicata alla **rilevazione di errori** detta **fcs**, acronimo di **frame check sequence**.

## Indirizzi mac

Ogni macchina capace di connettersi a una rete di calcolatori possiede il suo livello mac a cui è possibile riferirsi tramite un **indirizzo**; questo è **univoco** per ciascuna macchina connessa alla rete, ovvero non è possibile che due macchine abbiano lo stesso indirizzo mac.

Questa univocità degli indirizzi mac (valida anche per le macchine che non sono connesse ad internet ma bensì ad una rete locale) è necessaria per favorire l'interoperabilità degli host e il *plug and play* di dispositivi alla rete, evitando inoltre il problema che potrebbe insorgere nel caso in cui una macchina con indirizzo mac identico ad un'altra venisse connessa alla stessa rete in cui la seconda è già presente (si evitano le operazioni di controllo e riassegnazione del *mac address*!). Gli indirizzi mac conformi allo standard 802 sono lunghi **6 byte**, di cui 3 sono assegnati al **costruttore** dalla commissione IEEE, mentre i 3 restanti sono definiti dal costruttore per ciascuna macchina prodotta. Ogni indirizzo, infine, specifica ciascuna modalità di trasmissione (unicast: singola macchina, multicast: gruppi di macchine e broadcast: tutte le macchine).

Solitamente, un indirizzo mac è rappresentato in **notazione esadecimale (hex)**.

Prendendo come esempio l'indirizzo

08 00 2B 3C 07 9A

Possiamo distinguere i primi tre byte, che sono comuni a tutte le macchine dello stesso costruttore, dagli ultimi tre che variano da macchina a macchina e rendono di fatto l'indirizzo mac univoco universalmente.

## Il sottolivello LLC (*Logical Link Control*)

È il sottolivello superiore del livello Data Link, ed è comune a tutte le specifiche lan che adottano lo standard 802. Anche in esso viaggiano le **pdu**, provenienti dal livello di rete (che, ricordiamo, contengono i dati trasmessi dai livelli più alti), le quali vengono poi trasferite al livello mac che, come abbiamo visto, ne cura la trasmissione sul mezzo fisico prescelto. Viceversa, le pdu provenienti dal livello mac vengono elaborate secondo i criteri definiti dal protocollo, e quindi inoltrate al livello superiore.

Nel dettaglio, un pacchetto di livello llc avrà i seguenti campi:

- 1 byte di indirizzo del destinatario (**llc-dsap**);
- 1 byte di indirizzo del mittente (**llc-ssap**);
- 1 o 2 byte di campo **control**, che specificano tre diverse modalità di trasferimento;
- n byte di **info**, che non sono altro che i dati provenienti dal livello 3 della rete.

In funzione del valore del campo **control**, quindi, l'llc può fornire al livello di rete tre distinte modalità di servizio di trasferimento dati:

- Modalità *non connessa e senza conferma* (**logical data link**), che specifica una **unnumbered pdu**. È la modalità normalmente usata nelle reti locali, che permette tra le varie funzioni l'inizializzazione e la diagnosi della connessione;
- Modalità *connessa e con conferma* (**data link connection**), che specifica una **information pdu**;
- Modalità *non connessa ma con conferma* (**logical data link with acknowledgement**), che specifica una **supervisory pdu**, contenente informazioni di controllo del protocollo.

I **sap** del livello llc servono quindi a denotare i protocolli di livello superiore a cui sono destinati i pacchetti, consentendo la convivenza di protocolli di livello 3 diversi sulla stessa lan e sulla stessa macchina.

Per esempio, se gli llc-sap assumono come valore AA, allora il pacchetto conterrà un pacchetto di protocollo non standard; in questo caso (snap-pdu), dopo il campo control sarà presente un ulteriore campo protocol identifier di 5 byte.

**N.B.: questi llc-sap vengono attribuiti solo ai protocolli ufficialmente standard, mentre per protocolli non standard vengono attribuiti ulteriori pacchetti.**

### Mac pdu e llc pdu

In conclusione, nel livello data link definito da IEEE 802 viaggiano delle pdu in trasmissione e ricezione, che contengono specifiche sia per il livello mac che per il livello llc; è importante ricordare che il campo **llc-pdu** è ulteriormente diviso secondo le specifiche che abbiamo visto per il livello llc.

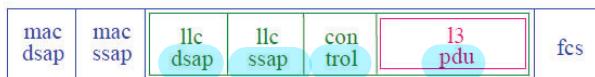
A sua volta, il campo **info** della **llc-pdu** contiene la pdu proveniente dai livelli superiori della rete.



FRAME CONTROL SEQUENCE, controlla che non si sia modificato qualche bit



payload - llc pdu  
pdu di livello 3



**Pacchetto di livello 3**  
(proveniente "dall'alto")

## 4 - Rilevamento delle Collisioni

11 ottobre 2021

### CSMA/CD e la rete Ethernet

Parlando del livello *mac*, abbiamo visto come nel caso delle reti *lan*, ovvero di canali di comunicazione condivisi globalmente su tutta la rete, sia possibile risolvere tutti i problemi di trasmissione (cioè la verifica di disponibilità del canale e la soluzione dei conflitti) tramite l'impiego di *algoritmi distribuiti*. L'algoritmo **csma/cd**, che sta per **carrier sense, multiple access with collision detection**, è l'algoritmo universalmente utilizzato nei livelli mac delle reti *Ethernet IEEE 802.3*. Vediamo il significato del nome:

- **Carrier sense** vuol dire che il *carrier* (ovvero la stazione che deve trasmettere) **ascolta** (*sense*) il bus, il mezzo trasmittivo, e **trasmette solo se questo è libero**;
- **multiple access** indica che, **se il canale è libero, tutte le stazioni possono trasmettere**. Potrà ovviamente verificarsi una collisione, ma ciò è contemplato;
- **collision detection** sta infatti per **rilevamento delle collisioni**: durante la trasmissione, la stazione trasmittente continua ad ascoltare il canale per poter eventualmente rilevare una collisione.

Quando una stazione trasmittente rileva una collisione arresta la sua trasmissione, essa trasmette una **sequenza di jamming**; grazie a questa, la stazione ricevente scarta i bit ricevuti, ovvero il frammento di pacchetto e i bit di jamming, mentre la stazione trasmittente, trascorso un tempo randomico multiplo di tempo stabilito (solitamente, per *Ethernet* a 10Mb/s, si tratta di 51,2 µs). È consentito un numero limitato di tentativi di trasmissione (solitamente 16), finché la connessione non fallisce e l'algoritmo termina.

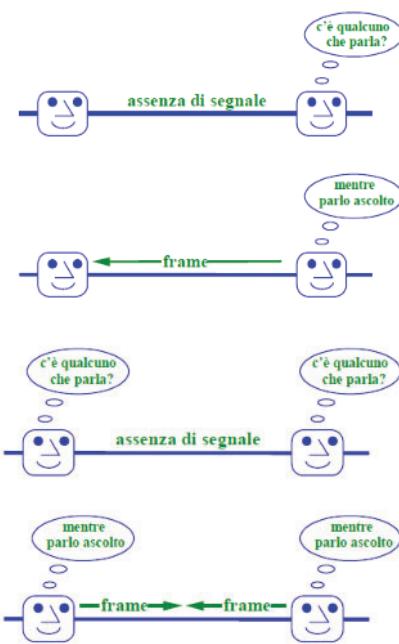
### Il round trip delay

L'algoritmo *csma/cd* è quindi vincolato da un **ritardo**, definito **round trip delay**  $2\tau$ , ovvero il tempo necessario per un bit per propagarsi da un estremo all'altro della rete e "tornare indietro" attraverso una conferma di ricezione.

N.B.: una stazione trasmittente è in grado di capire che il pacchetto che ha trasmesso è

entrato in collisione con un altro pacchetto solo mentre trasmette.

Supponiamo ora il caso basilare di due interlocutori A e B che parlano tramite un canale di trasmissione:

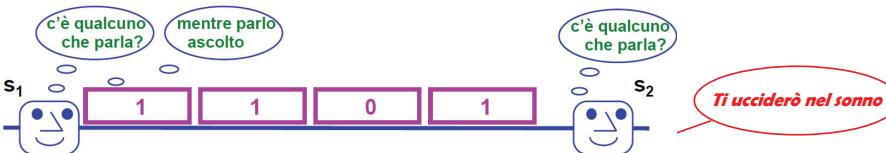


In una trasmissione senza collisione, il mittente rileva l'assenza di segnale sul canale, quindi lo occupa trasmettendo dei frame, "parlando" al destinatario; mentre parla, inoltre, continua ad ascoltare il canale per vedere se questo rimane libero e occupato solo da lui. Ascolta per verificare se c'è una collisione.

In una trasmissione con collisione, supponiamo che vi siano due mittenti agli estremi della rete:

- 1) A inizia a trasmettere sul canale libero;
- 2) B inizia a trasmettere sul canale libero un istante prima di essere raggiunto dal primo bit trasmesso da A;
- 3) Un istante dopo, B rileva la collisione e trasmette la sequenza di **jamming**;
- 4) A rileva la collisione solo quando il primo bit della sequenza di B arriva fino a lui;
- 5) La trasmissione di A, quindi, deve durare almeno quanto il **round trip delay**.

Nel caso in cui, infatti, una rete sia troppo "corta", se B (che vede il canale non occupato) inizia a trasmettere quando il primo bit del pacchetto inviato da A è quasi in arrivo, si verificherà una collisione che non sarà mai rilevata da A.



Nota quindi la velocità di propagazione sul cavo, e stabilita la velocità di trasmissione della rete, la dimensione minima del pacchetto e la lunghezza massima della rete si influenzano reciprocamente e determinano l'esito dell'algoritmo csma/cd.

Ad esempio, partendo dalla dimensione del pacchetto, calcoliamo la lunghezza massima della rete:

metodo di accesso al canale trasmittivo usato in ethernet e nelle prime versioni dello standard IEEE 802.3

### Un po' di storia

Storicamente, lo standard delle reti ethernet fu definito da un consorzio formato negli anni '70 da tre delle più grandi società del tempo, ovvero Digital, Intel e Xerox. Il consorzio DIX stabilì nel 1980 lo standard ethernet 1.0, aggiornato nel 1982 alla versione 2.0, mentre nel 1989, grazie al consorzio IEEE, Ethernet fu ridefinita dall'intero standard IEEE 802.3 / ISO 8802.3. L'obiettivo dello standard è di favorire l'equivalenza tra tutti i nodi, senza prevedere priorità o garanzie sul tempo massimo di risposta

Una collisione si può manifestare anche quando si pensa di aver già occupato il canale trasmittivo; infatti, scartando l'ipotesi di rilevazione istantanea del canale libero (specialmente nelle reti wan), è possibile che, per distanze lunghe, il tempo di rilevazione del carrier porti a una terminazione precoce della trasmissione.

jamming - sequenza di bit che "sporca" la trasmissione, bit di lunghezza fissa e senza significato concepita per invalidare fcs

Io trasmetto in serie, bit dopo bit  
alla velocità della luce  
il primo bit arriva, poi il secondo... verifica  
se il destinatario è giusto controllando  
l'indirizzo MAC e vedendo se corrisponde

- **definizione:** il **round trip delay**  $2\tau$  e' il tempo necessario per un bit per propagarsi da un estremo all'altro della rete e "tornare indietro"
- **osservazione:** una stazione trasmittente è in grado di capire che il pacchetto che ha trasmesso è entrato in collisione con un altro pacchetto **solo mentre trasmette**
- supponiamo che la stazione  $S_1$ , posizionata ad un estremo della rete, inizi a trasmettere
- supponiamo che  $S_2$ , posizionata all'altro estremo della rete, inizi a trasmettere un istante prima di essere raggiunta dal primo bit trasmesso da  $S_1$
- un istante dopo  $S_2$  rileva la collisione e trasmette la sequenza di jamming
- $S_1$  rileva la collisione solo quando il primo bit della sequenza di  $S_2$  arriva fino a lei
- quindi la trasmissione di  $S_1$  deve durare almeno quanto il **round trip delay**

Deve trasmettere 4 bit: 1011  
Manda il primo bit (1) (non arriva subito), manda il secondo bit etc...  
 $S_1$  mentre parla ascolta per vedere se ci sono collisioni.

Supponiamo che  $S_2$  voglia trasmettere quando il primo bit non è ancora arrivato davanti a se.  
Entra in collisione. Ma  $S_1$  non se ne accorgerebbe mai perché ha mandato già tutti i suoi bit e non ascolta più.

Se  $S_1$  trasmette almeno per tutto il tempo per un bit per andare dall'altra parte e tornare, me ne accorgo. Trasmettere dei bit riempitivi.

Almeno nove in questo caso.

Una seconda possibilità sarebbe avvicinare  $S_2$ . Il tempo che ci mette un bit ad arrivare dall'altra parte diminuisce.

- supponiamo che il pacchetto più corto previsto dal protocollo abbia 512 bit, che la banda sia 10 Mbit/s e che la velocità di propagazione del segnale sia 2/3 della velocità della luce nel vuoto ( $\approx 3 \cdot 10^8$  m/s)
- il tempo di immissione sul mezzo trasmissivo del pacchetto più corto è  $512/10^7 = 512 \cdot 10^{-7}$  sec., cioè  $51.2\mu s$
- quindi il roundtrip delay ammisible è  $2\tau = 51.2\mu s$
- in  $512/2 \cdot 10^{-7}$  sec. il segnale percorre  $512/2 \cdot 10^{-7} \cdot 2/3 \cdot 3 \cdot 10^8$  metri, circa 5 km; limite massimo di estensione della rete

Se l'algoritmo funziona correttamente la rete ha una distanza temporale tra i due estremi minore di  $\tau$ , allora una stazione dopo  $2\tau$  di trasmissione sicuramente non rileverà collisioni sul pacchetto che sta inviando; infatti, dopo un dato istante di tempo, il canale trasmissivo sarà completamente occupato dai bit della stazione mittente.

Dimensionando la rete in questo modo, è infatti possibile realizzare il *collision detection*; nel peggior dei casi, la collisione potrà avvenire tra due macchine poste agli estremi della rete. Se in un tempo  $\tau$  il pacchetto riesce ad occupare tutta la rete, vi potrà essere una collisione massima fino a  $2\tau$ , ma verrebbe riconosciuta regolarmente.

### Algoritmo Truncated Binary Exponential Backoff (calcolo del tempo di attesa)

Come abbiamo visto, quando si verificano delle collisioni, la stazione trasmittente attende un tempo **randomico** multiplo di un tempo stabilito prima di riprovare la trasmissione del pacchetto. Questo tempo viene calcolato tramite un algoritmo detto **truncated binary exponential backoff**, implementato come segue:

```
se tentativo=1 (primo tentativo di ritrasmissione del frame)
allora
    max:=2
altrimenti
    se tentativo < limite_di_backoff
        allora max:=max × 2
    aspetta(2τ × random(0,max-1))
```

Come suggerisce il nome, l'algoritmo produce valori di attesa che crescono esponenzialmente all'aumentare del numero di tentativi. Ciò tradotto vuol dire che all'aumentare delle collisioni, le stazioni si **auto-limitano**; la scelta, tuttavia, di attendere un tempo multiplo di  $2\tau$ , ovvero del **round trip delay**, permette di diminuire la probabilità di avere ulteriori collisioni, ma perché?

Vediamo con un esempio la relazione tra  $2\tau$  e l'algoritmo:

#### Analisi di una collisione

Per comprendere al meglio questo legame, studiamo analiticamente una collisione tra due stazioni che utilizzano csma/cd e l'algoritmo di backoff.

- **Situazione:** due stazioni ( $s_1$  e  $s_2$ ) a "distanza temporale"  $d$  cominciano a trasmettere iniziando rispettivamente ai tempi  $t_1$  e  $t_2$  (con  $t_1 \leq t_2$ , ovvero una è in ritardo rispetto all'altra);
  - Supponiamo  $d < \tau$ , dove  $\tau$  è il tempo massimo per andare da una stazione all'altra). Tale vincolo è necessario, altrimenti non si avrebbe *collision detection*;
  - Si verifica una collisione tra il pacchetto di  $s_1$  e quello di  $s_2$ .
- **Nota:** perché la collisione si verifichi deve essere  $t_2 < t_1 + d$ , infatti dopo questo tempo la stazione  $s_2$  non può più trasmettere per *carrier sense* (non può trasmettere se il bus non è libero);
  - Sia  $t_2 = t_1 + \Delta$  (con  $\Delta < d < \tau$ ).
- **Percezione della collisione:** siano  $t_{1p}$  e  $t_{2p}$  i tempi in cui  $s_1$  e  $s_2$  percepiscono la collisione (con  $t_{1p} \geq t_{2p}$ , perché  $s_2$  è più "vicina" alla collisione di  $s_1$ );
  - $t_{1p} = t_2 + d = t_1 + \Delta + d$ ;
  - $t_{2p} = t_1 + \Delta$ .
- **Canale nuovamente libero in prossimità di  $s_1$  e  $s_2$ :** sia  $j$  il tempo di trasmissione della sequenza di *jamming* da ciascuna stazione:
  - l'ultimo bit emesso da  $s_2$  passa davanti a  $s_1$  all'istante  $t_{2p} + j + d = (t_1 + \Delta) + j + d = t_1 + j + 2\Delta$ ;
  - l'ultimo bit emesso da  $s_1$  passa davanti a  $s_2$  all'istante  $t_{1p} + j + d = (t_1 + \Delta + d) + j + d = t_1 + \Delta + j + 2d$ .
- **Istante in cui  $s_1$  e  $s_2$  riprovano a trasmettere:** siamo  $t_{1r}$  e  $t_{2r}$  i tempi in cui  $s_1$  e  $s_2$  riprovano a trasmettere:
  - $t_{1r}$  e  $t_{2r}$  sono legati a due eventi:
    - disponibilità del canale (assenza di segnale) in prossimità di  $s_1$  e  $s_2$ ;
    - necessità di attendere un tempo random determinato dall'algoritmo di backoff;
  - tempo in cui  $s_1$  riprova a trasmettere:  

$$[t_{1r} = \max(t_1 + \Delta + d + j + n2\tau, t_1 + j + 2\Delta)] = t_1 + j + d + \max(\Delta + n2\tau, d);$$

$$(nel max, portiamo fuori i termini uguali, ovvero t_1, d e j)$$
  - tempo in cui  $s_2$  riprova a trasmettere:  

$$[t_{2r} = \max(t_1 + d + j + m2\tau, t_1 + \Delta + j + 2d)] = t_1 + j + d + \max(m2\tau, \Delta + d);$$

$$(nel max, portiamo fuori i termini uguali, ovvero t_1, d e j)$$
- **n** ed **m** sono i numeri sorteggiati dal backoff;
  - il primo argomento del max è il tempo minimo di inizio della ritrasmissione dovuto al **backoff**;
  - secondo è il tempo minimo di inizio della ritrasmissione dovuto all'**attesa di canale libero**.

- normalmente la dimensione minima del pacchetto è fissata
- dalla dimensione minima del pacchetto si può calcolare la estensione massima della rete
- **osservazione:** se il metodo funziona correttamente e la rete ha una "distanza temporale" tra i due estremi (tempo di propagazione del segnale tra i due estremi) minore di  $\tau$  allora una stazione dopo  $2\tau$  di trasmissione è sicura di non avere più collisioni sul pacchetto che sta inviando
  - dopo  $\tau$  il canale trasmissivo contiene ovunque bit della stazione

#### Funzionamento dell'algoritmo

- 1) Si definisce un intervallo di valori in cui si pescherà un numero;
- 2) Dopo la collisione le due macchine estraggono a caso un numero (all'inizio sceglieranno sempre 0 o 1) e aspettano un tempo pari al numero pescato moltiplicato per  $2\tau$ , quindi ricominciano a trasmettere;
- 3) Se una macchina ha pescato 0 e l'altra 1, una comincia a trasmettere subito, l'altra dopo  $2\tau$ . In questo modo non vi sarà collisione, perché dopo  $2\tau$  la prima macchina avrà trasmesso e liberato la rete;
- 4) Se invece entrambe le macchine pescano un numero uguale, si avrà una nuova collisione, perché entrambe trasmettono dopo aver aspettato lo stesso tempo. In tal caso l'algoritmo raddoppia l'intervento in cui pescare il numero;

Supponiamo che questo intervallo valga 4: le macchine potranno pescare 0, 1, 2 o 3, riducendo la probabilità di pescare lo stesso numero dal 50% al 25%. Dopo aver pescato il numero le macchine, prima di ritrasmettere, aspettano un tempo pari al prodotto  $2\tau$  per il numero; ovviamente, se avranno pescato lo stesso numero, l'intervento verrà nuovamente raddoppiato e si pescherà un altro numero.

N.B.: ricordiamo che esiste un numero massimo di tentativi di trasmissione, detto **limite di backoff**. Se ad esempio vale 16, dopo 16 collisioni consecutive si abortisce il processo di trasmissione.

istante nel quale il primo bit trasmesso da una stazione passa davanti all'altra

quando torna libero il canale?  
l'ultimo bit di jamming arriva dopo d

devo aspettare il massimo tra questi due tempi  
 $t_1 + \Delta + d + j + n2\tau$  tempo di backoff = tempo in cui ho percepito la collisione,  
tempo del jamming + n2tau  
n e m sono i numeri sorteggiati dal backoff

- Distanza temporale tra la ritrasmissione di s1 e la ritrasmissione di s2:

- $t_{1r} = t_1 + j + d + \max(\Delta + n2\tau, d);$
- $t_{2r} = t_1 + j + d + \max(m2\tau, \Delta + d);$
- $t_{1r} - t_{2r} = \max(\Delta + n2\tau, d) - \max(m2\tau, \Delta + d).$

| Caso                 | $t_{1r} - t_{2r} =$                   | Si ha una nuova collisione? |
|----------------------|---------------------------------------|-----------------------------|
| $m=n=0$              | $= d - (\Delta + d) = \Delta$         | Sì                          |
| $m=n>0$              | $= \Delta + n2\tau - n2\tau = \Delta$ | Sì                          |
| $n=0, m>0$           | $= d - m2\tau > \tau$                 | No                          |
| $n>0, m=0$           | $= n2\tau - d > \tau$                 | No                          |
| $m \neq n, n>0, m>0$ | $= \Delta + (n-m)2\tau > \tau$        | No                          |

- Conclusioni:** due stazioni collidono nuovamente solo se estraggono lo stesso numero casuale. Se avessimo scelto di moltiplicare per multipli di  $\tau$  invece che di  $2\tau$ , si sarebbe potuta verificare una collisione anche se i due numeri scelti fossero stati diversi; ciò spiega perché utilizziamo multipli di  $2\tau$  per l'algoritmo di backoff.

multipli di  $3\tau$  ??? cosa sarebbe successo?

**Si ha una nuova collisione perché la distanza temporale tra le ritrasmissioni di s1 e s2, essendo  $\Delta$ , sarà minore di  $2\tau$ , ovvero minore del round trip delay.**

**Non si hanno nuove collisioni in tutti gli altri casi in cui la distanza temporale è almeno pari al round trip delay!**

## 5 - IEEE 802.3 - Ethernet

11 ottobre 2021

### Lo standard IEEE 802.3/iso8802.3

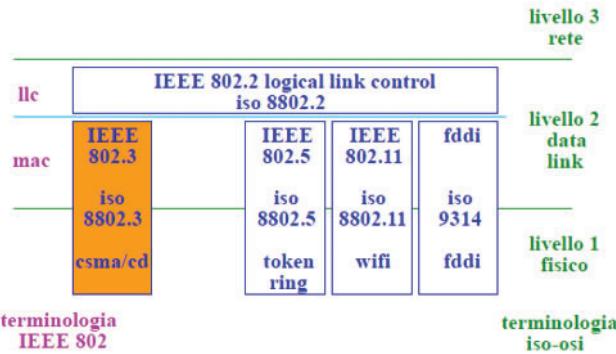
Come abbiamo accennato, lo standard IEEE 802.3 è uno standard per le reti locali (*lan*) nato nel 1985 come ridefinizione della precedente tecnologia *Ethernet*, ed è il più popolare tra i protocolli della famiglia IEEE 802. Esso è definito a livello fisico come standard ISO 8802.3 ed è lo standard che implementa il rilevamento delle collisioni nel progetto IEEE 802.

802.3 offre una grande flessibilità nella scelta della topologia di una lan. Originariamente si utilizzava solo la **topologia a bus** (ereditata da Ethernet), nella quale il collegamento logico e fisico coincidevano e tutte le stazioni condividevano il mezzo trasmittivo; successivamente, grazie all'impiego di dispositivi di rete quali gli **hub**, come vedremo più avanti, è divenuto possibile realizzare una **topologia fisica a stella** pur utilizzando una topologia logica a bus.

Come già detto, la distanza massima è limitata e in questo protocollo è vincolata a 4 chilometri, mentre il numero massimo di stazioni consentite in una singola rete lan è di 1024.

*IEEE 802.3 rappresenta l'evoluzione dello standard Ethernet 2.0 per le reti lan ed è l'implementazione della disciplina csmacd nello standard IEEE 802.*

Ritornando quindi al disegno generale di IEEE 802, approfondiremo il **rettangolo arancione** e quello che riguarda il *mac 802.3* e il mezzo trasmittivo. A destra della figura possiamo vedere come gli standard del livello mac e del livello llc corrispondano al livello Data Link nella terminologia ISO-OSI.



lo standard IEEE802.3/iso8802.3  
nella sua versione più semplice

- mezzi trasmittivi
  - coaxiali (in disuso)
  - fibre ottiche
  - cavi in rame twisted pair
- inoltre
  - circa 4km di distanza tra le stazioni più distanti
  - massimo 1024 stazioni
- topologie
  - bus e stella
- velocità
  - 10 Mb/s (ormai quasi ovunque sostituita da velocità superiori)

### Topologie di rete

Come vedremo più avanti, è possibile inoltre realizzare, grazie agli **hub**, reti più complesse fisicamente lasciando che la topologia logica rimanga quella a bus. I pacchetti, infatti, verranno comunque trasmessi su tutta la rete e questo porta a due limitazioni:

- Non possono esserci più di due **hub** nella stessa rete (per evitare di superare le temporizzazioni previste da *csmacd*);
- Non è possibile realizzare collegamenti ridondanti, in quanto si formerebbe un anello in cui i pacchetti continuerebbero a venire ritrasmessi saturando la rete e rendendola inutilizzabile;

Vedremo come, poi, sarà possibile realizzare topologie di rete con collegamenti ridondanti grazie all'impiego di **bridge** e **switch** dotati dello **Spanning Tree Protocol**, il quale si occupa di mettere fuori servizio tali collegamenti e riattivarli in caso di caduta dei collegamenti attivi, per ripristinare la connettività.

[spoiler alert]

### Il pacchetto a livello Data Link - Il frame Ethernet

Come tutti i protocolli della famiglia 802, anche 802.3 definisce il suo particolare sottolivello mac. Per prima cosa analizziamo come sia fatta una *mac-pdu* 802.3 (che chiameremo anche **frame Ethernet**, per analogia con lo standard Ethernet), che specifica dei campi ulteriori rispetto a quelli generici 802:

| Campo | PRE | SFD | DA  | SA  | L/T | Dati     | PAD  | FCS |
|-------|-----|-----|-----|-----|-----|----------|------|-----|
| Byte  | 7   | 1   | 2-6 | 2-6 | 2   | 0 - 1500 | 0-46 | 4   |

DSAP SSAP LLC PDU FCS  
Una mac-pdu generica IEEE 802.

- **Preamble** (7 Byte / 56 bit): si tratta di una sequenza randomica di 1 e 0 che consente al ricevente di impostare e di sincronizzare la comunicazione, "svegliando" l'adattatore mettendolo in guardia dell'arrivo del frame. La presenza del preamble è fondamentale, in quanto sulle reti 802.3 non vi è una frequenza portante del segnale. I 56 bit iniziali dei frame Ethernet vengono quindi scartati e non passano allo strato superiore in fase di trasmissione;

- **SFD - Starting Frame Delimiter** (1 Byte / 8 bit): si tratta di un singolo byte la cui sequenza è 10101011, in esadecimale AB, che dichiara che dal prossimo byte avrà inizio il vero e proprio frame. È importante notare che nel frame Ethernet non è presente un delimitatore in coda al pacchetto;

- **DA - Destination Address** (6 Byte / 48 bit): contiene l'**indirizzo mac** del destinatario, spesso rappresentato nella forma **aa:bb:cc:dd:ee:ff**; i primi due bit hanno significato particolare:
  - Se il primo bit vale 0, la destinazione è una singola unità, altrimenti è un gruppo;
  - Se il secondo bit vale 0, l'indirizzo ha **valore globale**, altrimenti ha valore locale;

- **SA - Source Address** (6 Byte / 48 bit): contiene l'**indirizzo sorgente** e ha la stessa struttura del DA, ma rappresenta sempre la singola unità e per tale motivo il primo bit è sempre impostato a zero;

Commando i bit di ogni campo è possibile risalire alla dimensione minima di un frame Ethernet: 512 bit (64 Byte), che sommati al preamble e allo Start Frame Delimiter arrivano a contare di 576 bit.

La dimensione massima del frame (senza contare il preamble e l'sfd) è invece fissata a 1518 Byte: un pacchetto di dimensione maggiore impiegherà un tempo di trasmissione critico e la trasmissione fallirà.

Queste due costanti sono state determinate attraverso calcoli probabilistici. Per esempio, la dimensione minima è stata adottata per evitare di dover costruire schede di rete con memorie troppo grandi (che di solito possono memorizzare contemporaneamente massimo 16 pacchetti).

Dallo studio dei campi del frame Ethernet, possiamo ricapitolare le funzioni svolte dal *mac 802.3*:

- Trasmissione dei pacchetti: il *mac* riceve un pacchetto dal livello llc, lo incapsula nel pacchetto di livello mac e lo trasforma in una stringa di bit che viene consegnata al livello phy per la trasmissione sul mezzo fisico;
- Ricezione dei pacchetti: il *mac* riceve una stringa di bit dal livello phy e lo interpreta come pacchetto di livello mac; se il pacchetto è indirizzato ad altri o contiene errori viene scartato, altrimenti la busta viene "scartata" e viene consegnato il pacchetto llc al livello superiore;
- Schedulazione ritrasmissioni e trasmissione in modalità differita (se il canale è occupato);
- Generazione del campo *fcs*: come visto, il *mac* può generare il codice a ridondanza ciclica (CRC) per il controllo degli errori;

- **L/T - Length/Type** (2 Byte / 16 bit): detto anche **EtherType**, questo campo può specificare la lunghezza o il tipo del pacchetto. Come vedremo avanti, nella rete possono coesistere pacchetti di protocolli diversi, tra cui vecchi pacchetti Ethernet 2.0. Per consentire a questi di usare la versione originale del frame, il valore **EtherType** assume un valore maggiore o uguale di 1536 (0x600); L'EtherType permette inoltre di identificare altri particolari pacchetti contenenti informazioni di diagnosi della rete; in questo caso, assume valore 1518. Questo campo, quindi, a seconda del valore assunto, può indicare sia la lunghezza (per pacchetti standard) che il tipo (per tutti gli altri pacchetti) dei byte di dati forniti dal livello superiore che saranno trasmessi in questo frame.

- **Data (payload)**: contiene le **llc-pdu** che sono i dati veri e propri e che, nel caso di frame standard, vengono trasmessi "a bordo" di questi. La dimensione effettiva dei dati in un pacchetto non può superare i 1500 byte, ma, se sono meno di 46 byte, occorre aggiungere dei byte supplementari di riempimento per arrivare almeno a 46. Questo serve a garantire in ogni caso una lunghezza minima totale del frame di almeno 64 byte, essenziale per evitare che la trasmissione di frame troppo corti provochi la mancata individuazione delle collisioni nei casi peggiori;

- **PAD** (46 - 1500 Byte): è un campo di riempimento utilizzato per garantire la lunghezza minima di 64 byte (512 bit); esso varia da 46 a 1500 byte, poiché 18 byte sono sempre presenti nella trama (header);

- **FCS - Frame Check Sequence** (4 Byte - 32 bit): contiene il valore di controllo calcolato dal mittente secondo l'algoritmo **CRC (Cyclic Redundancy Check)**; il ricevente farà lo stesso non appena ricevuto l'intero frame, e potrà così confrontare il valore del campo *fcs* con quello da lui calcolato.

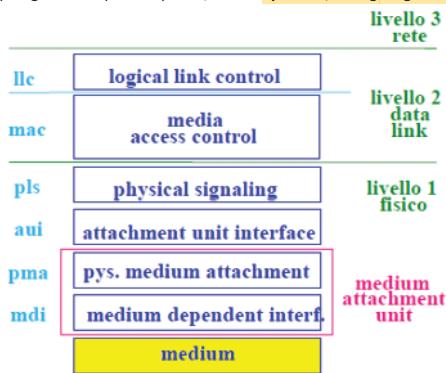
Ulteriori caratteristiche del frame Ethernet sono le seguenti:

- **parametri:**
  - inter frame spacing o **inter-packet gap** (ipg): tempo per trasmettere 96 bit (96 bit time); a 10 Mbit/sec è uguale a 9,6µs
  - massimo numero ritrasmissioni: 16
  - lunghezza del jam: 32 bit (96 per i repeater)
  - lung. min-max pacchetto senza preamble e sfd: 64-1518 byte
- **dominio di collisione:**
  - zona della lan dove due pacchetti possono collidere

- **Controllo del campo fcs:** il mac verifica che l'fcs ricevuto sia uguale a quello calcolato localmente. I pacchetti con errori vengono scartati senza richiesta di ritrasmissione;
- **Spaziatura di pacchetti:** il mac garantisce un tempo di pausa minimo tra l'invio consecutivo due pacchetti: tale ritardo è detto *inter frame spacing* o *inter-packet gap (ipg)*, e corrisponde a 9,6  $\mu$ s per trasmettere 96 bit con la specifica a 10 Mbit al secondo;
- **Verifica di lunghezza minima del pacchetto:** il mac verifica che la lunghezza minima del pacchetto ricevuto dal livello fisico non sia inferiore a 64 byte;
- **Generazione e rimozione del preambolo:** come abbiamo visto, il preambolo viene generato e rimosso dal mac per sincronizzare la trasmissione tra i livelli fisici della rete.

### Il livello fisico 802.3

Il livello fisico (*Phy*) è descritto come segue in figura da tre interfacce fisiche, che collegano il mac con il mezzo trasmisivo (*medium*). L'approfondimento dei **protocolli** di questo livello non è obiettivo di questo corso, mentre approfondiremo le **infrastrutture** fisiche che permettono di realizzare le topologie di rete più complessa, come i **ripetitori**, i **bridge** e gli **switch**.



A puro titolo di studio, vediamo comunque a grandi linee di cosa si tratta:

- **PLS (physical layer signaling):** collega il mac e l'*au*; il **mac** fornisce bit al *pls* con codifica *nrz*, e questo codifica i bit con codice *manchester*; Questa codifica risulta essere facile a comprendersi in fase di sincronizzazione, ma a una frequenza di 5 Mhz *nrz* corrisponde una frequenza *manchester* doppia. Inoltre il *pls* fornisce informazioni al mac su *carrier sense*, collisioni, dati ricevuti e malfunzionamenti;
- **AUI (attachment unit interface):** collega il *pls* e il *mau*; è un livello che codifica in *manchester* il segnale e lo trasmette all'unità di collegamento. È generalmente costituita da tre fili: *data out*, *data in* e *control in*; in particolare quest'ultimo trasferisce l'informazione di collisione;
- **MAU (medium attachment unit):** è un *transceiver*, un trasmettitore/ricevitore che varia a seconda del mezzo di trasmissione.

### I Ripetitori o hub

Il **ripetitore** è un apparecchio fisico che, oltre ad amplificare il segnale della rete, realizza un concentratore, ovvero un dispositivo di rete che funge da nodo di smistamento dati di una rete organizzata con una topologia **logica a bus** e di topologia **fisica a stella**. Nel caso delle reti 802.3 ed **Ethernet**, un **hub** è un dispositivo che inoltra i dati in arrivo da una qualsiasi delle sue porte su tutte le altre, cioè in maniera diffusiva (**broadcasting**), e per questa ragione può essere definito anche come ripetitore multiporta.

L'uso di hub non influisce sulla topologia logica della rete, che funzionerà come un bus.

I singoli bit che viaggiano attraverso un ripetitore sono inoltrati immediatamente sulle altre porte, e analogamente vengono trasferite tutte le collisioni ai terminali; alcuni ripetitori, tuttavia, possono escludere una porta quando sul segmento corrispondente si verificano troppe collisioni. Inoltre, quando un *frame* transita nel ripetitore (che viene visto dal mittente come il sistema destinatario), questo si occupa di rigenerarne il preambolo, ritemporizzando tutti i bit, e di trasferirlo al destinatario reale connesso ad un'altra delle porte del ripetitore.

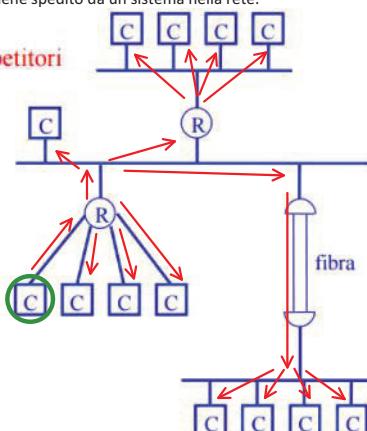
Poiché vengono trasmessi direttamente i bit del pacchetto e non il pacchetto intero, non possiamo definire l'hub come una macchina **store and forward**.

(ovvero un dispositivo che **memorizza** il pacchetto e lo **inoltra** attraverso una delle sue porte)

Raffigurando per convenzione gli **hub** con un cerchio, mentre le porte dei terminali con un quadrato, vediamo cosa succede quando un bit viene spedito da un sistema nella rete.

#### esempio di rete con ripetitori

Supponendo che il bit parta dalla macchina cerchiata di verde, il ripetitore ritrasmette il bit a tutte le sue porte, e così fanno tutti i ripetitori nella rete. Il bit raggiunge quindi **tutti i terminali connessi alla rete**, ovvero **occupa tutta la rete**.



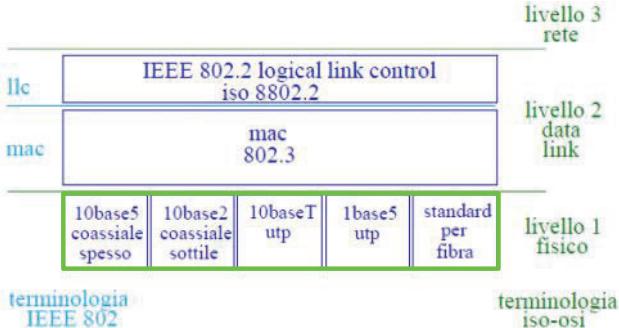
Questa visione della rete a "bus che viene occupato" è ciò che realmente è a livello fisico nella nostra pila ISO-OSI. Poiché il ripetitore non esiste nei livelli più alti della rete, possiamo ignorare la topologia fisica della rete e considerare tutti i computer come connessi ad un unico singolo bus **condiviso** (proprietà che, tra l'altro, è una delle ipotesi della rete lan).



### I mezzi trasmisivi 802.3

Come abbiamo detto in principio, esistono diverse versioni di connettori e di cavi di infrastruttura su cui è realizzato il livello fisico 802.3 e di cui si avvalgono le tecnologie mac e *csm/a/cd*. Dividendo il livello fisico della scatola arancione vista in introduzione, analizziamo le diverse scatole verdi di

livello più basso che specificano diversi standard per la trasmissione dei segnali di rete.



A livello fisico, 802.3 prevede esclusivamente **trasmissioni via cavo in banda base**, a velocità di 10, 100 e 1000 Mbit/s, su cavi coassiali, dppini intrecciati (schermati e non) e fibre ottiche. Queste caratteristiche sono riassunte negli acronimi usati per le varie implementazioni del livello fisico, tutti del tipo **NBaseA**, essendo **N** la velocità di trasmissione, **Base** indica che l'implementazione opera in banda base, ed **A** è una sigla legata al tipo di cavo utilizzato e ad altre caratteristiche salienti. Attualmente vengono installate soprattutto le varianti a 100 Mbit/s e superiori, come **100Base-T**, al momento certamente tra le più diffuse, mentre altre implementazioni come **10Base5 (thick Ethernet)**, **10Base2 (thin Ethernet)**, **10Base-T** non vengono più installate.

- La tecnologia **10Base-T**, detta anche **Unshielded Twisted Pair (utp)**, è caratterizzata dalla velocità di trasmissione di 10Mbps in banda base su due **doppini** intrecciati e non schermati, di categoria 4 o 5 derivati dai cavi telefonici; l'utilizzo di doppini è necessario perché vengono utilizzate **due coppie separate per la trasmissione e la ricezione** del segnale;
  - Questa tecnologia prevede solo **collegamenti fisici a segmento punto-punto**, mentre (come accennato) è possibile ottenere topologie logiche a stella/albero utilizzando sistemi intermedi come gli **hub**, e, come vedremo più avanti, **bridge e switch**;
  - Come i cavi stessi, i **connettori** sono di **derivazione telefonica**, di tipo **RJ-45**, molto pratici ma meno affidabili dei connettori BNC; essendo un cablaggio riciclabile a volte dagli impianti telefonici, i singoli tratti non possono superare i 100 metri. Spesso, inoltre, si utilizzano dei connettori di crossover, che solitamente sono presenti a bordo dei ripetitori;
  - Quando una connessione è attiva, sui cavi viaggia un segnale di **idle** costante; inoltre, tramite l'invio di segnali chiamati **link test pulse**, è possibile attuare una procedura detta di **autonegoziazione**, in cui i dispositivi connessi tramite cavi 10Base-T sincronizzano i parametri della trasmissione quali velocità, modalità **duplex** e controllo di flusso.
- La tecnologia **10Base-F** è lo standard generico per le reti Ethernet tramite **cavi in fibra ottica**, sempre a banda base con velocità di 10Mbps. Vi sono più versioni di questo standard, tra cui:
  - 10Base-FP (passive)**, che definisce un accoppiatore di segnale non alimentato (ripartitore di segnali ottici), il quale funge a tutti gli effetti da **ripetitore passivo** del segnale che **implementa una topologia fisica a stella**. Questo permette di collegare fino a 33 dispositivi, con ciascun segmento lungo fino a 500 metri, quindi collegando due dispositivi a cavallo della **stella ottica** fino a 1000 metri di distanza tra loro;
  - 10Base-FB (backbone)**, che definisce un segmento di rete utilizzato per collegare tra loro gli **hub** di rete, solitamente impiegato sulle **dorsali di rete** tra due ripetitori. Questa tecnologia è naturalmente impiegata a lunghe distanze e permette segmenti lunghi fino a 2000 metri; inoltre, la trasmissione è **sincrona** grazie a un segnale di **idle** sincronizzato, permettendo di ridurre i ritardi normalmente associati ai ripetitori Ethernet.



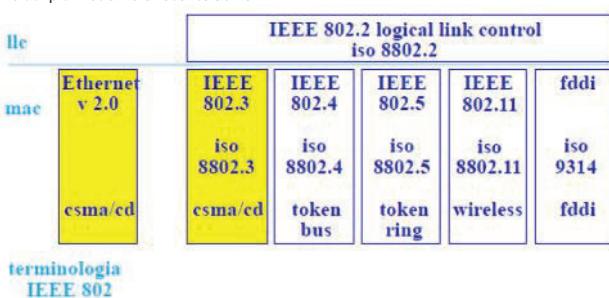
I connettori RJ-45 dei cavi 10Base-T



I connettori in fibra ottica dei cavi 10Base-F

### Differenze tra Ethernet 2.0 e IEEE 802.2/3

Ricordiamo che storicamente lo standard per le reti lan era **Ethernet 2.0**. Nelle reti locali oggi convivono comunque pacchetti IEEE 802.3 e pacchetti Ethernet 2.0, ma cosa differenzia questo protocollo dal più moderno e recente 802.3?



Innanzitutto, Ethernet 2.0 non era uno standard **ISO**, e non definiva lo strato llc; per questo motivo, il pacchetto di livello di Rete (livello 3) entrava direttamente nel pacchetto di livello mac. Inoltre, le funzioni di **multiplexing** per i protocolli di livello 3 svolti da llc erano svolte da un campo presente nel pacchetto mac, chiamato **type**, sostituito con il campo **length** in 802.2/3. Questo campo è stato poi modificato in 802.2/3 nel campo dinamico **EtherType**, che, come abbiamo visto, permette tutt'oggi la coesistenza di pacchetti di standard più vecchi nelle reti lan.

### Ethernet 2.0

|      |     |    |    |      |      |  |  |     |
|------|-----|----|----|------|------|--|--|-----|
| Pre. | SFD | DA | SA | Type | Data |  |  | FCS |
|------|-----|----|----|------|------|--|--|-----|

### IEEE 802.2 e 802.3

|      |     |    |    |     |      |      |      |      |     |
|------|-----|----|----|-----|------|------|------|------|-----|
| Pre. | SFD | DA | SA | Len | DSAP | SSAP | Ctrl | Data | FCS |
|------|-----|----|----|-----|------|------|------|------|-----|

### SNAP (SubNet Access Point)

|      |     |    |    |     |      |      |      |           |      |     |
|------|-----|----|----|-----|------|------|------|-----------|------|-----|
| Pre. | SFD | DA | SA | Len | DSAP | SSAP | Ctrl | SNAP Hdr. | Data | FCS |
|------|-----|----|----|-----|------|------|------|-----------|------|-----|

Ricapitolando, l'interpretazione e il riconoscimento, da parte dei mac dei vari dispositivi, di pacchetti non-802.3 è possibile grazie ad un expediente realizzato per il campo che ne indica la lunghezza (come abbiamo visto nel caso dei frame Ethernet): questo campo, a seconda di come viene codificato dal mac può indicare la lunghezza per pacchetti standard o il tipo per pacchetti non standard.

Per maggiori informazioni fare riferimento al campo **EtherType**.

<https://en.wikipedia.org/wiki/EtherType>

[https://en.wikipedia.org/wiki/Ethernet\\_frame](https://en.wikipedia.org/wiki/Ethernet_frame)

## 6 - Bridge e Switch

14/18 ottobre 2021

### I bridge nelle reti locali

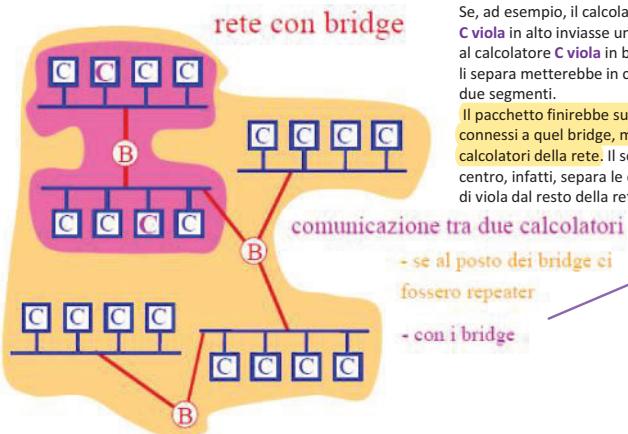
Continuiamo l'approfondimento sulle infrastrutture fisiche della rete e consideriamo ora tutte quelle macchine che consentono lo **Store and Forward** dei pacchetti. Come abbiamo visto, questa tecnica consente nel memorizzare i pacchetti per poi reinserirli nella rete, ed è fondamentale per superare i limiti fisici delle reti lan permettendo di connettere più lan tra loro. tutti i limiti, infatti, sulle distanze massime che possono essere ricoperte dalle lan, sul numero massimo di sistemi interconnessi e sul carico massimo sopportabile, possono essere superati per mezzo di macchine come i **bridge**. Un bridge (letteralmente *ponte*) è un dispositivo di rete che si colloca al livello Data Link nella pila ISO/OSI, e che traduce da un mezzo fisico ad un altro all'interno di una stessa lan; esso è in grado di riconoscere i **frame** dei dati, a partire dai segnali elettrici che capta dal mezzo trasmissivo, e di individuare all'interno di tali pacchetti l'**indirizzo del nodo mittente** e quello del **destinatario**, e in base a questi operare un **indirizzamento dei pacchetti** tra più segmenti di rete ad esso connessi.



Un bridge è un dispositivo fondamentale che permette di connettere tra loro diverse lan, tenendo separati tra loro i traffici locali e filtrando solo i pacchetti diretti da una lan all'altra, ritrasmettendoli con modalità **store and forward**.

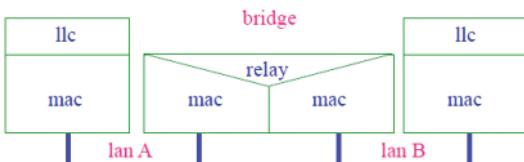
### Differenze tra bridge e hub

Tipicamente, un bridge è munito di porte con cui è collegato a diversi segmenti della rete sui quali indirizza i pacchetti. Il funzionamento è analogo a quello degli **hub**, che permettono di mettere in comunicazione più porte tra loro, tuttavia un **bridge permette effettivamente di separare tra loro diversi segmenti di rete**, nascondendo ai segmenti di rete ai capi opposti di esso il traffico di pacchetti che avviene dall'altra parte.



### I bridge e IEEE 802

Si trovano al livello 2 della pila ISO-OSI, sottolivello MAC dello standard IEEE 802. Un bridge è quindi un dispositivo che **inoltra** i pacchetti da una lan all'altra. A livello del Data Link, nel caso di IEEE 802, possiamo dire quindi che esso realizza un'interconnessione tra due o più sottolivelli mac del livello Data Link OSI. Gli algoritmi di **intradramento** utilizzati dai bridge sono semplici e locali, e differiscono quindi dall'intradramento realizzato dal livello 3, in quanto qui si tratta di un **routing isolato** e limitato alle sole lan connesse al bridge.



I bridge in IEEE 802 devono essere conformi allo standard 802.1D, hanno **tabelle di intradramento a bordo** e risultano invisibili ai dispositivi a cui sono connessi; per questo motivo, i bridge si dicono **transparenti**. Le macchine sulla rete ne ignorano la presenza.

Un bridge trasparente opera in modo **attivo**, accettando ogni pacchetto trasmesso da tutte le lan a cui è connesso. Quando però un bridge riceve un **frame** deve decidere a quale dominio inoltrarlo, oppure se scartarlo. Per inoltrare i **frame** verso i domini giusti, il bridge mantiene una tabella (detta **forwarding table** o anche **filtering database**), di indirizzi mac per ciascuna porta, e, in base al suo contenuto, è in grado di capire verso quale porta (e quindi quale dominio) inoltrare il **frame**. Questa tabella è solitamente generata automaticamente in un processo di **learning** automatico, e si riempie man mano che sul bridge transitano pacchetti. Analizziamo dei casi di affinamenti progressivi:

### i limiti su:

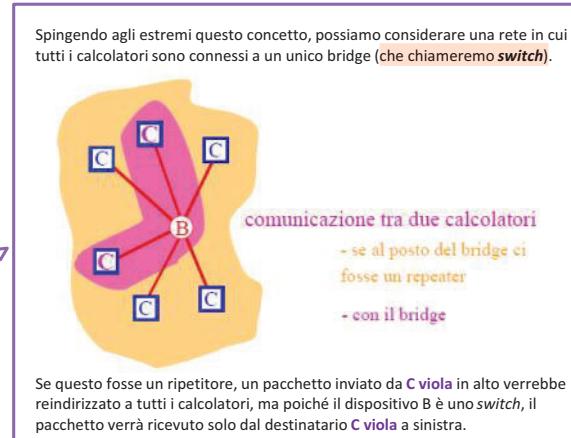
- distanze massime che possono essere coperte dalle LAN finora studiate
- numero massimo di sistemi interconnessi
- traffico massimo sopportabile

sono superati dai **bridge** (in ciò che segue **bridge** e **switch** sono sinonimi)

Possiamo notare da subito una **differenza** tra un **bridge** e un **hub**: il secondo, infatti, non opera a livello Data Link.

In altri termini, l'hub inoltra i singoli bit a tutta la rete, senza preoccuparsi di riconoscere dei pacchetti in questi, mentre il bridge è più "intelligente" e, potendo riconoscere i pacchetti, è in grado di inoltrare il pacchetto solo alla lan a cui è indirizzato.

- un bridge ritrasmette solo i pacchetti che devono effettivamente transitare da una parte della LAN a un'altra parte della LAN: **funzione di filtering**
  - i traffici «locali» sono tenuti separati
- la ritrasmissione avviene con modalità **store & forward** (memorizza e ritrasmetti)
  - ogni pacchetto viene prima ricevuto completamente dal bridge e poi ritrasmesso



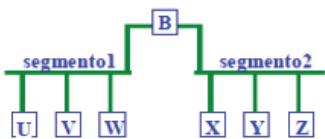
Differenza tra algoritmi di instradamento dei bridge e instradamento del livello 3: quelli dei bridge sono locali.

N.B.: Possono essere connesse allo stesso bridge reti lan con lo stesso mac o con mac differenti: nel caso di mac differenti occorrerà effettuare una traduzione di formato del pacchetto, compreso il ricalcolo del fcs. Se l'interconnessione è inoltre con lan non conformi allo standard IEEE 802 (quindi con livelli llc diversi tra loro), allora occorrerà preoccuparsi anche di tradurre il pacchetto di questo livello!

Inoltre un problema che può presentarsi è quello relativo alle differenze tra le varie lan dei pacchetti di dimensione massima.

relay entity prende un pacchetto che arriva da un certo mac e lo fa transitare su un'altra porta.

Tabella di learning o filtering database che viene costruita con un processo di learning.



| evento         | lista segmento1 | lista segmento2 |
|----------------|-----------------|-----------------|
| bootstrap di B |                 |                 |
| U->V           | U               |                 |
| V->U           | U,V             |                 |
| Z broadcast    | U,V             | Z               |
| Y->V           | U,V             | Z,Y             |
| Y->X           | U,V             | Z,Y             |
| X->W           | U,V             | Z,Y,X           |
| W->Z           | U,V,W           | Z,Y,X           |

Inizialmente, l'host U vuole trasmettere all'host V; il bridge riceve il frame di U e aggiorna la posizione di U collocandolo nella lista della porta 1; quando V risponde a U, il bridge verrà a conoscerne la posizione e anche V verrà collocato nella lista della porta 1. Successivamente l'host Z invia un messaggio di broadcast; il bridge aggiornerà anche la sua posizione e così per ogni nuovo host che trasmette; si avrà alla fine una tabella completa.

Ma nel caso in cui U voglia trasmettere il primo pacchetto a Z, come si comporta il bridge? Non sapendo dove sia Z, questo invierà il pacchetto ricevuto dalla lan 1 a tutte le altre lan connesse tranne quella di provenienza del messaggio; il bridge memorizzerà la posizione di U ma conoscerà quella di Z solo quando questi comincerà a trasmettere.

Dall'esempio visto possiamo riepilogare che:

- Il primo pacchetto che transita sul bridge dalla sua accensione viene inoltrato su tutte le linee (questa procedura è chiamata **flooding**);
- Il bridge aggiorna la **forwarding table** quando un nuovo utente (host) inizia a trasmettere;
- Se la lan di destinazione è diversa da quella di provenienza, il pacchetto viene inoltrato secondo la tabella;
- Se la lan di destinazione e di provenienza sono le stesse, il bridge **scarta il frame** (tiene il traffico locale separato);
- Un bridge vede (considera) due macchine allo stesso modo anche se tra di esse c'è un altro bridge in mezzo;

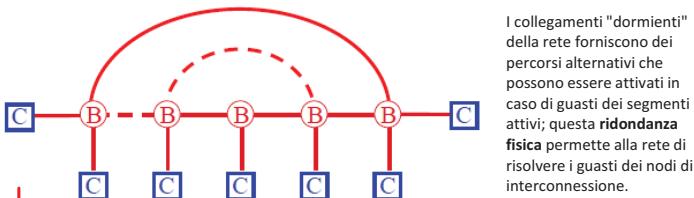
In pratica, i bridge sono sempre più dispositivi plug and play; questo, in coppia col fatto che è possibile **tralasciare i ritardi dovuti alla ritrasmissione**, è il motivo per cui si parla di **bridge trasparenti**.

### Lo Spanning Tree Protocol - STP (l'albero ricoprente della rete)

A questo punto dello studio occorre fare un'osservazione: il "punto debole" in una lan è proprio il nodo di interconnessione, e quindi il bridge; nel caso si guastasse, infatti, la lan verrebbe scollegata dal resto della rete, cioè verrebbero scollegati i singoli segmenti di rete collegati al bridge.

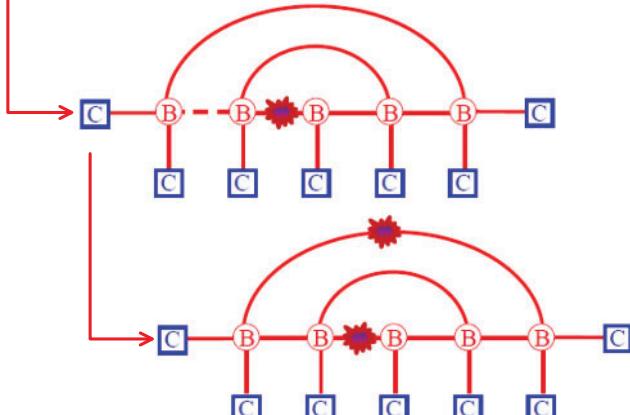
Una delle soluzioni universali al problema dei punti nevralgici è quello di creare **più connessioni** fisiche tra gli host, realizzando dei cicli; facendo ciò, però, ci si trova a dover fare i conti con un altro problema: i pacchetti, in una rete del genere, rischiano di seguire dei percorsi ciclici e di moltiplicarsi, e di fatto renderla inutilizzabile; La topologia fisica cosiddetta a grafo è più affidabile, in termini di sicurezza e resistenza ai guasti, ma necessita di meccanismi che ne permettano il corretto funzionamento.

I bridge moderni possono risolvere il problema creando dinamicamente di uno **spanning tree**, un **albero ricoprente**, ovvero un sottogruppo della rete privo di anelli sul quale **le porte ridondanti, cioè che creano dei cicli, vengono tenute spente** finché non si rende necessario utilizzarle se uno dei segmenti dell'albero viene compromesso.



I collegamenti "dormienti" della rete forniscono dei percorsi alternativi che possono essere attivati in caso di guasti dei segmenti attivi; questa **ridondanza fisica** permette alla rete di risolvere i guasti dei nodi di interconnessione.

In caso di guasti, lo **spanning tree** della rete viene ricalcolato automaticamente dai **bridge**; vengono attivate le porte tenute finora "dormienti".

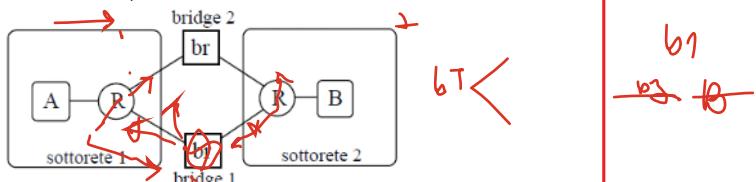


Grazie allo **Spanning Tree Protocol** è possibile astrarre, da una rete con topologia fisica a grafo, ad una topologia logica ad albero, su cui poi verrà costruita la **forwarding table**.

Lo studio del funzionamento automatico degli algoritmi che generano gli **spanning tree** non è obiettivo di questo corso, in quanto argomento avanzato.

#### Bridge in una rete con cicli

Abbiamo detto che l'algoritmo di learning funziona solamente se la rete ha una topologia logica ad albero. Vediamo perché considerando una rete locale IEEE 802.3:



La gestione dinamica della forwarding table può essere resa ancora più sofisticata ed efficiente prevedendo la **cancellazione** da parte del bridge stesso di un indirizzo mac dopo un certo periodo di tempo in cui questo non viene usato (indirizzi obsoleti), evitando così l'aggiornamento manuale e problemi di scalabilità all'aumentare del numero di host della rete.

N.B.: il processo di **learning** di uno **switch** funziona solo se la topologia logica della rete è ad albero.

Detta in altri termini, nella rete non devono esistere **cicli logici**.

Inoltre, in una rete in cui sia presente un **ripetitore (hub)**, non è possibile costruire uno **spanning tree** e non esiste standard che definisca un protocollo a riguardo.

#### spanning tree

- il processo di learning funziona solo se la topologia della LAN è ad albero
- la topologia è normalmente a grafo (contiene uno o più cicli), per motivi di affidabilità
  - viene **dinamicamente calcolato** uno **spanning tree** dei bridge e delle reti
  - solo le porte dei bridge che sono sullo **spanning tree** sono attive

Le R indicano dei ripetitori e A e B sono due calcolatori o host, e le connessioni sono realizzate tutte su cavi utp (10Base-T).

Supponiamo che i bridge siano transparente e conformi allo standard, ma che non siano in grado di calcolare uno spanning tree della rete. In altri termini, supponiamo che tutte le porte dei due bridge siano pienamente attive e che non siano presenti collegamenti dormienti. I bridge sono stati appena accesi e A invia un pacchetto a B, che indichiamo con (A->B).

1. Quanti pacchetti circolano nella rete dopo l'invio del pacchetto e che percorso seguono?

Il pacchetto (A->B) viene ricevuto da entrambi i bridge, i quali, non trovando B nel proprio filtering database, intendono spedire (A->B) sulla sottorete 2. Indichiamo con (A->B)<sup>1</sup> e (A->B)<sup>2</sup> le due copie del pacchetto, e supponiamo che nella contesa del dominio di collisione della sottorete 2 il bridge 1 riesca a trasmettere prima del bridge 2. Il pacchetto (A->B)<sup>1</sup> viene quindi ricevuto da B e dal bridge 2; quindi questo può accedere alla sottorete 2 e inviare il pacchetto (A->B)<sup>2</sup>.

• I pacchetti (A->B)<sup>1</sup> e (A->B)<sup>2</sup> ciclano continuamente nella rete. In particolare, (A->B)<sup>1</sup> segue il percorso antiorario, mentre (A->B)<sup>2</sup> segue il percorso orario.

2. Cosa succede alla forwarding table di bridge 1 e di bridge 2? Ricordiamo che questa viene aggiornata subito dopo l'inoltro di un pacchetto.

Ogni volta che (A->B)<sup>1</sup> transita per bridge 2 il filtering database di bridge 2 annota la presenza di A su sottorete 2.

Ogni volta che (A->B)<sup>1</sup> transita per bridge 1 il filtering database di bridge 1 annota la presenza di A su sottorete 1.

Ogni volta che (A->B)<sup>2</sup> transita per bridge 2 il filtering database di bridge 2 annota la presenza di A su sottorete 1.

Ogni volta che (A->B)<sup>2</sup> transita per bridge 1 il filtering database di bridge 1 annota la presenza di A su sottorete 2.

3. Supponiamo che dopo qualche secondo B invii un pacchetto ad A. Il pacchetto riesce ad arrivare a destinazione? Sempre?

Se l'entità di forwarding del bridge 2 prende la decisione di inoltrare (B->A) in un istante in cui (A->B)<sup>1</sup> è appena transitato per bridge 2 e l'entità di forwarding di bridge 1 prende la decisione di inoltrare (B->A) in un istante in cui (A->B)<sup>2</sup> è appena transitato per bridge 1 allora il pacchetto (B->A) non viene trasmesso sulla sottorete 1. In tutti gli altri casi il pacchetto è recapitato ad A almeno una copia.

Una situazione del genere in cui un pacchetto continua a circolare nella rete inondando di copie tutti gli host è detta broadcast storm, e può portare all'aumento esponenziale del traffico di rete, con conseguenze critiche!!!

## Bridge e prestazioni, gli Switch

Sul mercato esistono diversi tipi di bridge, a seconda dal tipo di rete in cui vengono impiegati; la differenza principale tra questi sta nelle loro prestazioni, che possono influenzare direttamente le prestazioni dell'intera lan. I parametri su cui si distinguono diversi bridge sono sostanzialmente due:

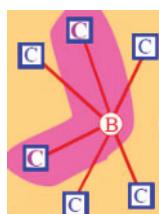
- Numero massimo di pacchetti processabili al secondo;
- Tempo medio di latenza, tempo di attraversamento del bridge da parte di un pacchetto (dall'ingresso del primo bit alla sua ritrasmissione)

È preferibile, inoltre, che il bridge operi a full speed, ovvero con parametri pari al massimo teorico di velocità di trasmissione in base al mezzo trasmisivo. Una delle difficoltà maggiori per perseguire tale obiettivo sta nella gestione dei pacchetti più piccoli: più un pacchetto è corto, infatti, e più alta sarà la frequenza con cui il bridge dovrà prendere decisioni di filtraggio; eventuali test di prestazioni, quindi, devono essere effettuati con pacchetti di lunghezza minima.

Dagli studi condotti, un bridge in una rete 802.3 è full speed se inoltra 14880 pacchetti (di dimensione minima) al secondo; il tempo di latenza è, quindi, funzione della lunghezza del pacchetto.

## Differenze tra bridge e switch

Da un punto di vista di ottimizzazione e di prestazioni, abbiamo visto come il caso migliore di utilizzo di un bridge sia quello in cui ciascun segmento è costituito da un unico calcolatore, ovvero gli host sono direttamente connessi al bridge. Poiché praticamente tutte le reti moderne sono configurate in questo modo, d'ora in poi non parleremo più di bridge, ma di switch (come abbiamo visto nell'esempio sopra). Un bridge viene quindi utilizzato per connettere diversi segmenti di rete, ciascuno dei quali è costituito potenzialmente da molti calcolatori, sostituendo il repeater, mentre uno switch viene collegato direttamente ai singoli host.



La differenza principale con lo switch è essenzialmente nel numero di porte: un bridge possiede al massimo una decina di porte, mentre uno switch può arrivare fino ad alcune centinaia nei modelli più complessi. Questa differenza ha a sua volta effetto sulla dimensione dei domini di collisione: entrambi i dispositivi permettono di ridurre la dimensione, ma lo switch, come abbiamo detto, arriva al caso limite, in cui riduce una parte della rete ad un insieme di domini di collisione di dimensioni minime, al limite costituiti ciascuno da un singolo nodo. Questa situazione consente di ridurre drasticamente le collisioni, ma la contropartita è la grande quantità di cavi necessari a collegare ogni singolo nodo allo switch, piuttosto che i nodi ad alcuni hub e questi ultimi allo switch.

Un'altra differenza sta in un particolare caso del processo di learning: quando non conosce l'host destinatario di un frame (come succede per il primo frame dopo il bootstrap della macchina), lo switch lo inoltra su tutte le sue linee, compresa quella di entrata, realizzando broadcasting al contrario del flooding del bridge.

Come vedremo più avanti, anche all'impiego degli switch (che essendo dispositivi full duplex non generano conteste del mezzo trasmisivo), e sostituendoli agli hub, oggi le reti Ethernet originali in cui si faceva csma/cd sono quasi del tutto sparite, e le uniche collisioni rilevabili in una rete in cui tutte le connessioni sono realizzate su cavi punto-punto sono quelle che possono occorrere tra un host e uno switch o tra due switch contigui.

## Frammentazione dei pacchetti a livello 2?

Nel realizzare una connessione tra due lan diverse tra loro, un bridge/switch deve risolvere il problema del massimo pdu accettabile. Se ad un singolo switch sono connessi diversi tipi di lan 802 (ad esempio Ethernet da una parte e Token Ring dall'altra), questo dovrà decidere se accettare o meno i pacchetti provenienti dalla Token Ring le cui dimensioni superano il massimo consentito per una pdu Ethernet (ovvero, una lunghezza del campo dati massima di 1500 Byte). Alcuni switch permettono di ritrasmettere pdu che eccedono la lunghezza massima frammentando i pacchetti in più unità.

Ricordiamo però che la frammentazione dei dati in pdu è un'operazione che compete al livello di Trasporto (livello 4) nella pila ISO-OSI; un'operazione del genere svolta a livello Data Link costituisce una violazione della gerarchia. Nonostante tutto, pur di evitare di perdere pacchetti, alcuni switch usano la frammentazione e aggiungono un campo di info ad ogni spezzata per numerare i sottopacchetti, agevolando la fase di ricostruzione.

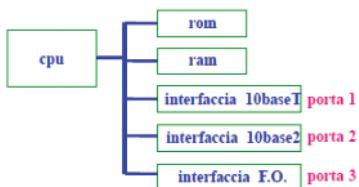
## prestazioni dei bridge

- è preferibile che un bridge sia full speed: parametri pari al massimo teorico
- difficoltà: più corti sono i pacchetti e più è alto il numero di decisioni da prendere nell'unità di tempo
  - eventuali esperimenti di verifica vanno fatti con pacchetti di lunghezza minima
  - in IEEE 802.3 a 10 Mbit/sec. un bridge è full speed se processa 14880 pacchetti al secondo per ogni porta (perché?)
- il tempo di latenza è anche funzione della lunghezza del pacchetto

## Architettura degli switch

### • Architettura fisica

Una delle prime domande che ci poniamo, guardando uno switch nella realtà, è la seguente: dove viene memorizzata la tabella di instradamento? In primo luogo consideriamo il fatto che uno switch è pur sempre un computer dotato, oltre che di una CPU e di memorie primarie e secondarie, volatili e non, anche di una o più **schede di rete** che permettono di interfacciarsi con i vari mezzi trasmittivi IEEE 802; ciascuna tipologia di mezzo necessita infatti di una porta specializzata. Sugli switch moderni, in realtà, è molto comune trovare porte che supportano più standard, come ad esempio le porte **Gigabit Ethernet** che funzionano con cavi RJ-45 a velocità di 10, 100 o 1000 Mbps.



Diverse versioni di moderni switch Cisco.

Com'è possibile vedere dalla figura, alcuni switch sono dotati di più di un modulo, disposti orizzontalmente o verticalmente, o addirittura di più alimentatori, per permetterne un funzionamento continuo in caso di guasti.

Per gli switch conformi allo standard 802.1D (che definisce la tipologia *transparent bridge*), tutte le istruzioni del "programma" dello switch sono contenute nella **rom**, mentre nella **ram**, oltre a venir costruite le tabelle di instradamento, vengono conservati i pacchetti in un **buffer** e tutte le eventuali strutture ausiliarie.

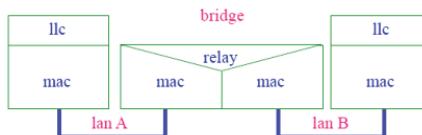
Un'altra soluzione, adottata soprattutto negli switch di fascia alta, sta nel dotarlo di una o più **schedeasic**, ciascuna in grado di risolvere e gestire localmente parte dell'instradamento, favorendo di fatto una maggiore resistenza ai guasti e modularità dello switch.

### • Architettura logica: IEEE 802.1D

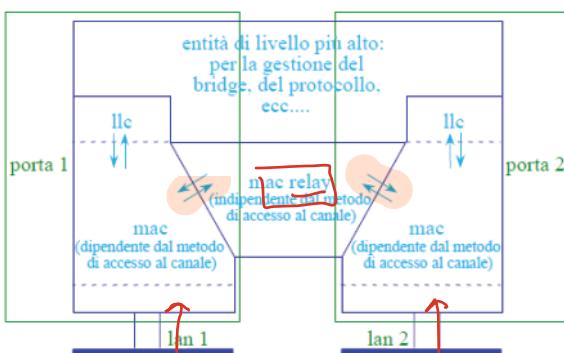
Tornando a considerare il nostro bridge/switch come macchina *OSI* a due livelli, faremo riferimento allo standard **IEEE 802.1D**, standard derivato dall'**802.1 (Higher Layer and Management)**, il cui obiettivo è quello di evitare il loop di switching e i conseguenti broadcast storm nelle reti Ethernet (*come quello che abbiamo visto nell'esempio della rete senza spanning tree*).

Dividiamo l'architettura logica del bridge/switch nelle seguenti entità funzionali:

- Una **funzione di distribuzione (relay entity)** che collega le porte di accesso;
- Almeno due **porte di accesso** per il traffico, appartenenti ciascuna ad una lan differente;
- Funzioni ausiliarie di livello superiore (**higher-layer entities**), tra le quali rientrano ad esempio i protocolli di *Spanning Tree*.



mac relay entity è quella parte che consente ai pacchetti di transitare da un mac all'altro

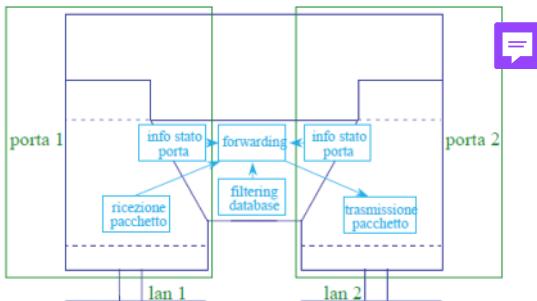


Ogni porta riceve pacchetti dalla lan (1) a cui è direttamente collegata, e li trasferisce alla relay entity (2) che applica le regole per la distribuzione dei pacchetti tra le diverse lan a cui fanno capo le porte, includendo il learning e il filtraggio dei pacchetti, (3) e trasferisce il pacchetto (4) verso la porta in uscita relativa alla lan di destinazione.



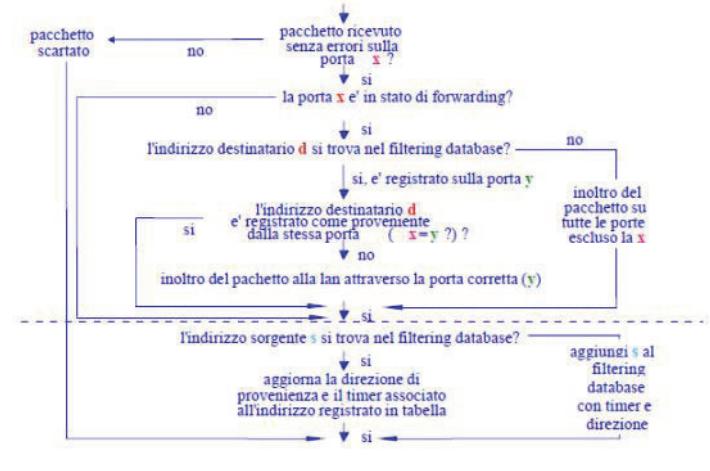
La **relay entity** (o **mac relay**) implementa tre elementi chiave:

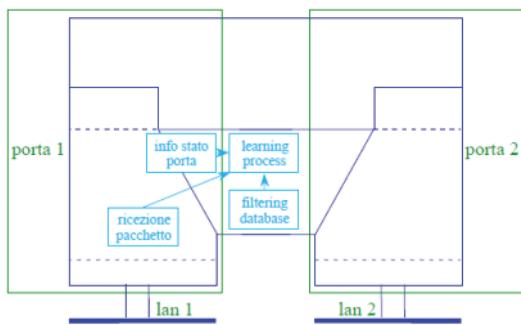
- Il processo di **forwarding** dei pacchetti, selezionando le porte sulla base del loro stato di lavoro e delle informazioni presenti nel **filtering database / forwarding table**;



- Il processo di **auto-learning**, che consente, all'analisi dell'indirizzo della sorgente del pacchetto, di aggiornare il **filtering database** tenendo sempre conto dello stato delle porte;
- Il **filtering database**, che contiene le informazioni relative alle regole dello **store and forward** dei pacchetti e interagisce coi processi di **forwarding** e **learning**, fornendo l'informazione di quali porte sono abilitate a trasmettere/ricevere un determinato pacchetto.

Come abbiamo visto, la **forwarding table** contiene **entry statiche e dinamiche**; queste ultime vengono aggiornate ad intervalli regolari, poiché il processo di **learning** può essere ripetuto più volte. Se, ad esempio, un host dovesse spostarsi all'interno della lan, la sua posizione nella tabella di instradamento dovrebbe essere aggiornata; solitamente, tutto il processo di learning viene ripetuto in uno switch ogni cinque minuti.



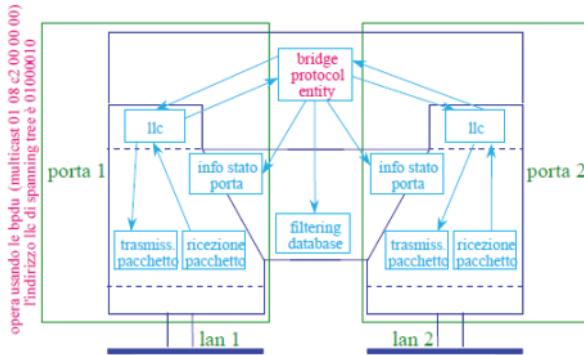


Le **porte** dello switch hanno ciascuna un indirizzo mac, e sono numerate progressivamente a partire da 1; l'indirizzo fisico dello switch è solitamente uguale proprio all'indirizzo mac della porta 1, che lo identifica. **Una porta è un dispositivo con uno stato**, che può essere assegnato in modo statico da un amministratore di rete o in modo dinamico da parte dell'algoritmo di *spanning tree*.

Per ogni porta sono possibili i seguenti stati:

|                   |   |
|-------------------|---|
| <b>Forwarding</b> | La porta è abilitata a trasmettere pacchetti e a partecipare al processo di learning                  |
| <b>Discarding</b> | La porta è spenta, oppure non è stata abilitata né a scambiare pacchetti né a partecipare al learning |
| <b>Learning</b>   | La porta è abilitata a partecipare al learning, ma non a trasmettere pacchetti                        |
| <b>Blocking</b>   | La porta è in <i>stand by</i> in quanto non facente parte dello <i>spanning tree</i> della rete       |
| <b>Disabled</b>   | La porta è disabilitata   |

Le **entità di protocollo** costituiscono tutte le entità di livello superiore, tra cui quella già citata per il calcolo dello *spanning tree* ed il controllo generale dello switch. Queste entità fungono da ponte di collegamento tra un bridge e l'altro per sviluppare l'interconnessione tra le tabelle di instradamento di ciascuno switch; si tratta macchine in grado di realizzare una **backbone**, che sono definite **stackable**, poiché rendono le porte interoperabili, rendendo molto flessibile la modalità di link. Per questo motivo, le entità di protocollo fanno uso dei livelli *IIC* di ciascuna porta.



#### Modalità *cut-through* (contro *store-and-forward*)

Per concludere, abbiamo visto all'inizio del capitolo come la tipologia di instradamento di cui bridge e switch fanno uso sia la *store and forward*. Se volessimo evitare di dover memorizzare i pacchetti per aumentare la velocità della rete, potremmo far uso di una modalità di instradamento alternativa; si tratta della cosiddetta ***cut-through***, il cui funzionamento è il seguente:

*Se, in una trasmissione, la porta di destinazione è libera, uno switch cut-through inizierà a trasmettere ancor prima di aver ricevuto l'intero pacchetto dalla porta di provenienza.*

Negli switch *store and forward* un pacchetto prima di essere ritrasmesso, viene riletto completamente e ne viene calcolato il **cyclic redundancy check**, il quale poi viene confrontato con il campo *fcs* del frame, e se i due valori corrispondono il frame viene mandato al destinatario. Negli switch *cut-through* vengono letti solo i primi 64 Byte del frame in modo da rilevare solo alcune anomalie; ciò porta a un notevole incremento di prestazioni, tuttavia sacrificando (quando è possibile adoperare questa modalità) la trasmissione corretta di alcuni pacchetti.

Con la *cut-through* è infatti impossibile calcolare l'*fcs* prima di ritrasmettere il pacchetto, e per tal motivo, se sono presenti errori, verranno ritrasmessi insieme al pacchetto. Inoltre, gli switch *cut-through* non possono essere impiegati nelle reti in cui:

- Lo switch è tra reti con diverso protocollo lan;
- Lo switch è tra reti con protocollo uguale ma a diversa velocità (**switching asimmetrico**);
- La porta di destinazione è occupata.

- l'administrator può mettere ogni porta in stato di **enabled** (attiva) o **disabled**
- una porta attiva può essere in stato di **forwarding** o di **blocking**, a causa dell'algoritmo di *spanning tree*
- le porte hanno un indirizzo MAC e sono numerate progressivamente nel bridge a partire da 1, l'indirizzo del bridge è uguale all'indirizzo MAC della porta 1

TABELLA DI INSTRADAMENTO

- la tabella contiene entry (righe) **statiche** ed entry **dinamiche**

- il processo di learning si basa sugli indirizzi mittente dei pacchetti ascoltati
- il valore di default per la sopravvivenza delle entry dinamiche è 5 minuti

consideri "buono" un mac address per 5 minuti

I possibili passaggi da uno stato all'altro sono i seguenti:

| Stato di partenza | Stato di arrivo        |
|-------------------|------------------------|
| Inizializzazione  | Blocking               |
| Blocking          | Listening<br>Disabled  |
| Listening         | Learning<br>Disabled   |
| Learning          | Forwarding<br>Disabled |
| Forwarding        | Disabled               |

Il processo di learning è attivo nella misura in cui la porta su cui è stato ricevuto il pacchetto è attiva.

Se il pacchetto viene ricevuto da una porta non attiva quel processo di learning non viene utilizzato.

BPDUs corrispondono all'indirizzo multicast 01 08 c2 00 00 00. Arriva il pacchetto sulla porta, se è un pacchetto in transito si passa per la mac relay entity, se è destinato al bridge si passa a IIC che riconosce l'indirizzo lo passa all'entità di protocollo che si occupa del calcolo dello spanning tree.

evita lo store e fa direttamente il forward diminuendo il tempo di latenza

## 7 - Evoluzione di Ethernet

18 ottobre 2021

### Gli obiettivi: velocità e modalità duplex

La tendenza generale, nell'evoluzione di Ethernet, è stata quella di sviluppare nuove tecnologie per riuscire ad ottenere reti locali più performanti. Così come l'evoluzione dei computer ha portato ad un utilizzo di file sempre più grandi, anche lo scambio di questi ha richiesto **velocità di trasmissione** che non fungessero da collo di bottiglia (con velocità intendiamo sempre la *larghezza di banda o throughput della rete*, espresso in bit al secondo).

Nello sviluppo della velocità, commissioni e aziende hanno puntato fondamentalmente a due obiettivi:

- Velocità di **100Mb/s** sulla maggior parte dei computer connessi in rete locale;
- Velocità di **1Gb/s** sulle dorsali più importanti della rete globale.

Come preannunciato, inoltre, dispositivi come gli switch si sono ampiamente diffusi sulle reti e hanno quasi del tutto rimpiazzato i gloriosi **hub**; il vantaggio fondamentale degli switch, lo ripetiamo, sta nell'assegnare ciascuna sua porta ad un solo dispositivo, in un rapporto 1:1, e di interfacciarsi e combinarsi con altri switch, permettendo l'utilizzo di tecnologie veloci come la cosiddetta **full duplex**.



Una macchina full-duplex è dotata di **due canali**, uno per la **ricezione** e l'altro per la **trasmissione**, che funzionano contemporaneamente senza dare luogo a **collisioni**.

Questo aspetto è importantissimo, poiché **spariscono tutti i problemi legati alla dimensione del pacchetto e alla lunghezza della rete**, e di fatto le vecchie reti **half duplex**, su cui era necessario fare **csma/cd**, sono quasi sparite del tutto. La modalità **full duplex** è realizzata grazie agli **switch**, siano essi connessi tra loro o ai singoli host, ed è anche per questo motivo se gli **switch** hanno avuto una diffusione capillare.

Come abbiamo già detto, il mezzo di trasmissione che più si è diffuso è quello derivato dal doppino telefonico, o **100Base-T utp**; si tratta di uno standard molto versatile, che può funzionare a diverse velocità con la stessa porta di rete e **favorisce l'autonegoziazione tra le schede** che sono agli estremi dello stesso cavo; è grazie al 100Base-T se è possibile implementare tutte e due le modalità duplex.

Vediamo tre diversi standard derivati dalla prima versione di 802.3:

#### • Fast Ethernet (IEEE 802.3u)

A partire da questa versione si è cominciato ad adottare lo standard per doppino telefonico **100Base-T** e lo standard per fibra ottica **100Base-F**, di fatto aumentando di 10 volte l'ampiezza di banda della rete rispetto alla precedente tecnologia; quest'ultima ha continuato ad essere supportata permettendo il funzionamento delle porte ad entrambe le velocità (**10/100 Mbps**), per garantire la trasformazione graduale degli impianti. La scelta della velocità viene determinata dalla fase di **autonegoziazione** delle schede che sono agli estremi dello stesso cavo, così come la scelta della modalità di comunicazione (half duplex o full duplex).

Questo aumento di banda, tuttavia, ha creato non pochi problemi: inviare un pacchetto 10 volte più velocemente significa "accorciare" la trasmissione (più informazioni al secondo), e di conseguenza la rete intera, per rispettare i vincoli di **csma/cd**, è diminuita di **diametro**.

Se la dimensione minima dei pacchetti (512 bit) non cambia, ad un aumento di velocità della corrisponde una diminuzione del roundtrip delay. Nel caso di 100 Mbps, l'estensione massima della rete sarà 10 volte più corta di una a 10 Mbps.

Grazie però ad una massima diffusione degli switch, è stato possibile risolvere questo problema lasciando sostanzialmente tutto intatto, dalla dimensione minima dei pacchetti al funzionamento di **csma/cd**. Riassumiamo quindi le differenze principali tra IEEE 802.3 e 802.3u:

|                          | 802.3   | 802.3u  |
|--------------------------|---------|---------|
| Velocità (in Mb/s)       | 10      | 100     |
| Bit time (in ns)         | 100     | 10      |
| Inter-packet gap (in µs) | 9,6     | 0,96    |
| Slot time per il backoff | 51,2 µs | 5,12 µs |

Il **bit time** definisce il tempo che impiega un bit per essere spedito da una scheda di rete che opera ad una velocità predefinita. Una scheda di rete a 10 Mbps può spedire 1 bit ogni 0,1 µs, quindi il suo bit time è 100 nanosecondi.

#### • Controllo di flusso (IEEE 802.3x)

Nel trattare gli switch, abbiamo visto come fosse possibile, grazie al loro impiego, risolvere il problema di connettere segmenti di rete che operano a diverse velocità. Consideriamo il seguente caso:



L'host C1 invia dei pacchetti all'host C2 connesso tramite lo switch verde ad una velocità inferiore; se è pur vero che lo switch effettua **store and forward**, dal momento che riceve pacchetti 10 volte più velocemente della velocità a cui può ritrasmetterli, può capitare che il suo buffer venga saturato. Andando rapidamente in saturazione il buffer, in quanto trasmette a un decimo della velocità di ricezione, molti pacchetti verranno scartati e dovranno essere ritrasmessi.

È per risolvere inconvenienti di questo tipo che è nata la versione **802.3x**, che introduce in Ethernet un

#### • tendenza

- 10 Gb/s sui computer
- 10 Gb/s – 100 Gb/s sulle dorsali LAN
- più di 100 Gb/sec sulle dorsali WAN
- motivazioni
  - centralità della rete in tutte le attività
- situazione attuale
  - ampia diffusione degli switch, ma repeater ancora presenti
  - sui computer, ethernet a 100 Mb/s a 1 Gb/s e a 10 Gb/s ampiamente diffusi
  - sulle dorsali, ethernet a 10 Gb/s ampiamente diffuso

Nelle vecchie reti **Ethernet half-duplex csma/cd**, la caratteristica peculiare era la comunicazione "a turni"; oggi questa prassi è sparita quasi ovunque e le trasmissioni sono tutte di tipo **bidirezionale contemporaneo, sia tra computer e switch che tra due switch**.

- sostituzione, quasi ovunque, dei repeater con degli switch (bridge)
- switch con una porta dedicata ad ogni calcolatore e con una porta dedicata ad ogni collegamento con altri switch
- se il calcolatore e lo switch (o due switch) sono collegati da un cavo dedicato che consente la comunicazione contemporanea nelle due direzioni, allora le collisioni, su quel tratto di cavo, non ci sono più
- funzionamento **full duplex**: trasmissione bidirezionale contemporanea tra computer e switch e tra switch e switch
- spariscono le collisioni; ethernet csma/cd – detto anche **half duplex** (si parla a turno) diventa **full duplex**
- spariscono i vincoli sulla distanza massima tra le stazioni dovuti a csma/cd

- lo sviluppo di ethernet ha portato all'uso di meccanismi di controllo di flusso, soprattutto per gli switch
- esempio: consideriamo uno switch multi-porta; una porta è a 100Mb/s e ospita un server e le altre porte sono a 10Mb/s
- la richiesta di un file da parte di un client richiede il trasferimento di pochi byte verso il server, ma richiede il trasferimento di un notevole volume di dati verso il client
- il buffer dello switch, visto che può ritrasmettere ad un decimo della velocità di ricezione può andare rapidamente in saturazione

pacchetto di controllo, mai visto prima d'ora; questo pacchetto prende nome di **pause frame**, una richiesta di pausa lunga 512 bit riconoscibile grazie al campo L/T configurato con i Byte hex 88-08; Inoltre, in questo pacchetto si specifica quanto tempo vuole essere concesso di pausa (pause time).

È comunque da specificare che questo pacchetto viene usato solo nelle 802.3 che operano in modalità full duplex; in reti half duplex con csma/cd, lo switch crea delle finte collisioni verso un host, in modo che questi si astenga dal trasmettere per un po'. L'uso del pause frame è quindi necessario per realizzare il controllo di flusso nelle reti Fast e Gigabit Ethernet, oltre ad essere stato usato per altri mac tra cui i Token Ring e gli FDDI (in cui veniva introdotto un ulteriore strato denominato *mac control*, dedicato esclusivamente alla gestione del frame).

Per concludere vediamo le differenze tra un pause frame e un pacchetto per lo spanning tree 802.1D:

|   | Pause frame 802.3x | <b>STP frame (per bridge) 802.1D</b>                     |
|---|--------------------|--|
| Indirizzo multicast ( <i>mac-dsap</i> ) | 1-80-C2-00-00-01   | 1-80-C2-00-00-00   |
| Length/type                             | 0x8808             | /  |
| Contiene                                | Pause time         | Dati per lo spanning tree provenienti da un altro bridge |

Ricordiamo che per valori più grandi di 1500, i due byte del campo length vengono usati per indicare il tipo di frame Ethernet o altri frame di controllo; questo ne è un esempio!

La presenza di pacchetti MAC che non portano dati ma che contengono solo informazioni di controllo (control frame) è una novità; viene introdotto un nuovo sottostrato di MAC denominato MAC control.

### • Gigabit Ethernet (IEEE 802.3z)

Questo standard, infine, rappresenta un'evoluzione naturale di Fast Ethernet; si tratta infatti di un'ulteriore aumento delle velocità di rete. È stato proposto inoltre per risolvere i problemi che si avevano nella gestione delle *backbone* sia nel caso delle reti lan che delle wan: sulle dorsali esistevano ancora mac 802 diversi da Ethernet, come FDDI e ATM, e ciò richiedeva l'utilizzo di meccanismi di conversione del formato dei pacchetti per adattarli ai mac delle dorsali, diminuendo di fatto l'efficienza nei collegamenti a lunga distanza.

Con Gigabit Ethernet, utilizzando principalmente trasmissioni tramite fibra ottica **1000Base-F** o cavi in rame **1000Base-T**, si è raggiunta la significativa velocità di **1 Gb/s**; con questa, però, si verificano di nuovi i problemi che abbiam visto per Fast Ethernet. Lasciando infatti invariata la lunghezza del pacchetto più corto, e aumentando i bit al secondo, la rete diminuisce ulteriormente di estensione massima; ovviamente una rete Gigabit Ethernet di soli 20 metri sarebbe poco utile, così come realizzare una rete Gigabit utilizzando uno switch ogni 20 metri!!

Per questi motivi si è reso necessario porre delle modifiche ai parametri di *csma/cd*. Oltre ad adottare un'altra codifica dei bit (la 8B/10B) rispetto alla 4B/5B, è stato deciso di aumentare lo **slot-time** da 512 bit a **512 byte** (4096 bit), che era rimasto invece invariato nello standard a 100 Mb/s. La lunghezza del pacchetto più corto rimane sempre di 512 bit, per mantenere la completa compatibilità con le reti esistenti, ma per pacchetti di lunghezza inferiore ai 4096 bit che transitano sui mac di questo standard viene aggiunta un'estensione in coda al pacchetto (dopo la *fcs*) con caratteri speciali, fino a raggiungere lo slot time. Se un ricevitore, dunque, dovesse ricevere un pacchetto più corto del minimo, questo andrebbe scartato anche se il codice *crc* che contiene è corretto.

In sostanza per trasmettere un pacchetto di 512 bit, in Gigabit Ethernet se ne trasmettono 4096 (8 volte in più) ad una velocità che è 10 volte maggiore di quella di Fast Ethernet.

In ogni caso, visti gli investimenti notevoli per realizzare dorsali Gigabit, solitamente si utilizza la modalità di connessione **full duplex**, utilizzando anche le tecnologie di controllo di flusso **802.3x**. La rete Gigabit oggi rappresenta lo standard per le dorsali delle reti locali e può essere realizzata con più mezzi trasmisivi, vediamo le principali differenze:

|             | Cavo         | Distanza massima |
|-------------|--------------|------------------|
| 1000Base-SX | Fibra ottica | 500 m            |
| 1000Base-LX | Fibra ottica | Vari km          |
| 1000Base-ZX | Fibra ottica | Vari km          |
| 1000Base-T  | Doppino      | 100 m            |
| 1000Base-TX | Doppino      | 100 m            |

In conclusione a questo capitolo, vediamo due standard ancor più recenti derivati da Gigabit Ethernet:

- **10 Gigabit Ethernet (IEEE 802.3ae, 802.3ak)**: sono standard nati tra il 2002 e 2008 esclusivamente dedicati alle connessioni a **10 Gb/s** full duplex, e non supportano quindi csma/cd. Le dimensioni minime e massime dei pacchetti non variano rispetto a 802.3z; variano invece i mezzi trasmisivi chiamati **10GBase-R** per la fibra e **10GBase-T** per il rame (quest'ultimo offre collegamenti fino a 100 m);
- **40 Gigabit e 100 Gigabit Ethernet (IEEE 802.3ba)**: è uno standard nato tra il 2007 e il 2010, che rappresenta un ulteriore passo in avanti rispetto alla 10 Gigabit, prevedendo per la prima volta due velocità di funzionamento indipendenti. Lo standard a 100 Gb/s servirà per creare dorsali di comunicazione ad alta velocità, ma rispetto a quello a 40 Gb/s richiederà più energia, apparati più complessi e quindi complessivamente sarà più costoso da installare e da gestire. Tutti gli altri parametri (modalità duplex e lunghezza dei pacchetti) non variano rispetto allo standard a 10 Gb/s.

Alcuni standard

- **1000BASE-SX** fibra 500 m
- **1000BASE-LX** fibra vari km
- **1000BASE-ZX** fibra vari km
- **1000BASE-T** rame 100 m
- **1000BASE-TX** rame 100 m
- standard nato tra il 2002 e il 2008
- fibra
  - **10GbaseR**, ....
- rame
  - **10GbaseT** (i soliti 100 m), ....

40 gigabit e 100 gigabit ethernet  
in corso di definizione 200/400 Gb

### Frame bursting

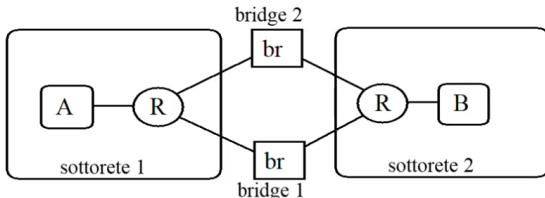
Se un mac 802.3z ha più pacchetti da trasmettere esso può inviarli tutti in un'unica trasmissione csma/cd, grazie ad una tecnica detta **frame bursting**; questa consiste nell'"incollare" i pacchetti tra loro spaziandoli con dei caratteri speciali; l'operazione di incollaggio viene svolta dal mac come segue:

- Si controlla se l'invio del primo pacchetto sia andato a buon fine (non sono state rilevate collisioni);
- Si verifica che il **timer di bursting** non abbia raggiunto il massimo;
- Si inviano 96 bit di caratteri speciali per codificare l'*inter-packet gap*;
- Si comincia ad inviare il secondo pacchetto e si resetta il timer di bursting.
- **standard per il mercato automotive**
  - **802.3bw** (2015) **100baseT1** Fast su singola coppia
  - **802.3bp** (2016) **1000baseT1** Gigabit su singola coppia
- **power over ethernet**
  - **802.3af** (2003)
  - **802.3at** (2009)

Esempi power over ethernet: telecamere

## Esercizio - Bridge e reti con cicli

Considera la seguente rete locale IEEE 802.3.



Le R indicano dei repeater e A e B sono due computer.

Le connessioni tra le apparecchiature sono tutte su cavi utp.  
Supponi che i bridge siano conformi allo standard ma che non siano in grado di calcolare uno spanning tree della rete.  
In altri termini supponi che tutte le porte dei due bridge siano pienamente attive. I bridge sono stati appena accesi.

A invia un pacchetto a B.

- 1 Quanti pacchetti circolano nella rete dopo l'invio del pacchetto e che percorso seguono?
- 2 Cosa succede al filtering database di bridge1 e di bridge2?
- 3 Supponiamo che dopo qualche secondo B invii un pacchetto ad A. Il pacchetto riesce ad arrivare a destinazione? Sempre?
  - 1 Il pacchetto ( $A \rightarrow B$ ) viene ricevuto da entrambi i bridge, i quali, entrambi, non trovando B nel proprio filtering database, intendono spedire ( $A \rightarrow B$ ) sulla sottorete 2. Indichiamo con ( $A \rightarrow B$ )' e ( $A \rightarrow B$ )'' le due copie del pacchetto. Supponiamo (il caso opposto è identico) che nella contesa del dominio di collisione della sottorete 2 il bridge1 riesca a trasmettere prima del bridge2. Il pacchetto ( $A \rightarrow B$ )' viene ricevuto da B e dal bridge2. Quindi il bridge2 può accedere alla sottorete 2 ed inviare ( $A \rightarrow B$ ). I pacchetti ( $A \rightarrow B$ )' e ( $A \rightarrow B$ )'' ciclano continuamente nella rete. In particolare ( $A \rightarrow B$ )' segue il percorso (antiorario nel disegno) bridge1-sottorete2-bridge2-sottorete1 mentre ( $A \rightarrow B$ )'' segue il percorso (orario nel disegno) bridge2-sottorete2-bridge1-sottorete1.
  - 2 Ogni volta che ( $A \rightarrow B$ )' transita per bridge2 il filtering database di bridge2 annota la presenza di A su sottorete2. Ogni volta che ( $A \rightarrow B$ )' transita per bridge1 il filtering database di bridge1 annota la presenza di A su sottorete1. Ogni volta che ( $A \rightarrow B$ )'' transita per bridge2 il filtering database di bridge2 annota la presenza di A su sottorete1. Ogni volta che ( $A \rightarrow B$ )'' transita per bridge1 il filtering database di bridge1 annota la presenza di A su sottorete2.
  - 3 Se l'entità di forwarding di bridge2 prende la decisione di inoltro di ( $B \rightarrow A$ ) in un istante in cui ( $A \rightarrow B$ )' e' appena transitato per bridge2 e l'entità di forwarding di bridge1 prende la decisione di inoltro di ( $B \rightarrow A$ ) in un istante in cui ( $A \rightarrow B$ )'' e' appena transitato per bridge1 allora il pacchetto ( $B \rightarrow A$ ) non viene trasmesso sulla sottorete1. In tutti gli altri casi il pacchetto e' recapitato ad A in almeno una copia.

Le porte sono tutte in stato di forwarding.

Il pacchetto parte da A e arriva sul repeater (non fa store e forward) e li ripete sia su bridge 1 che su bridge 2. I bridge fanno passare il pacchetto perché non sanno dove sia B.

Il repeater riceve un pacchetto da due parti diverse, collisione sul repeater. Una delle due passerà e l'altro dovrà aspettare il suo turno. Chi riceve il pacchetto? B lo vede (c'è il suo mac address) ma viene ricevuto anche dal bridge che non l'ha mandato che a sua volta lo manda sull'altra porta.

Sono nate due istanze del pacchetto che continuano a girare per la rete per un tempo indefinito.

C'è un problema prestazionale (la banda viene usata tantissimo e inutilmente da questi pacchetti).

Ma c'è anche un problema funzionale.

Cosa succede al filtering database di bridge 1 e bridge 2?

A sta a sinistra, B sta a destra.

Il pacchetto di B ad A viene ricevuto? I bridge se sono in un momento in cui entrambi credono che A sia a sinistra lo fanno passare ma negli altri casi non passa.

# Esercizi di Ricapitolazione - I

18 ottobre 2021

## Pacchetti di dimensione massima

- A) Quanto tempo impiega una stazione Ethernet a 10 Mbps a trasmettere un pacchetto di dimensione massima, ovvero di 1518 Bytes?  $1518 \cdot 8 \text{ BIT} / (\text{BIT/T})$

Per prima cosa convertiamo i Byte in bit, quindi moltiplichiamo per 8 il numero di byte e moltiplichiamo per il *bit time* di una rete Ethernet a 10 Mbps (100 nanosecondi per bit)

$$1518 \text{ [byte]} \cdot 8 \text{ [bit/byte]} \cdot 10^{-7} \text{ [s/bit]} =$$

$$= 12144 \cdot 10^{-7} \text{ [s]} =$$

$$= 1,2144 \text{ [ms]}$$

- B) Quanto sarebbe lungo un pacchetto di dimensione massima se fosse trasmesso su un conduttore di lunghezza infinita?

Dalla domanda precedente abbiamo il tempo che tale pacchetto impiega per essere trasmesso, ovvero 1,2144 ms; moltiplicando per la 2/3 della velocità della luce, ovvero  $2 \times 10^8 \text{ m/s}$ , otteniamo la "lunghezza" del pacchetto:

$$1,2144 \cdot 10^{-3} \text{ [s]} \cdot 2 \cdot 10^8 \text{ [m/s]} = S * (\text{m/s})$$

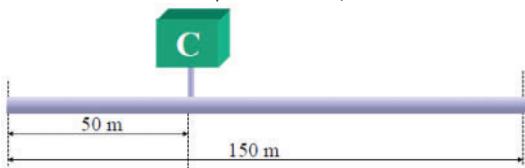
$$= 2,4288 \cdot 10^5 \text{ [m]} =$$

$$= 242,88 \text{ [Km]}$$

## Disabilitazione del rilevamento delle collisioni

Il calcolatore C è situato in una rete IEEE 802.3 **10Base-2**, composta da un unico dominio di collisione e senza ripetitori; la rete è lunga 150 metri e C è posto a 50 metri da uno degli estremi. In un certo istante C inizia a trasmettere.

Dopo aver trasmesso quanti bit C potrebbe disabilitare il circuito di rilevamento di collisione senza provocare malfunzionamenti del protocollo *csma/cd*?



Per poter rilevare una collisione correttamente, il calcolatore C deve lasciare il tempo al segnale di arrivare all'estremo di rete più lontano e tornare indietro; in altre parole, dobbiamo calcolare il *round trip delay* della rete. Le nostre ipotesi sono che:

- L'estremo della rete più lontano da C è a 100 metri, quindi per tornare indietro dovrà percorrere una distanza non inferiore a 200 metri;
- La velocità di propagazione del segnale sul mezzo trasmisivo è 2/3 della velocità della luce ( $2/3 \times 300.000 \text{ km/s}$ , ovvero  $200.000 \text{ km/s} - 2 \times 10^8 \text{ m/s}$ ).

Dunque è necessario che il calcolatore C rilevi le collisioni per un tempo pari almeno a:

$$\frac{2 \cdot 10^2 \text{ [m]}}{2 \cdot 10^8 \text{ [m/s]}} = 10^{-6} \text{ [s]} = 1 \mu\text{s}$$

distanza che il segnale deve percorrere  
velocità del segnale sul mezzo trasmisivo  
(2/3 la velocità della luce)

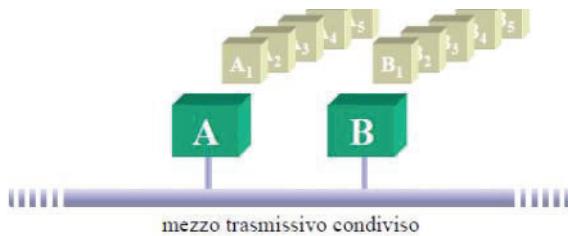
Ci siamo chiesti però dopo quanti bit sia possibile disabilitare il rilevamento delle collisioni; calcoliamo quindi, moltiplicando per il *bit time*, quanti bit vengono spediti nella rete in 1 microsecondo:

$$10^{-6} \text{ [s]} \cdot 10^7 \text{ [bit/s]} = 10 \text{ bit}$$

Ricordiamo che, disabilitando la funzione di rilevamento delle collisioni, il calcolatore C non sarà più in grado di rilevare le *late collision*, e quindi perderà parte della capacità di monitorare il funzionamento della rete. In un'ipotesi reale, non conviene quasi mai disabilitare csma/cd, poiché non sa quanta distanza ci sia due stazioni sul dominio di collisione.

## Csma/cd è un protocollo che garantisce equità?

Vediamo come si comporta il protocollo *csma/cd* nel gestire le priorità delle stazioni che condividono il mezzo trasmisivo. Consideriamo due stazioni A e B su un dominio di collisione Ethernet, entrambe con infiniti pacchetti da trasmettere; numeriamo i pacchetti per stazione come sono indicati in figura:



La rete è inizialmente priva di traffico, e ad un certo istante di tempo A e B tentano di trasmettere contemporaneamente  $A_1$  e  $B_1$ ; le stazioni rilevano entrambe una collisione. Nel corrispondente backoff, A estrae il numero 0, mentre B estrae il numero 1; in questo caso, A riesce a trasmettere  $A_1$  e B deve aspettare che il canale sia disponibile. Nel momento in cui A smette di trasmettere, entrambe le stazioni hanno un pacchetto da trasmettere ( $A_2$  e  $B_1$ ), e quindi si verificherà sicuramente una nuova collisione.

- A) Quali sono le probabilità che, dopo la collisione,
  - a. Si verifichi una nuova collisione?
  - b. B trasmetta e A debba aspettare?
  - c. A trasmetta e B debba aspettare?
- a. A sceglierà il massimo di backoff dall'intervallo [0...1] perché su  $A_2$  ha già avuto **una** collisione, mentre B sceglierà dall'intervallo [0...3] perché su  $B_1$  ha già avuto **due** collisioni. Possiamo calcolare quindi le probabilità del caso in cui si verifichi una terza collisione, considerando che ciò avviene quando entrambe le stazioni scelgono lo stesso numero; la probabilità che A scelga un numero è  $1/2$  (sceglie tra 0 e 1), mentre per B è  $1/4$  (sceglie tra 0, 1, 2, 3), sommiamo i prodotti e otteniamo:

$$P[\text{collisione}] = P[B=0] \times P[A=0] + P[B=1] \times P[A=1] = 1/4 \times 1/2 + 1/4 \times 1/2 = 1/4$$

- b. Affinchè B trasmetta e A debba aspettare, B dovrà selezionare un numero inferiore a quello sorteggiato da A; l'unico caso è che B selezioni 0 e A selezioni 1, quindi ne calcoliamo il prodotto:
 
$$P[B,A] = P[B=0] \times P[A=1] = 1/4 \times 1/2 = 1/8$$
- c. Affinchè A trasmetta e B debba aspettare, A dovrà selezionare un numero inferiore da quello sorteggiato da B; senza ripetere i calcoli, possiamo considerare che questa probabilità è esattamente complementare alle probabilità degli altri due eventi:
 
$$P[A,B] = 1 - P[B,A] - P[\text{collisione}] = 1/8$$

*Da questi risultati possiamo fare un'osservazione preliminare; possiamo notare, infatti, che è più probabile che la stazione che ha già "vinto" un conflitto sul dominio di collisione riesca a trasmettere di nuovo, mentre è molto meno probabile che una stazione che ha dovuto aspettare riesca a trasmettere finalmente il suo pacchetto; ma vediamo un altro caso:*

Assumiamo ora che A riesca a trasmettere il pacchetto  $A_2$ , vincendo la contesa con B. Nel momento in cui A smette di trasmettere  $A_2$ , entrambe le stazioni hanno un pacchetto da trasmettere ( $A_3$  e  $B_1$ ), quindi si verificherà una nuova collisione.

- B) Quali sono le probabilità che, dopo la collisione,
  - a. Si verifichi una nuova collisione?
  - b. B trasmetta e A debba aspettare?
  - c. A trasmetta e B debba aspettare?
- a. Ricalcoliamo di nuovo le probabilità, considerando tuttavia che, mentre l'intervallo da cui il backoff di A sorteggia è invariato, quello di B ora è l'intervallo [0...7]. Di nuovo, la collisione può verificarsi se entrambi i numeri sorteggiati sono uguali e pari a 0 o 1:
 
$$P[\text{collisione}] = P[B=0] \times P[A=0] + P[B=1] \times P[A=1] = 1/8 \times 1/2 + 1/8 \times 1/2 = 1/8$$
- b. Anche in questo calcolo consideriamo che la probabilità che B selezioni 0 è diminuita rispetto a prima:
 
$$P[B,A] = P[B=0] \times P[A=1] = 1/8 \times 1/2 = 1/16$$
- c. Ripetiamo questo calcolo allo stesso modo di prima, complementando con i risultati ottenuti dai punti a e b:
 
$$P[A,B] = 1 - P[B,A] - P[\text{collisione}] = 13/16$$

A questo punto la nostra osservazione è chiara, e ci chiediamo:

- C) Pensi che l'algoritmo di backoff sia equo nel risolvere le collisioni?

*In generale il sistema non è equo. Dopo che B ha avuto n collisioni nel trasmettere lo stesso pacchetto, la probabilità che A vinca la contesa è decisamente maggiore; nel caso generale è*

$$P[A,B] = 1 - P[B,A] - P[\text{collisione}] = 1 - 1/2 \cdot 2^{-n} \cdot 2^{-n} = 1 - 3 \cdot 2^{-(n+1)}$$

E ciò vuol dire che il protocollo csma/cd non garantisce equità tra le stazioni.

Il problema è costituito dal fatto che una stazione trasmittente tenda a "catturare il canale", cioè trovarsi alternativamente negli stati di: tentativo, trasmissione con successo, tentativo, collisione, tentativo, trasmissione con successo, etc...

Occorrerebbe quindi un meccanismo per far sì che la macchina "rilasci il canale", posponendo la propria trasmissione e permettendo ad altre stazioni di trasmettere; un nuovo algoritmo potrebbe essere il seguente:

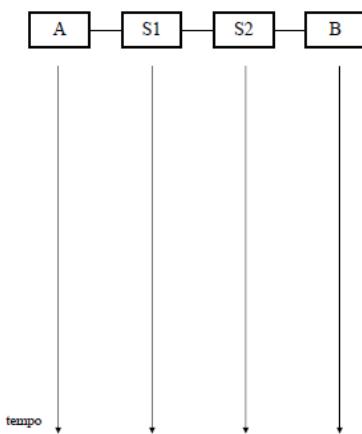
- o  $n$  = numero di collisioni;



- Se  $n = 1$ , allora ritarda la trasmissione per 2 slot;
- Se  $n = 2$ , allora ritarda la trasmissione per 0 slot;
- Se  $n > 2$ , allora usa lo standard *exponential backoff*.

### Switch store and forward

Consideriamo la rete mostrata nella figura seguente:



A e B sono computer,  $S_1$  ed  $S_2$  sono switch; i collegamenti sono realizzati con cavi ethernet *utp* e il ritardo di propagazione tra gli apparati è trascurabile; le schede di rete sono tutte a 10 Mbps (il cui *bit time* è di 100 nanosecondi).

Il computer A deve spedire un file di 100.000 bt al computer B; per farlo, suddivide il file in 100 pacchetti. Supponiamo che il trasferimento del file inizi all'istante  $t=0$ , per ipotesi inoltre:

1. Trascuriamo gli *header* di qualunque tipo dei pacchetti e gli *inter packet gap*;
2. La rete è a completa disposizione del trasferimento del file (il canale è libero);
3. Gli switch non operano in modalità *cut-through*, quindi funzionano normalmente in *store and forward*;
4. Non vengono usati meccanismi di riscontro di alcun tipo.

A) Completa il diagramma temporale in figura, mostrando la sequenza di invio dei pacchetti da parte dei vari apparati;

- Il file è composto da 100.000 bit, ovvero quindi per **ciascuno** dei 100 pacchetti ci saranno  $10^3$  bit. Chiamiamo **I** l'immissione di un pacchetto nella rete, e indichiamo con  **$t_I$**  il tempo di immissione per spedire un pacchetto da A a  $S_1$ .
- Per arrivare a  $S_2$ , lo stesso pacchetto impiegherà un ulteriore tempo  **$t_I$** , quindi il tempo che il pacchetto impiega per essere "immesso" verso B è  **$2t_I$** .
- I pacchetti sono posti, tra un blocco e l'altro, a gradino. Le condizioni sono favorevoli, e abbiamo supposto per ipotesi che non siano presenti header o gap tra i pacchetti, quindi questi avranno forma "**rettangolare**". Il valore di  **$t_I$**  è deducibile dal calcolo del rapporto tra il *bit time* della rete e la dimensione del pacchetto (che abbiamo detto essere di 1000 bit):  

$$t_I = 10^3[\text{b}] / 10^7[\text{s}/\text{b}] = 10^{-4}[\text{s}] = 0,0001 \text{ s}$$

$$1000\text{bit}/10^7(\text{bit/s})=10^{-4} \text{ s}$$
- Dopo il primo invio del primo pacchetto, appena il canale di trasmissione tra A e  $S_1$  è libero, viene subito trasmesso il secondo pacchetto; il tempo di immissione verso B di ogni pacchetto è quindi  **$t_I$**  (se si dovesse attendere invece l'arrivo del primo pacchetto a B, avremmo  **$3t_I$** ).  
ogni quanto tempo un pacchetto viene inviato (oppure arriva) verso B =  $t_I$

B) In quale istante tutti i bit del file sono arrivati a B? Ovvero, in quanto tempo viene completato il trasferimento del file?

- Indicato con  **$t_T$**  il tempo totale di trasmissione, questo equivale alla somma di tutti i  **$t_I$** , tenendo conto però che il tempo che si impiega per immettere il **primo pacchetto** è  **$2t_I$** , e il tempo affinché B lo riceva è  **$3t_I$** . Considerando che abbiamo 100 pacchetti da trasmettere, e che dopo il primo tutti gli altri 99 pacchetti verranno ricevuti ad intervalli regolari uguali a  **$t_I$** , possiamo esprimere il valore di  **$t_T$** :

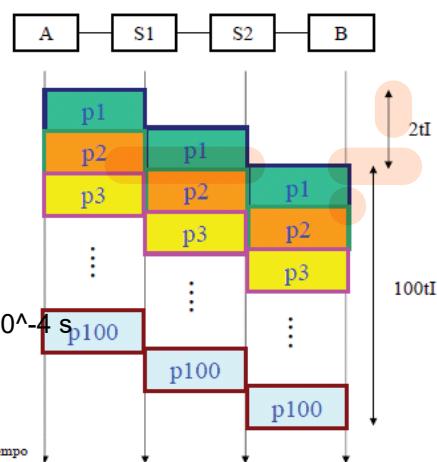
$$t_T = (2 + 100) * t_I = 102 * 10^{-3} [\text{s}] = 0,0102 \text{ s}$$

### tabella delle misure

Ricordiamo che per calcolare il *bit time* di una scheda di rete basta la formula:

$$\text{Bit time} = 1 / \text{velocità della scheda}$$

Nel caso di una rete a 10 Mbps, quindi a  $10 \times 10^6 = 10^7$  Mbps, il bit time sarà  $1 / 10^7 = 10^{-7}$  secondi, ovvero  $100 \times 10^{-9}$  secondi, ovvero 100 nanosecondi.



$$t_I = 0,0001 \text{ sec}$$

$$t_T = 0,0102 \text{ sec}$$

$$t_T = (n+(k-1)) * t_I + k * t_p$$

k fili intendiamo k collegamenti che sono anche numero dei switch

C) Generalizza il calcolo precedente per una rete in cui sono presenti k switch.

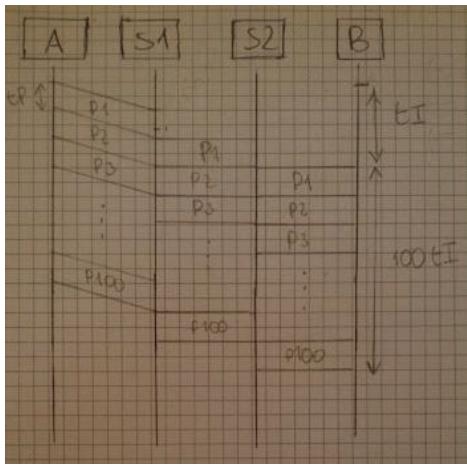
- Possiamo quindi stabilire uno schema generale, considerando **n** pacchetti e **k** switch o **sistemi intermedi** (che non sono altro che un certo numero di "fili" da attraversare). Nell'esempio precedente avevamo 100 pacchetti e 3 fili, e abbiamo trovato che:

$$t_T = (n+2) * t_I$$

Ovvero bastava sommare il numero di pacchetti più il numero di fili meno uno. Per logica induttiva, nel caso in cui avessimo k switch, la formula sarà:

$$t_T = (n+(k-1)) * t_I$$

D) Ripeti l'esercizio, stavolta supponendo che, su una delle linee o su più linee, il ritardo  **$t_P$**  non sia più trascurabile. Fai il disegno corrispondente e, assumendo un valore arbitrario per il ritardo, indica come cambia il tempo di trasmissione in funzione del ritardo  **$t_P$** .



Esempio in cui sulla linea da A a  $S_1$  vi è presente un ritardo di propagazione. Ovviamente il tempo di immissione di un pacchetto aumenta, così come il tempo totale; la formula di per sé però non varia molto e credo basti semplicemente sommare, al tempo totale  $t_I$ , il ritardo su ciascun filo. Indicato quindi  $t_{\text{tot}}$  come somma degli **m** ritardi della rete, la formula diventa:

$$t_T = (n + (k - 1)) * t_I + t_{\text{tot}}$$

E) Come sopra, considera inoltre che le ampiezze di banda di ogni collegamento siano diverse tra loro.



- Nel caso in cui le ampiezze di banda siano diverse, occorrerà calcolare un tempo di immissione individuale per ciascun filo che collega schede di rete a velocità diverse. Supponendo, per esempio, che il primo switch operi a 100 Mbps, il tempo di immissione di un pacchetto verso il secondo switch sarà 10 volte inferiore al tempo di ricezione di un pacchetto dal computer A.



F) Supponiamo infine che, per velocizzare la rete, si acquistino due schede di rete a 100 Mbps. Ci si trova di fronte a tre possibilità: sostituire le schede di A ed  $S_1$ , di  $S_1$  ed  $S_2$  o di  $S_2$  e B. Quale coppia sostituiresti?

???

Significato di banda bassa rispetto alla velocità

Quando si parla di banda bassa, si fa riferimento a collegamenti con una capacità limitata di trasmissione dati, cioè con velocità di trasferimento inferiori rispetto ad altri collegamenti. Una banda bassa significa che il collegamento può trasmettere meno dati al secondo, il che rallenta il trasferimento complessivo delle informazioni

Motivazione: Il tratto con banda più bassa rappresenta il collo di bottiglia della rete (quello che è limitante per gli altri collegamenti e causa di accumulare dei ritardi), e velocizzarlo ridurrebbe maggiormente il tempo di trasmissione totale