Politecnico di Torino
Laurea Triennale in Ingegneria Informatica

appunti di
Reti di calcolatori

Autori principali: Luca Ghio
Docenti: Guido Marchetto, Paolo Giaccone
Anno accademico: 2012/2013
Versione: 1.0.4.1
Data: 21 ottobre 2018

Ringraziamenti

Oltre agli autori precedentemente citati, quest'opera può includere contributi da opere correlate su WikiAppunti e su Wikibooks, perciò grazie anche a tutti gli utenti che hanno apportato contributi agli appunti *Reti di calcolatori*, al libro *Introduzione alle reti telematiche* e al libro *Internet: architettura e protocolli*.

Informazioni su quest'opera

Quest'opera è pubblicata gratuitamente. Puoi scaricare l'ultima versione del documento PDF, insieme al codice sorgente LaTeX, da qui: http://lucaghio.webege.com/redirs/7

Quest'opera non è stata controllata in alcun modo dai professori e quindi potrebbe contenere degli errori. Se ne trovi uno, sei invitato a correggerlo direttamente tu stesso realizzando un commit nel repository Git pubblico o modificando gli appunti *Reti di calcolatori* su WikiAppunti, oppure alternativamente puoi contattare l'autore principale inviando un messaggio di posta elettronica a artghio@tiscali.it.

Licenza

Indice

Parte I

Introduzione alle reti telematiche

Capitolo 1

Nozioni introduttive

comunicazione trasferimento di informazioni secondo un protocollo (= insieme di convenzioni prestabilite tra sorgente e destinatario)

telecomunicazione comunicazione a distanza (cavo, segnali radio, sistemi ottimi, sistemi elettromagnetici...

1.1 Funzioni

Il **servizio** è ciò che viene dato all'utente.

Le **funzioni** vengono gestite dall'operatore:

- **segnalazione di utente:** le informazioni di controllo tra utente e rete (ad esempio, in una telefonata la segnalazione di utente serve per iniziare o terminare di chiamata, inviare il numero destinatario, ecc.);
- **commutazione:** la rete predispone le risorse necessarie alla comunicazione vera e propria;
- **segnalazione di rete:** i segnali di controllo che vengono scambiati internamente tra un nodo e l'altro della rete durante la comunicazione (nel 1800 la segnalazione di rete consisteva nel "dialogo" tra i centralinisti);
- **trasmissione:** la comunicazione vera e propria del segnale;
- alla fine le risorse vengono rilasciate e ridestinate ad altri utenti.

1.2 Topologie

Una **rete di telecomunicazione** è un insieme di nodi e canali che fornisce un collegamento tra due o più punti per permettere la telecomunicazione tra essi:

- il **nodo** è un punto in cui avviene la commutazione;
- il **canale** è il mezzo di trasmissione, unidirezionale (ad es. fibra ottica) o bidirezionale (ad es. cavo elettrico).

Un canale può essere:

- **punto-punto:** collega due soli nodi;
- **multi-punto:** più nodi, uno master e gli altri slave, possono accedere al canale condiviso (non contemporaneamente);

- **broadcast:** più nodi possono accedere senza distinzione al canale condiviso; per raggiungere un singolo utente è necessario fornire anche l'indirizzo del nodo di destinazione, in modo che quando il nodo di destinazione riceve l'informazione può iniziare a elaborarlo.

La **topologia** di una rete di telecomunicazione è la rappresentazione grafica a grafo $G = (V, A)$ dei nodi della rete e delle loro interconnessioni. Ogni topologia è caratterizzata da: affidabilità, complessità di controllo, costo (numero dei canali).

Si distinguono:

- **topologia fisica:** rappresenta i canali corrispondenti ai mezzi di trasmissione fisici, ma ogni canale può essere più complesso di un semplice arco (ad es. centro stella);
- **topologia logica:** i canali di comunicazione sono rappresentati in modo astratto, ma possono essere anche complessi a livello fisico (ad esempio un centro stella che consente solo certi instradamenti).

1.2.1 Topologia a maglia completa

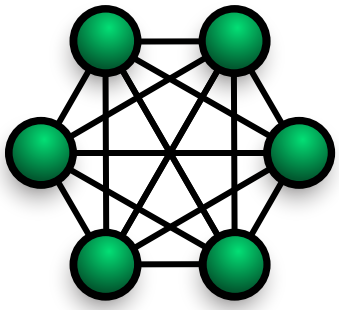


Figura 1.1: Esempio di topologia a maglia completa.[1]

Si definisce **topologia a maglia completa** il grafo completo:[2]

$$C = \frac{N(N-1)}{2} = O\left(N^2\right)$$

Svantaggi

- massimo numero di canali ⇒ costo massimo
- complessità di controllo: è necessario un algoritmo per individuare il percorso di lunghezza minima tra sorgente e destinazione in tempi rapidi (∼decine di clock) ⇒ solo le reti con pochi nodi sono a maglia completa

Vantaggio maggiore tolleranza ai guasti ⇒ maggiore affidabilità: un arco guasto viene facilmente sostituito da un altro

1.2.2 Topologia ad albero

In una **topologia ad albero** tutti i nodi sono connessi tra loro ma, a differenza della **topologia a maglia**, non esistono cicli:

$$C = N - 1 = O\left(N\right)$$

[1] Questa immagine è tratta da Wikimedia Commons (NetworkTopology-FullyConnected.svg), è stata realizzata dall'utente Foobaz e dall'utente BMacZero e si trova nel dominio pubblico.

[2] $C = |A|$ (numero di canali o archi); $N = |V|$ (numero di nodi)

[3] Questa immagine è tratta da Wikimedia Commons (NetworkTopology-Tree.png), è stata realizzata dall'utente Foobaz e si trova nel dominio pubblico.

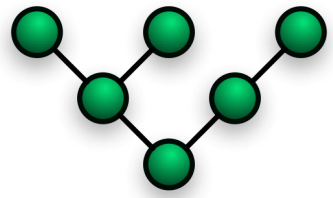


Figura 1.2: Esempio di topologia ad albero.[3]

Vantaggi

- minimo numero di canali ⇒ costo minimo
- complessità di controllo: semplice perché non esistono percorsi alternativi

Svantaggio vulnerabilità ai guasti ⇒ minore affidabilità

1.2.3 Topologia a stella attiva

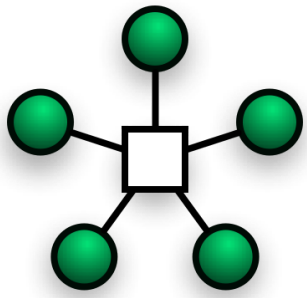


Figura 1.3: Esempio di topologia a stella attiva.[4]

In una **topologia a stella attiva**, la commutazione viene effettuata dal **centro stella** a cui tutti i nodi sono collegati:

$$C = N$$

Vantaggio complessità di controllo: tutte le informazioni da un nodo all'altro passano attraverso il centro stella ⇒ l'intelligenza è demandata al centro stella

Svantaggio vulnerabilità ai guasti del centro stella

Esempi reti locali, reti via satellite, reti radio cellulari

1.2.4 Topologia a stella passiva

In una **topologia a stella passiva**, il canale è unico (con N archi) ⇒ la comunicazione è sempre broadcast:

$$C = 1$$

[4]Questa immagine è derivata da un'immagine su Wikimedia Commons (NetworkTopology-Star.png), realizzata dall'utente Foobaz, e si trova nel dominio pubblico.

[5]Questa immagine è derivata da un'immagine su Wikimedia Commons (NetworkTopology-Star.png), realizzata dall'utente Foobaz, e si trova nel dominio pubblico.

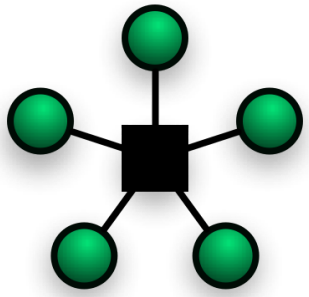


Figura 1.4: Esempio di topologia a stella passiva.[5]

Vantaggio il centro stella è semplice perché non ha intelligenza

Svantaggio vulnerabilità ai guasti del centro stella

1.2.5 Topologia ad anello

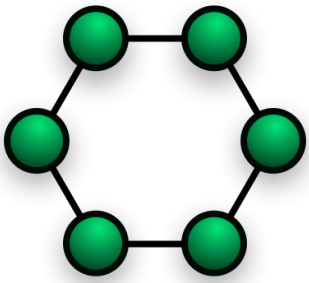


Figura 1.5: Esempio di topologia ad anello.[6]

In una **topologia ad anello**, i nodi sono disposti circolarmente:

- anello unidirezionale: $C = \frac{N}{2}$
- anello bidirezionale: $C = N$

Vantaggio il numero di canali è molto basso ⇒ basso costo

Tolleranza ai guasti

- anello bidirezionale: se avviene un singolo guasto, basta un deviatore che manda l'informazione sull'altra porta nella direzione opposta ⇒ complessità semplice dell'algoritmo di gestione dei guasti;
- anello unidirezionale: alcuni nodi non riescono più a comunicare tra loro.

[6] Questa immagine è tratta da Wikimedia Commons (NetworkTopology-Ring.png), è stata realizzata dall'utente Foobaz e si trova nel dominio pubblico.

1.2.6 Topologia a bus

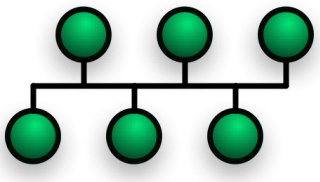


Figura 1.6: Esempio di topologia a bus passivo.[7]

- **topologia a bus attivo:** è un albero in cui su ogni nodo sussistono al massimo due archi;
- **topologia a bus passivo:** è composto da un canale condiviso a cui sono connessi tutti i nodi (ad es. reti locali e metropolitane, prima versione dell'Ethernet).

1.2.7 Prestazioni

Le **prestazioni** di una rete dipendono dal throughput e dalla topologia. Il **throughput** è il traffico corrente, cioè la quantità di dati trasmessi da tutti i nodi in una certa unità di tempo (bit/s). Se il traffico è uniforme[8] e la struttura è regolare, il massimo throughput $x_{\max}$, cioè la massima quantità di traffico smaltibile dalla rete, è inversamente proporzionale alla media della distanza $\overline{D}$ tra ogni coppia di nodi: è possibile trasmettere più informazioni se la distanza media è minore, e viceversa.

Minore è la distanza media, maggiore è il massimo throughput possibile e migliori sono le prestazioni della rete:

- anello: $\overline{D} = O(N) \Rightarrow$ prestazioni peggiori: distanza media molto alta;
- manhattan: $\overline{D} = O\left(\sqrt{N}\right)$;
- shuffle: $\overline{D} = O(\log N)$;
- maglia completa: $\overline{D} = 1 \Rightarrow$ prestazioni migliori: il percorso minimo tra un nodo e l'altro è costituito da un solo arco.

1.3 Servizi

Le reti possono essere:

- **dedicate:** offrono un unico servizio;
- **integrate:** integrano più servizi (ad es.: l'ISDN a banda stretta (N-ISDN) e larga (B-ISDN) supportano sia la telefonia sia la trasmissione dati).

Si distinguono i servizi portanti e i teleservizi.

I **servizi portanti** sono quelli che garantiscono all'utente solo la trasmissione del segnale da un punto all'altro (es. circuito diretto).

[7] Questa immagine è tratta da Wikimedia Commons (NetworkTopology-Bus), è stata realizzata dall'utente Foobaz e si trova nel dominio pubblico.

[8] Il traffico è **uniforme** se ogni nodo trasmette in media lo stesso numero di bit degli altri.

1.3.1 Teleservizi

I **teleservizi** offrono di più rispetto ai servizi portanti:

- **servizio di base:** fornisce funzionalità minime (es. televisione, telefonia);
- **servizio supplementare:** fornisce funzionalità aggiuntive (es. TV on demand).

Il flusso di informazioni può essere:

- **bidirezionale simmetrico:** in entrambe le direzioni e la stessa quantità di dati viene in media trasmessa in entrambe le direzioni (es. telefono);
- **bidirezionale asimmetrico:** in entrambe le direzioni ma in una direzione vengono trasmessi più dati;
- **unidirezionale:** in una sola direzione (es. sensore, radio).

oppure:

- **punto-punto** (es. telefonata);
- **punto-multipunto** (es. radio, broadcasting);
- **multipunto-multipunto** (es. videoconferenza).

I teleservizi si classificano in:

- **diffusivi:** l'utente non può interagire (es. televisione, YouTube)
- **interattivi:** permettono l'interazione tra gli utenti ⇒ devono rispettare certi vincoli di qualità (es. in una videochiamata l'audio deve essere senza scatti)
  - conversazionali
  - messaggistica
  - di consultazione e reperimento

I teleservizi si realizzano in due modalità che si differenziano per le modalità di comportamento degli applicativi utente:

- **modello client-server:** è possibile distinguere due ruoli ben distinti nella rete, cioè il client (es. browser) e il server (es. server che ospita il sito), e il server deve essere sempre attivo in attesa della richiesta del client;
- **modello peer-to-peer:** ciascun utente è sia client sia server per la stessa applicazione: chi usufruisce del servizio lo fornisce anche ⇒ sono paritetici.

1.4 Trasmissione

L'informazione può essere trasmessa in due modi:

- **segnale analogico:** il segnale elettrico è definito come una funzione continua e limitata che può assumere infiniti valori;
- **segnale digitale:** il segnale elettrico è definito come una funzione discontinua e limitata che può assumere un numero finito di possibili valori.

1.4.1 Segnale analogico

Attraverso il campionamento e la quantizzazione, un segnale analogico può essere trasformato in un segnale numerico in modo da facilitare la trasmissione:

- **campionamento:** è preciso se rispetta il teorema di Nyquist-Shannon:[9] $f_c \geq 2B_{\max}$, cioè se viene campionato con frequenza sufficiente, ma la banda non è esattamente definita normalmente;
- **quantizzazione:** l'errore è minore tanti più livelli si fissano.

Esempio 1 La voce del telefono è affetta da numerosi errori ma è sufficiente che l'utente capisca le parole:

- frequenza di campionamento: $f_c = 8$ kHz $= 8000$ campioni al secondo
- (PCM) 256 livelli $\Rightarrow$ 8 bit
- frequenza di dati trasmessi: $8 \times 8 = 64$ kbit/s

Esempio 2 Un CD musicale è più fedele:

- frequenza di campionamento: $f_c = 44$ kHz
- (PCM) 2^{16} livelli $\Rightarrow$ 16 bit
- frequenza di dati trasmessi: 704 kbit/s

1.4.2 Segnale digitale

Dal punto di vista fisico un segnale digitale è comunque un segnale continuo, soggetto a disturbi e interferenze, e il ricevitore tramite un filtro passa-basso deve ricostruire l'informazione discreta. In natura la trasmissione di un segnale non ha mai una probabilità di errore nulla.

Inoltre il ricevitore deve campionare il segnale ricevuto negli istanti di tempo corretti $\Rightarrow$ servono delle informazioni di sincronizzazione:

- **trasmissione asincrona:** il trasmettitore e il ricevitore hanno due clock separati, e un clock può non andare a velocità perfettamente uguale rispetto all'altro $\Rightarrow$ l'informazione stessa porta delle informazioni utili per compensare gli errori di sincronia (ad es. lo Start Bit è il riferimento della durata di un bit e permette di correggere la velocità del clock all'inizio della trasmissione);
- **trasmissione sincrona:** esiste una rete, parallela a quella delle informazioni, che sincronizza i due clock $\Rightarrow$ è necessaria una sincronizzazione non a livello di bit ma a livello di **trama** (ad es. una tastiera deve far capire quando inizia e finisce ogni carattere).

L'informazione può venire trasmessa in due modi:

- **trasmissione parallela:** l'informazione viene trasferita più bit alla volta (per esempio un byte o 2 ma anche molto di più, dipende dal cavo) su un bus di comunicazione;
- **trasmissione seriale:** l'informazione viene trasferita un bit alla volta $\Rightarrow$ è necessario ricostruire il byte.

La ricezione può essere:

- **continua:** punto-punto, trasmissione sincrona;
- **burst mode:** multi-punto o broadcast, trasmissione asincrona.

[9] Si veda il capitolo "Campionamento" negli appunti di *Teoria ed elaborazione dei segnali*.

1.5 Modi di trasferimento

1.5.1 Condivisione di canale

La condivisione dei canali permette di abbassare i costi.

Un canale può essere condiviso in due modi:

- **multiplazione:** tanti flussi di informazioni accedono tutti in un punto preciso della rete (come una centrale telefonica) e condividono un unico canale di uscita ⇒ l'accesso all'unico canale è gestito in modo da evitare le interferenze tra un utente e l'altro (router, ponte radio, satellite, stazione base di rete cellulare);
- **accesso multiplo:** i flussi accedono al canale da punti differenti e arrivano distribuiti nello spazio (ad es. Internet via wireless) ⇒ due utenti che accedono contemporaneamente alla risorsa condivisa si interferiscono a vicenda (reti locali, terminali mobili rete cellulare, stazioni di terra in comunicazioni via satellite).

Internet è basato sulla multiplazione statistica, la telefonia è basata sulla multiplazione predeterminata:

- **multiplazione predeterminata:** l'allocazione delle risorse è determinata a priori (ad es. per tutta la durata di una telefonata i parametri di frequenza, tempo... non cambiano, anche se per un certo tempo chi telefona sta zitto);
- **multiplazione statistica:** l'allocazione delle risorse viene decisa in base ai dati che arrivano sul momento, cioè in base alle variazioni istantanee di traffico (ad es. in Internet ogni dato è in lotta con gli altri dati per l'accesso al canale).

Sia la multiplazione sia l'accesso multiplo prevedono varie modalità di suddivisione dei flussi di informazione: frequenza, tempo, spazio e codice.

Divisione di frequenza (Frequency Division Multiplexing [FDM], Frequency Division Multiple Access [FDMA])

Un canale è suddiviso in vari sottocanali, e ogni utente può accedere a uno dei sottocanali liberi. Ad ogni sottocanale è associata una porzione della banda<footnoteLa banda è intesa come un intervallo di frequenze. complessiva del canale. Le bande di guardia sono dei margini di sicurezza tra un canale e l'altro.

Divisione di tempo (Time Division Multiplexing [TDM], Time Division Multiple Access [TDMA])

La separazione dei canali avviene nel tempo: ogni utente può comunicare solo in certi intervalli di tempo, ripetuti secondo un pattern, ma può usare l'intero spettro di frequenza. I tempi di guardia compensano gli errori di sincronizzazione tra gli utenti.

TDM + FDM Un utente può parlare solo in un certo intervallo di tempo e su una certa banda di frequenze (es. GSM). Il ponte radio deve però comunicare le informazioni dello slot in cui comunicare, e durante la comunicazione potrebbe anche decidere di modificare lo slot concesso.

Divisione di spazio

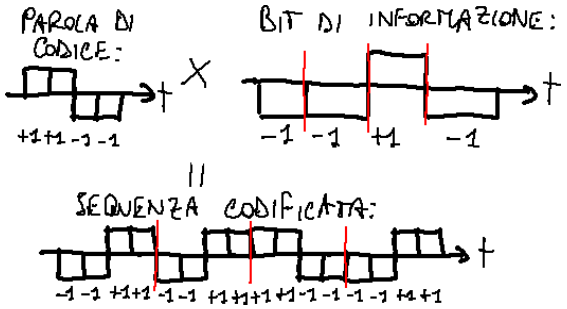
La separazione dei canali avviene nello spazio: due canali possono essere lontani tra loro (ad es. telefonia cellulare: comunicazione telefono-antenna) o fisicamente separati (ad es. due fili) ⇒ nessuna interferenza, neanche se le comunicazioni avvengono alla stessa frequenza e/o nello stesso momento.

Divisione di codice (Code Division Multiplexing [CDM], Code Division Multiple Access [CDMA])

Usata per esempio da UMTS e GPS. A ogni utente è associato un codice differente, e tutti trasmettono contemporaneamente su tutta la banda di frequenze a disposizione e per tutto il tempo della comunicazione; la distinzione è fatta solo in base al codice.

Si ottiene una migliore protezione dalle interferenze, perché le interferenze tipicamente sono associate a una specifica frequenza, mentre l'utente comunica usando tutto lo spettro. I flussi usano segnali digitali tutti ortogonali tra loro.

Il trasmittente i-esimo ottiene la sequenza codificata dai prodotti scalari tra la parola di codice e ogni bit di informazione (tenendo conto che il bit 1 corrisponde a $+1$ e il bit 0 corrisponde a -1):



Siccome nell'esempio per ogni bit bisogna trasmettere 4 bit più veloci è necessaria più banda, ma non è un problema perché la comunicazione avviene sull'intero spettro delle frequenze, che per questo motivo viene detto **spettro allargato**.

In ogni istante tutte le sequenze codificate trasmesse dagli utenti si sommano in potenza.

Il ricevitore i-esimo effettua il prodotto scalare tra il segnale ricevuto e la parola di codice del trasmittente i-esimo:

- se ottiene un numero positivo è stato trasmesso un bit 1;
- se ottiene un numero negativo è stato trasmesso un bit 0;
- se ottiene 0 il trasmittente non ha trasmesso niente.

Il ricevitore è semplice da costruire perché non deve sintonizzarsi su una particolare frequenza.

1.5.2 Condivisione di un nodo

Commutazione di circuito

A ogni richiesta di servizio viene allocata una specifica risorsa che è esclusiva dell'utente anche se non viene sfruttata (ad es. l'utente sta zitto in una telefonata), e viene rilasciata solo al termine della comunicazione.

Il **diagramma spazio-temporale** rappresenta su una linea temporale i **tempi di propagazione** tra due nodi collegati da un canale di comunicazione e disposti su una linea spaziale. Il tempo di propagazione è diverso dal tempo di trasmissione: il tempo di propagazione è regolato solo dalla natura e non si può modificare (in particolare non può superare la velocità della luce c), il tempo di trasmissione si può ridurre ingegneristicamente.

La propagazione del segnale avviene in 3 fasi:

- **impegno:**
  - U1-N1: segnalazione di utente al nodo più vicino;

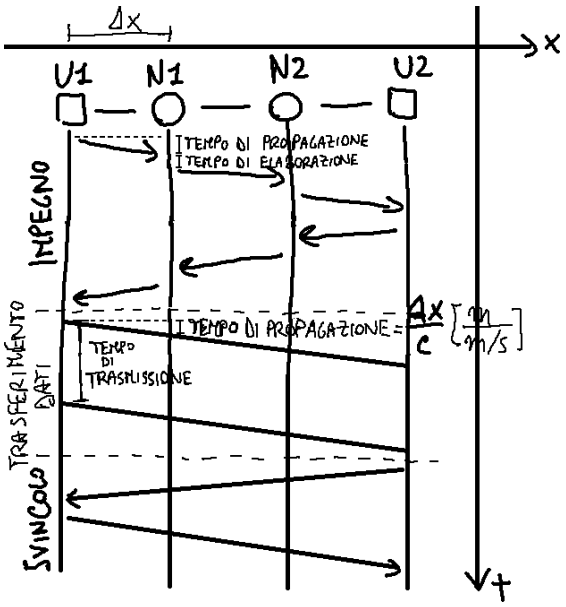


Figura 1.7: Diagramma spazio-temporale di una commutazione di circuito.

  - N1-N2: i nodi iniziano ad allocare le risorse (commutazione) e interagiscono con altri nodi in direzione della destinazione; inizia con un certo ritardo perché il nodo deve elaborare la segnalazione di utente;
  - N2-U2: segnalazione di rete al destinatario;
  - U2-N2-N1-U1: la segnalazione ritorna indietro al mittente;

- **trasferimento dati:** il trasferimento dell'informazione vera e propria passa su ogni nodo senza ritardi, perché le risorse sono già state allocate;
- **svincolo:** serve un'ulteriore segnalazione per il rilascio delle risorse.

Esempio - Rete telefonica

- impegno: inizia chiamata → chiamata in ingresso → accetta chiamata → chiamata accettata;
- trasferimento dati: trasmette dati → riceve dati;
- svincolo.

Vantaggi

- banda costante garantita (ad es. telefonata: c'è un filo[10] riservato solo all'utente);
- ritardi di trasferimento costanti: durante la fase di trasferimento dati, a ogni passaggio attraverso un nodo non c'è alcun ritardo perché il nodo conosce già il prossimo nodo a cui inviare l'informazione ricevuta;
- trasparenza del circuito: una volta dato il circuito, l'utente può cambiare il formato dei dati, la velocità e il protocollo (ad es. telefonata: si può parlare qualunque lingua).

[10] In realtà è uno slot.

Svantaggi

- spreco di risorse: la risorsa è dedicata per tutta la durata della comunicazione (ad es. telefonata: quando l'utente non parla la risorsa non può essere riallocata);
- ha una buona efficienza solo se le sorgenti non sono intermittenti, cioè emettono sempre informazione;
- si paga un tempo di apertura del circuito a ogni richiesta di comunicazione;
- una volta inviato un dato con certi formato, velocità e protocollo, arriva a destinazione senza la possibilità di convertirlo;
- la tariffazione si basa sul tempo della comunicazione (ad es. bolletta telefonica a tempo).

Commutazione di pacchetto

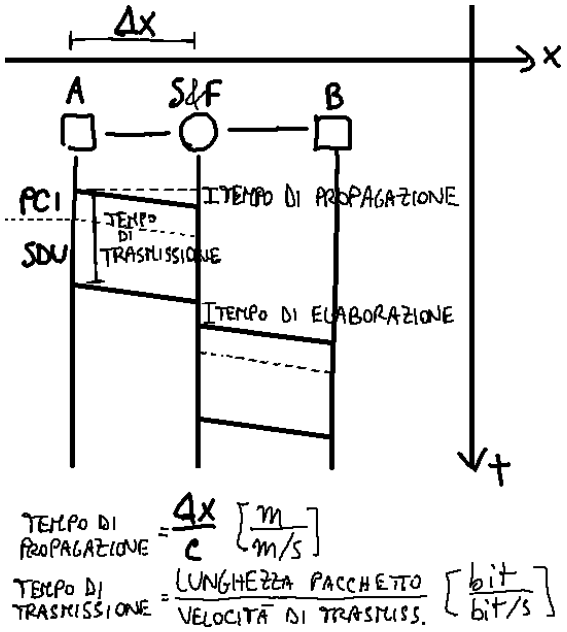


Figura 1.8: Diagramma spazio-temporale di una commutazione di pacchetto.

La risorsa non è esclusiva di un utente ma è condivisa. L'allocazione delle risorse avviene dinamicamente in base ai dati trasmessi in ogni istante ⇒ l'informazione viaggia in modo intermittente a seconda del traffico altrui (ad es. Internet). È stata studiata appositamente per sorgenti intermittenti che non sanno a priori quanti dati saranno inviati.

I dati da trasferire vengono frazionati in unità dati dette **Protocol Data Unit** (PDU), che sono formate da PCI e SDU:

- **Protocol Control Information** (PCI): l'informazione aggiuntiva di controllo (intestazione);
- **Service Data Unit** (SDU): i dati effettivi da trasferire.

Nell'analogia con il servizio postale: la PDU è la busta, la PCI è l'indirizzo, l'SDU è il contenuto.

Le unità dati (PDU) possono essere chiamate in vari modi a seconda del contesto: pacchetto, cella, datagramma, segmento, messaggio, trama.

L'intestazione è molto importante al fine di non perdere il pacchetto ⇒ occorre qualche forma di protezione degli errori sull'intestazione. Per velocizzare il più possibile l'elaborazione

l'intestazione del pacchetto viene inviata prima della SDU. Per garantire la corretta ricostruzione dell'informazione, nell'intestazione è necessario specificare:

- l'indice del pacchetto, perché i pacchetti potrebbero arrivare a destinazione anche non nell'ordine di invio;

e

- un modo per capire se sono stati ricevuti tutti i pacchetti della sequenza:
  - un bit di informazione che informa se il pacchetto è l'ultimo o se esiste un pacchetto successivo (molto vulnerabile a errori di trasmissione);

  oppure

  - il numero totale di pacchetti.

Store and forward Le reti a commutazione di pacchetto si basano sullo **store and forward**:

- store: il nodo (switch o router) memorizza il pacchetto ricevuto;
- processing: il nodo elabora l'intestazione (PCI) per determinare il canale su cui inoltrarlo;
- forward: il nodo mette il pacchetto in coda per la trasmissione sul canale.

Il trasferimento di ogni pacchetto subisce vari ritardi:

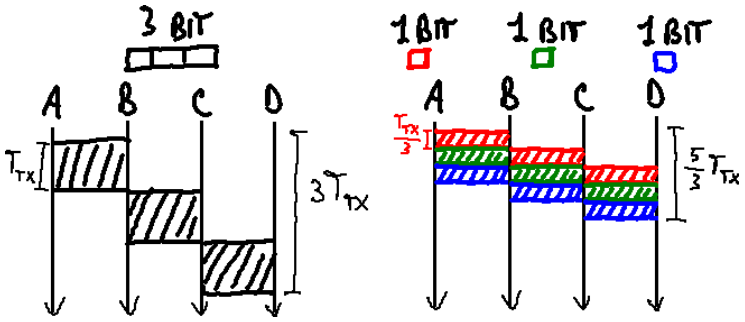
- **ritardo di trasmissione:** minimizzabile dal punto di vista ingegneristico;
- **ritardo di propagazione:** dipende solo dalla velocità della luce;
- **ritardo di elaborazione** in ogni nodo;
- **ritardo di accodamento:** se un pacchetto arriva a un nodo che sta trasmettendo un altro pacchetto, deve aspettare che il nodo finisca di trasmetterlo.

Lunghezza dei pacchetti I pacchetti possono essere:

- **a dimensione fissa** se il numero di bit trasportati non può essere cambiato (es. ATM[11]); se i dati da trasportare effettivamente hanno però una dimensione inferiore, occorre riempire ugualmente il pacchetto aggiungendo alla fine dei bit casuali e specificare nell'impostazione che quei bit aggiunti sono privi di validità;
- **a dimensione variabile** se la dimensione del pacchetto dipende dal numero di dati trasportati (es. Ethernet).

La lunghezza dei pacchetti è determinata da un compromesso tra:

- **possibilità di parallelizzazione** (pipeline): pacchetti brevi favoriscono la trasmissione in parallelo e riducono il tempo complessivo di trasferimento:[12]



[11] Si veda la voce Asynchronous Transfer Mode su Wikipedia in italiano.
[12] Nel disegno sottostante si trascurano i ritardi di propagazione e i ritardi di elaborazione.

- **ritardo di pacchettizzazione:** se la sorgente invia dati non già pacchettizzati ma in modo continuo e a velocità costante (ad es. voce), è necessario attendere un certo tempo necessario affinché il pacchetto si riempa man mano che i dati vengono inviati. Anche il ritardo di pacchettizzazione si riduce accorciando la dimensione dei pacchetti in modo che ogni pacchetto risulti pieno dopo meno tempo;
- **percentuale di informazioni di controllo:** siccome ogni pacchetto deve avere un'intestazione, frazionare l'informazione in un elevato numero di pacchetti significa aumentare i bit destinati alle informazioni di controllo;
- **probabilità di errore:** se n è il numero di bit e p è la probabilità che un bit inverta il suo valore:

  $$P\{\text{almeno 1 errore}\} = 1 - P\{\text{tutti giusti}\} = 1 - (1-p)^n \simeq 1 - (1 - np) = np$$

  da cui la probabilità di errore si riduce col numero n di bit, quindi tanto più quanto i pacchetti sono corti.

Vantaggi e svantaggi

Vantaggi

- è efficiente anche per traffici intermittenti;
- il nodo può verificare se il dato è corretto, e in caso contrario non vengono trasmessi al nodo successivo;
- il nodo nell'elaborazione può convertire il formato e può cambiare la velocità e il protocollo;
- la tariffazione può essere fatta in base al traffico effettivamente trasmesso.[13]

Svantaggi

- *send & pray*: non è garantita la banda perché dipende dal traffico corrente;
- ogni pacchetto deve essere elaborato dal nodo;
- il ritardo di trasferimento è variabile.

Modi di trasferimento Una rete a commutazione di pacchetto può essere:

- **datagram:** non c'è un accordo preliminare con il fornitore e i pacchetti possono seguire percorsi diversi;
- **circuito virtuale** (es. ATM): emula la commutazione di circuito su una rete a commutazione di pacchetto.

A loro volta i circuiti virtuali si distinguono in:

- **permanenti** (PVC): l'utente (es. banca) richiede all'operatore un circuito, l'operatore gli crea una rete semi-statica, e quel circuito viene mantenuto fino a quando l'utente non comunica che la rete può essere dismessa ⇒ maggiore affidabilità;
- **commutati** (SVC): il circuito viene creato in tempo reale: il PC dell'utente quando necessario deve fare una segnalazione di rete al PC destinatario e aspettare risposta.

La comunicazione in una rete a circuito virtuale è suddivisa in 3 fasi, analoghe a quelle della commutazione a circuito:

[13] In realtà la maggior parte degli operatori sceglie comunque di offrire abbonamenti a Internet di tipo flat.

- apertura connessione (segnalazione)
- trasferimento dati
- chiusura connessione (segnalazione)

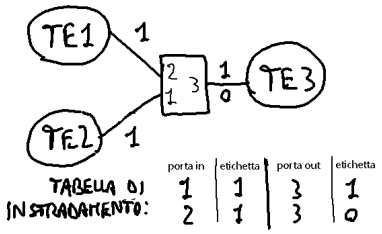
Deve esistere un accordo preliminare con il fornitore del servizio: tutti i pacchetti seguono lo stesso percorso accordato, e si cerca di garantire che arrivino nell'ordine di invio.
La commutazione a pacchetto a circuito virtuale differisce dalla commutazione di circuito perché non si allocano in modo statico delle risorse.

Indirizzamento In una rete di comunicazione, i vari interlocutori devono essere identificati in modo univoco: ad esempio nella rete Internet ogni nodo ha un indirizzo IP, che deve essere univoco a livello mondiale.[14]
Nella rete datagram ogni pacchetto deve contenere oltre all'indirizzo del destinatario anche quello del mittente, perché:

- il destinatario può rispondere inviando a sua volta dati al mittente;
- se il pacchetto ricevuto risulta danneggiato, il destinatario può provare a leggere l'indirizzo del mittente per richiedergli la ritrasmissione del pacchetto.

Invece, nella rete a circuito virtuale l'instradamento avviene solo prima e dopo la fase di trasferimento ⇒ l'intestazione dei pacchetti ha dimensioni minori:

- per aprire e chiudere il circuito serve comunque una coppia di identificatori globali per la sorgente e la destinazione;
- durante il trasferimento è possibile minimizzare l'informazione di controllo: basta un'etichetta che identifica localmente la singola tratta appartenente al circuito virtuale, e le etichette possono essere codificate su un numero inferiore di bit perché poi ogni nodo è in grado di distinguere due etichette uguali basandosi sulla porta d'ingresso:



In una rete ATM l'etichetta, detta ID, è costituita da una coppia di valori, Virtual Path Identifier (VPI) e Virtual Circuit Identifier (VCI), che definiscono un instradamento gerarchico: varie linee con VCI distinti vengono raggruppate in un VPI. Per esempio a una videoconferenza può essere destinato un virtual path suddiviso in due virtual circuit: uno per l'audio, l'altro per il video.

1.6 Tecniche di segnalazione

La segnalazione serve per controllare il canale di comunicazione (apertura e chiusura).
Si distinguono le segnalazioni di utente (tra utente e nodo) e di rete (tra nodo e nodo).

[14] In realtà esistono anche gli indirizzi privati che non sono univoci.

1.6.1 Segnalazione associata al canale

Esiste una corrispondenza biunivoca tra:

- **canale controllante** per le informazioni di segnalazione;
- **canale controllato** per le informazioni di utente.

Si distinguono:

- **segnalazione in banda:** il canale controllante e il canale controllato coincidono, ma vengono usati in tempi diversi (ad es. rete telefonica: i toni per la composizione del numero viaggiano sullo stesso canale vocale);
- **segnalazione fuori banda:** esistono due reti fisicamente distinte, una per il canale controllante e l'altra per il canale controllato.

1.6.2 Segnalazione a canale comune

Un unico canale di segnalazione controlla più canali di informazione utente. Il canale di segnalazione funziona a pacchetto.

1.7 Tecniche di gestione

In una rete l'operatore oltre a dare il servizio vero e proprio deve gestire la rete per quanto riguarda:

- la configurazione (Configuration Management);
- le prestazioni (Performance Management): la velocità della rete va misurata;
- i guasti (Fault Management): serve una rapida manutenzione;
- la sicurezza (Security Management): una banca richiede un certo livello di sicurezza;
- la tariffazione (Accounting Management): l'utente deve pagare solo in base ai bit che effettivamente scambia.

Le Software Defined Network (SDN) permettono di centralizzare la gestione di tutti questi aspetti in un unico server Web basato sullo standard open flow.

1.8 Qualità di servizio

La qualità del servizio è determinata dalla disponibilità di risorse sulla rete e dalle tecniche di allocazione delle risorse (commutazione).

Due approcci

- problema di analisi: data una rete già esistente (traffico generato e risorse a disposizione), determinare la qualità del servizio;
- problema di progetto: dati il traffico previsto e la qualità del servizio richiesta, determinare le risorse necessarie.

È difficile risalire a un modello statistico che rappresenti le richieste di servizio: per esempio, in media una chiamata dura 3 minuti, e gli inizi di chiamata sono descritti in un modello chiamato processo di Poisson.

Le sorgenti analogiche vengono rappresentate con più facilità nel dominio della frequenza.

Le sorgenti numeriche possono essere:

- **a velocità costante** (Constant Bit Rate [CBR]) (es. voce numerizzata);
- **a velocità variabile** (Variable Bit Rate [VBR]) (es. video MPEG).

Le sorgenti a velocità costante (es. telefonata) sono caratterizzate da:

- velocità in bit/s (es. 64 kbit/s);
- dimensione dei dati in bit (es. 1 byte): i bit possono essere mandati all'interno del secondo o tutti insieme o un po' per volta, e la dimensione dei dati è la dimensione del pacchetto;
- durata in s (es. 180 s in media);
- processo di generazione delle chiamate (es. processo di Poisson).

Le sorgenti a velocità variabile sono caratterizzate da:

- velocità di picco in b/s;
- velocità media in b/s;
- durata in s;
- processi di generazione delle chiamate (es. YouTube: la popolarità dei contenuti).

Il **grado di intermittenza** (o burstiness) è il rapporto tra la velocità di picco e la velocità media.

Il modello **token bucket** può essere usato per descrivere una sorgente a velocità variabile e permette di regolare l'immissione del traffico in rete: i token entrano in un bucket a una certa velocità costante, e la sorgente preleva un token dal bucket ogniqualvolta invia un pacchetto.

Le prestazioni di un servizio dipendono da alcuni indici di qualità:

- ritardo;
- velocità;
- probabilità di errore di un bit;
- probabilità di perdita di un pacchetto;
- probabilità di blocco del servizio.

Tabella 1.1: Esempi di indici di qualità.

	Ritardo massimo	**Velocità**	**Probabilità di errore**	**Probabilità di blocco**
Telefonia CBR	qualche decimo di secondo, quasi in tempo reale (la voce è molto sensibile ai ritardi)	al massimo 64 kbit/s	inferiore a qualche % (la voce è poco sensibile ai disturbi di sottofondo)	bassa
Posta elettronica (VBR)	fino a diversi minuti	bassa	trascurabile	trascurabile
Video su richiesta	fino a qualche secondo, quasi in tempo reale	decine di Mbit/s	inferiore a qualche %	molto bassa

Capitolo 2

OSI

Il **modello Open System Interconnection** (OSI) è il modello di riferimento per le reti di telecomunicazione a pacchetto. È recepito nei seguenti standard: ISO IS 7498 e CCITT X.200. È organizzato gerarchicamente in livelli:

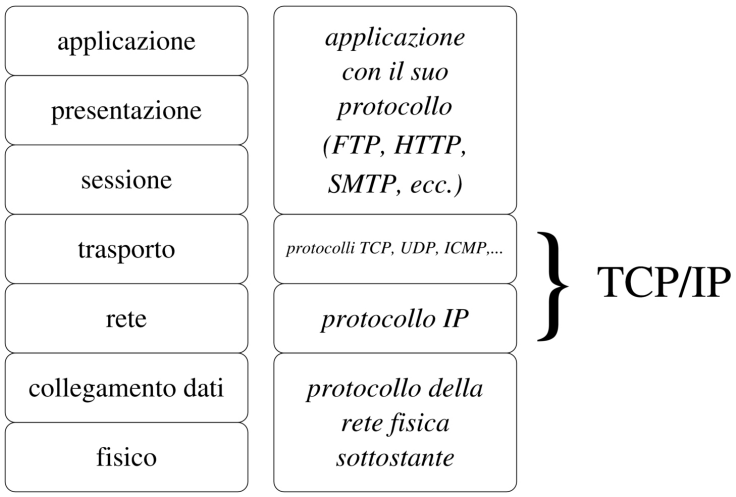


Figura 2.1: Implementazione del modello OSI nell'Internet Protocol Suite.[1]

Una rete si può definire rete Internet se a livello rete vi è implementato il protocollo IP. In una rete Internet i 3 livelli superiori sono lasciati agli applicativi sul PC dell'utente.

Le reti B-ISDN (es. ATM) hanno delle architetture molto più complesse ma consentono una maggiore affidabilità.

2.1 Nomenclatura

In un **sistema** gli **strati** sono organizzati in modo gerarchico: ogni strato fornisce servizi allo strato superiore, usando i servizi dello strato inferiore (detto **black-box**) e le proprie determinate funzioni.

Le **entità** sono gli elementi attivi di un **sottosistema** e interagiscono all'interno di uno strato. Ad esempio, le entità che si trovano al livello più alto sono gli applicativi in esecuzione sul PC dell'utente.

[1] Questa immagine è tratta da Wikimedia Commons (Abbinamento ISO-OSI e TCP-IP.jpg), è stata realizzata da Daniele Giacomini ed è concessa sotto la licenza Creative Commons Attribuzione - Condividi allo stesso modo 2.5 Generico.

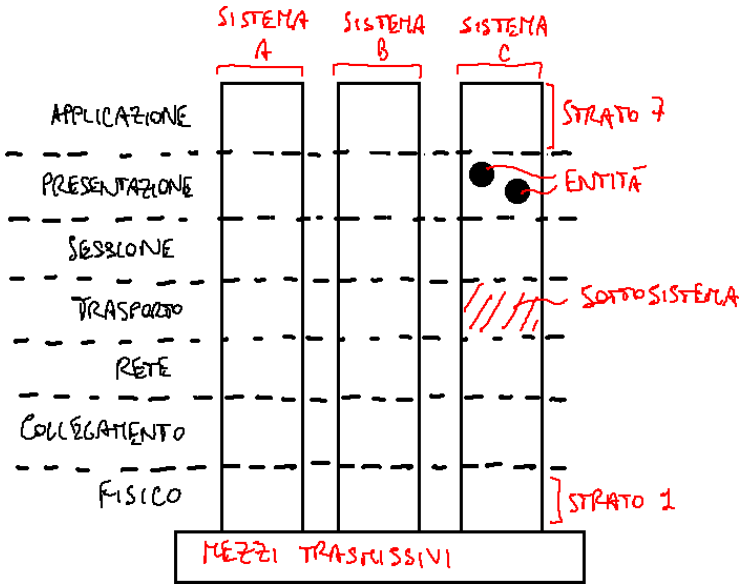


Figura 2.2: Nomenclatura.

2.2 Strati

I sistemi si distinguono in:

- **sistemi terminali:** implementano tutti i livelli;
- **sistemi di rilegamento** (relay): implementano sono alcuni dei livelli più bassi (es. switch Ethernet).

[non chiaro]

Nelle reti pubbliche gli strati di utilizzazione, che corrispondono all'utente, accedono alla sezione di accesso della rete tramite protocolli di accesso, e i nodi della rete interna comunicano tra di loro attraverso protocolli di rete.

2.2.1 Livello 1: fisico[3]

Il **livello fisico** (physical) è lo strato deputato al trasporto fisico dei bit di informazione tra un sistema e l'altro. Un livello fisico è definito in base a codifiche di linea, connettori, livelli di tensione, ecc.

2.2.2 Livello 2: collegamento[4]

Il **livello di collegamento** (data link) permette il trasferimento di unità dati del livello rete e cerca di fronteggiare i malfunzionamenti dello strato fisico.

Funzioni

- rivelare e recuperare gli errori di trasmissione;
- controllare il flusso, cioè evitare di inondare la destinazione se non è pronta a ricevere dati;
- delimitare le unità dati, cioè individuare l'inizio e la fine di una PDU, di una PCI, e così via.

[2]Questa immagine è tratta da Wikimedia Commons (OSI Model v1.svg), è stata realizzata da Dino Korah e dall'utente Offnfopt ed è concessa sotto la licenza Creative Commons CC0 1.0 Universal.

[3]Per approfondire, si veda il capitolo 4.

[4]Per approfondire, si veda il capitolo 5.

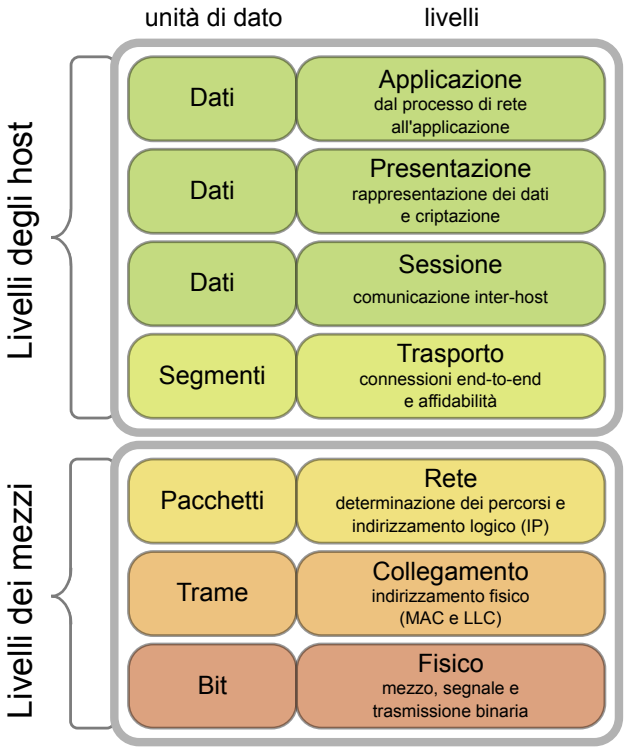


Figura 2.3: Modello OSI.[2]

2.2.3 Livello 3: rete[5]

Il **livello di rete** (network) controlla:

- l'instradamento, cioè trova il percorso all'interno della rete;
- il flusso, cioè evita di inondare la destinazione;
- la congestione, cioè evita di intasare la rete;
- la tariffazione.

Siccome l'Internet Protocol Suite non segue rigorosamente lo standard, i controlli di flusso, congestione e tariffazione non vengono svolti nel livello di rete (IP) ma nel livello di trasporto.

2.2.4 Livello 4: trasporto[6]

Il **livello di trasporto** (transport):

- effettua controlli di errore e di sequenza: cerca di garantire una certa qualità di servizio riparando gli errori che si verificano nello strato di rete;
- controlla il flusso, cioè evita di inondare la destinazione;
- gestisce la multiplazione, la suddivisione, la segmentazione e la concatenazione dei pacchetti.

Nell'Internet Protocol Suite il livello di trasporto controlla anche la congestione, ed è il primo livello ad essere implementato solo nel PC dell'utente, mentre il livello di rete è ancora presente in tutti gli apparati di commutazione.

[5] Per approfondire, si veda il capitolo 9.
[6] Per approfondire, si veda il capitolo 14.

2.2.5 Livello 5: sessione

Il **livello di sessione** (session) gestisce lo scambio di dati a livello temporale. Ad esempio, in una applicazione che riproduce una sorgente video il livello di sessione è quello che ricostruisce l'ordine corretto dei fotogrammi e gestisce la funzione di pausa. Su Internet questo livello è interamente all'interno dell'applicazione stessa.

2.2.6 Livello 6: presentazione

Il **livello di presentazione** (presentation) controlla come i dati vengono rappresentati, risolve eventuali problemi di compatibilità e può fornire anche servizi di cifratura dei dati.

2.2.7 Livello 7: applicazione[7]

Il **livello di applicazione** (application) è lo strato che interagisce con l'utente.

2.3 Comunicazione

2.3.1 Protocollo

Il **protocollo** è un insieme di convenzioni prestabilite. Attraverso un protocollo un'entità può comunicare con un'altra entità che si trova sullo stesso livello gerarchico, anche in sistemi diversi.

Definire un protocollo significa definire:

- algoritmi: la semantica del protocollo (i significati delle parole);
- formati: la sintassi del protocollo (la costruzione delle parole);
- temporizzazione: le sequenze temporali.

2.3.2 Service Access Point (SAP)

Attraverso un **Service Access Point** (SAP) un'entità può comunicare con un'altra entità di livello inferiore o superiore.

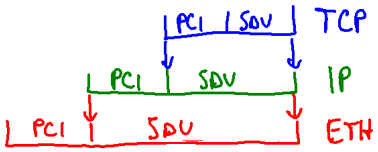
Ad ogni entità è associato un **titolo**. Ad ogni SAP è associato un **indirizzo**.

La connessione tra entità può essere:

- **multiplata:** più entità di livello superiore condividono un'unica entità di livello inferiore;
- **suddivisa:** un'entità di livello superiore è collegata a più entità di livello inferiore.

2.3.3 Creazione PDU

La PDU di livello superiore diventa la SDU di livello inferiore, e ogni livello aggiunge la sua PCI:



Ogni volta che un'entità crea una PDU usando come SDU la PDU di livello superiore cambia il protocollo da usare per leggere la PDU.

A ogni passaggio le dimensioni della PDU aumentano sempre di più. Possono esistere dei vincoli relativi alle dimensioni dei pacchetti:

[7] Per approfondire, si veda il capitolo 15.

- **segmentazione:** il pacchetto dello strato superiore viene separato in più parti, e ogni parte viene inserita in una diversa SDU;
- **concatenazione:** più pacchetti dello strato superiore vengono caricati tutti su una sola SDU.

Il processo continua fino a quando il pacchetto arriva ai mezzi trasmissivi, che lo trasferiscono fisicamente a un altro sistema.

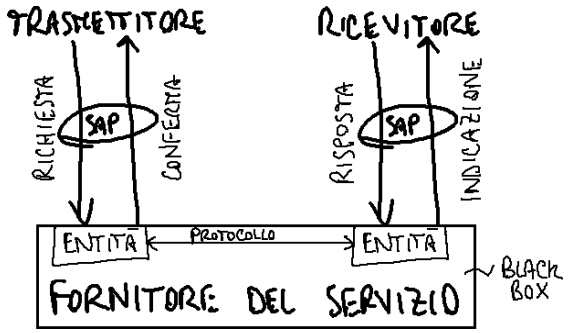
2.4 Servizio

Un servizio può essere:

- **connection-oriented** (CO): prima di trasferire i dati il trasmittore si assicura che il ricevitore sia raggiungibile;
- **connectionless** (CL): send & pray.

Ad esempio, per trasferire informazioni tra un'entità e un'altra entità dello stesso livello ma di un altro sistema è necessario un fornitore del servizio di livello inferiore:

- trasferimento con connessione (CO): esiste un accordo preliminare tra le tre parti entità-fornitore-entità che precede l'invio delle informazioni;
- trasferimento senza connessione (CL): sono necessari accordi tra le singole parti che sono indipendenti tra loro: un accordo tra l'entità sorgente e quella di destinazione, un accordo tra l'entità sorgente e il fornitore del servizio, e un accordo tra il fornitore del servizio e l'entità di destinazione.



Le **primitive** sono le interazioni tra le entità che permettono di offrire un servizio:

- **richiesta:** la primitiva chiamata dall'entità trasmettitore verso l'entità fornitore del servizio;
- **indicazione:** la primitiva chiamata dall'entità fornitore del servizio verso l'entità ricevitore;
- **risposta:** la primitiva chiamata dall'entità ricevitore verso l'entità fornitore del servizio;
- **conferma:** la primitiva chiamata dall'entità fornitore del servizio verso l'entità trasmettitore.

La chiamata di una primitiva genera le chiamate di altre primitive di livello inferiore fino ad arrivare ai mezzi trasmissivi.

Capitolo 3

Protocolli a finestra

I **protocolli a finestra** sono dei protocolli, utilizzati in vari livelli nella gerarchia OSI, che servono per recuperare gli errori:

- **Forward Error Correction** (FEC): si tenta di correggere l'errore;
- **Automatic Retransmission reQuest** (ARQ): viene richiesta la ritrasmissione dei dati.

3.1 FEC

Il FEC è utile per esempio nelle applicazioni real-time dove la ritrasmissione arriverebbe troppo tardi, o nel caso di una sonda lontana dalla Terra dove la ritrasmissione richiederebbe troppo tempo. Il FEC non garantisce però che l'informazione venga ricostruita in maniera corretta.

Esempi di protezione dagli errori

- bit di parità: rileva ma non corregge errori singoli, non rileva due errori;
- codice a ripetizione: i bit vengono mandati più volte e poi viene scelta la sequenza ricevuta il maggior numero di volte ⇒ molta affidabilità ma spreco di banda;
- parità di riga e colonna: rileva e corregge errori singoli.

I codici di protezione dagli errori si inseriscono tipicamente nelle intestazioni dei pacchetti. Può venire protetta la PCI, la SDU o l'intera PDU; è importante proteggere almeno l'intestazione. I codici Cyclic Redundancy Check (CRC) sono dei codici di protezione dagli errori che vengono calcolati in modo molto veloce.

I bit di parità:

- nel FEC vengono usati anche per correggere l'errore;
- nell'ARQ si limitano a rilevare l'errore.

3.2 ARQ

Nell'ARQ oltre ai bit di parità si introducono nella PCI anche dei bit di numerazione che permettono di ricostruire la sequenza di pacchetti.

3.2.1 Stop and wait

1. inizializzazione: il trasmettitore e il ricevitore si sincronizzano per avere lo stesso valore;
2. il trasmettitore, dopo averne fatto una copia, invia la PDU al ricevitore con numero d'ordine pari al suo valore;
3. il ricevitore quando riceve la PDU:
   (a) ne verifica l'integrità attraverso i codici di controllo;
   (b) verifica di aver ricevuto il pacchetto con il corretto numero d'ordine;
   (c) se la PDU è corretta, rispedisce indietro una PCI detta acknowledgment (ACK) per confermare la ricezione e per richiedere il pacchetto con numero d'ordine successivo, incrementando il proprio valore interno;
   (d) se la PDU è quella attesa, inoltra l'informazione all'applicativo utente;
4. il trasmettitore quando riceve la PDU:
   (a) verifica l'integrità della PCI;
   (b) verifica il numero d'ordine;
   (c) invia il pacchetto con numero d'ordine successivo, incrementando il proprio valore interno.

Se il trasmettitore non riceve l'ACK entro un tempo prestabilito, ripete la trasmissione.

Piggybacking Nel caso di flussi di informazione bidirezionali, la PDU di riscontro viene anche sfruttata per l'invio di altri dati.

Criticità

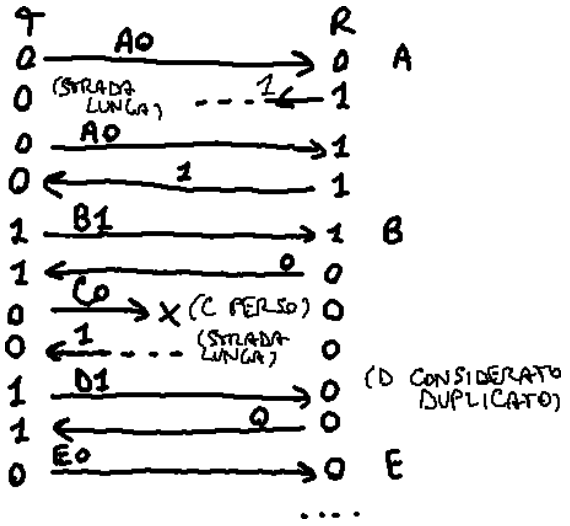
È difficile trovare il giusto valore del timeout, che deve essere idealmente di poco superiore al Round Trip Time (RRT):

- se è troppo breve, il trasmettitore invia molti pacchetti duplicati;
- se è troppo lungo, in caso di pacchetto perso si spreca molto tempo.

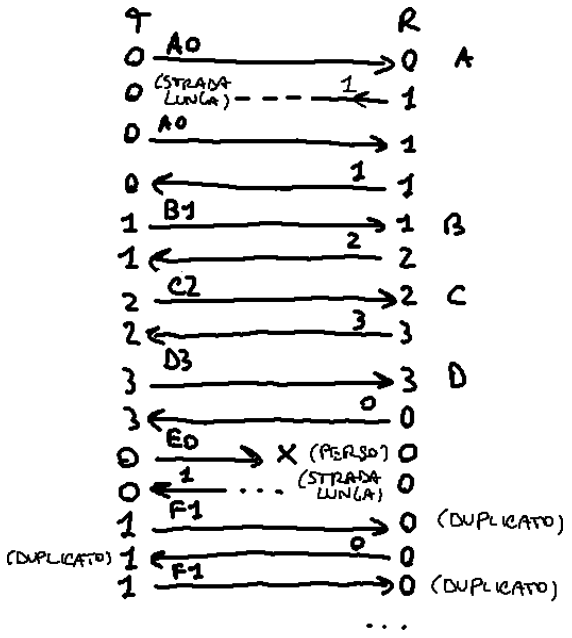
Si riducono le possibilità di malfunzionamento usando:

- un maggior numero di bit per la numerazione;
- un tempo di vita massimo per le PDU e gli ACK, oltre il quale il pacchetto si "suicida".

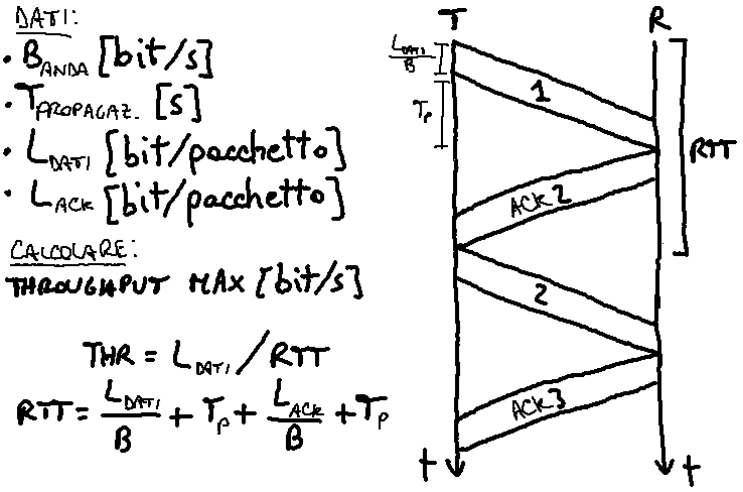
Alternating bit protocol La numerazione delle PDU è indispensabile, perché il trasmettitore deve capire precisamente qual è l'ultimo pacchetto che è arrivato al ricevitore. Se però si trasmettono tanti pacchetti, i numeri d'ordine diventerebbero molto grandi.
L'**alternating bit protocol** cerca di ovviare al problema prevedendo un unico bit per la numerazione, e alterna bit 0 e bit 1.
Su una rete non sequenziale questo protocollo non è affidabile, perché ogni pacchetto può essere instradato in una strada diversa più o meno lunga, e può verificarsi la perdita di pacchetti in determinati casi dovuti ad ACK vaganti:



Numerazione modulo 4 I pacchetti sono nominati da 0 a 3. Anche aumentando il numero di bit in un canale non sequenziale il protocollo potrebbe addirittura entrare in un loop:

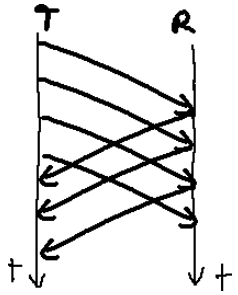


Prestazioni



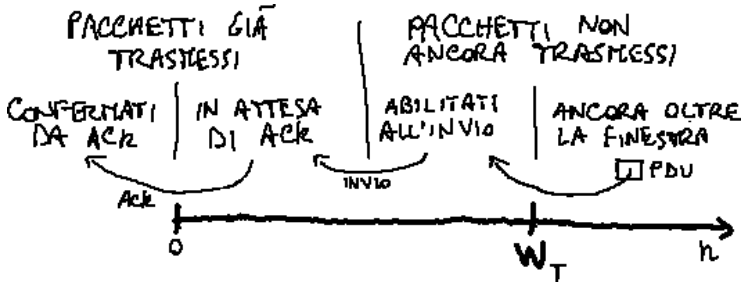
In generale le prestazioni sono limitate a causa dei tempi di attesa degli ACK.

3.2.2 Selective repeat



Il **selective repeat** cerca di ottimizzare le prestazioni dello stop and wait: il trasmettitore è in grado di gestire l'invio consecutivo di pacchetti senza dover aspettare ogni volta l'ACK.

Trasmissione

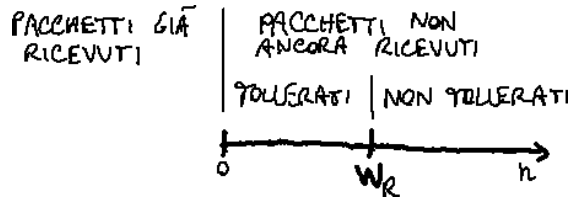


Si definisce **finestra di trasmissione** W_T il numero massimo di PDU che il trasmettitore è autorizzato ad inviare in sequenza senza aver ricevuto riscontro (ACK).

Tutti i pacchetti all'interno della finestra di trasmissione W_T, dopo essere stati memorizzati, vengono inviati consecutivamente dal trasmettitore.

La finestra di trasmissione non può essere maggiore dell'intervallo finito di valori che può assumere il numero d'ordine.

Ricezione



Si definisce **finestra di ricezione** W_R il numero massimo di PDU fuori sequenza che il ricevitore è disposto a tollerare, oltre il quale la PDU viene scartata.

Il ricevitore quando riceve una PDU:

1. ne controlla la correttezza e il numero di sequenza;
2. se la PDU non è corretta o non rientra nella finestra di ricezione la ignora, altrimenti:
   (a) invia la conferma di ricezione (ACK);
   (b) se la PDU è la prima della sequenza la consegna ai livelli superiori, altrimenti attende le precedenti.

Il ricevitore accetta tutti i pacchetti della finestra di ricezione, anche se arrivano completamente fuori sequenza. Tutti i pacchetti ricevuti al di fuori della finestra di ricezione vengono scartati. Ad esempio, se la finestra di ricezione è pari a 2, e l'ultimo pacchetto ricevuto è il pacchetto n-esimo:

- se arriva per primo il pacchetto $n+1$-esimo, tutto ok e la finestra di ricezione trasla;
- se arriva per primo il pacchetto $n+2$-esimo, il ricevitore lo accetta e rimane in attesa del pacchetto $n+1$-esimo;
- se arriva per primo il pacchetto $n+3$-esimo (o successivo), il ricevitore lo scarta.

ACK

Il ricevitore, quando riesce a completare una certa sequenza, invia un ACK che informa dell'ultimo pacchetto ricevuto in sequenza, quindi invia la sequenza ai livelli superiori.

Esistono 3 tipi di ACK:

- **ACK individuale** (o selettivo): ACK(n) significa "ho ricevuto il pacchetto n":
  svantaggio: per ogni pacchetto ricevuto si deve mandare un ACK $\Rightarrow$ molto traffico nella rete;
- **ACK cumulativo:** ACK(n) significa "ho ricevuto tutto fino a n escluso":
  svantaggio: se viene perso solo uno dei primi pacchetti della finestra, il trasmettitore ritrasmette oltre al pacchetto perso anche gli altri pacchetti successivi nella sequenza che in realtà sono già stati ricevuti;
- **ACK negativo** (NAK): NAK(n) significa "ritrasmetti il pacchetto n":
  vantaggio: è utile quando la finestra è molto ampia, o la probabilità di perdita di un pacchetto non è troppo alta.

Trasmettitore e ricevitore si devono accordare preventivamente sulla semantica degli ACK.

Il timeout è unico per la finestra di trasmissione: se scade il timeout prima dell'arrivo di tutte conferme, il trasmettitore ripete la trasmissione delle PDU non ancora confermate.

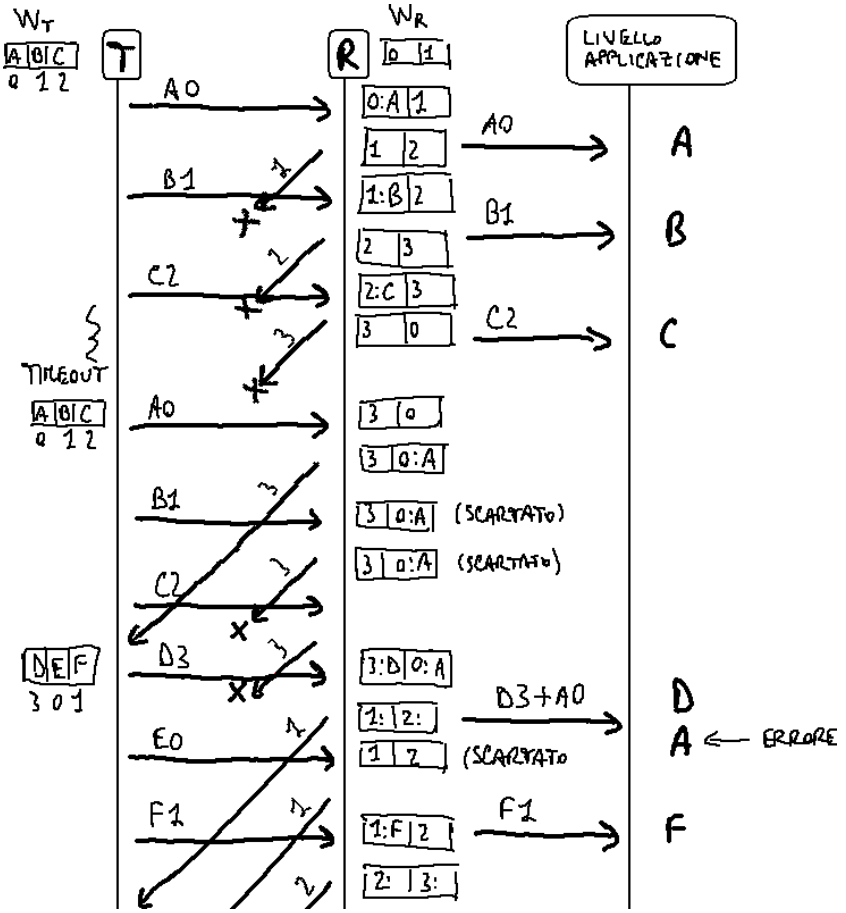
Il trasmettitore si accorge della perdita di pacchetti con la ricezione di ACK duplicati. Quando il pacchetto i-esimo va perso o è in ritardo, il ricevitore continua ad avvisare che l'ultimo pacchetto ricevuto è stato il pacchetto $i-1$-esimo. Quando il trasmettitore riceve due volte lo stesso ACK,

si accorge che c'è stato o un ritardo o una perdita del pacchetto. Se ritrasmettesse subito il pacchetto, ciò sarebbe svantaggioso nel caso in cui il pacchetto sia solamente in ritardo e arrivi subito dopo perché aumenta inutilmente il traffico in rete ⇒ conviene ritrasmettere il pacchetto solo dopo un certo numero, maggiore di 2, di ACK duplicati (il TCP per esempio ne prevede 3). Il selective repeat è efficace perché quando finalmente arriva il pacchetto i-esimo, il ricevitore può mandare subito un ACK e comunicare al trasmettitore il pacchetto fino a quale è arrivato a ricevere tra quelli successivi al pacchetto i-esimo: il trasmettitore così non deve ritrasmettere tutti i pacchetti successivi dall'$i+1$-esimo in poi.

La numerazione delle PDU è ciclica: dati k bit, giunti all'ultimo numero rappresentabile (2^k) si ritorna allo 0. Per evitare ambiguità:

$$W_\mathrm{T} + W_\mathrm{R} \leq 2^k$$

Se questa condizione non viene rispettata potrebbero verificarsi degli errori. Ad esempio, se $W_T = 3$, $W_R = 2$, $k = 2$:

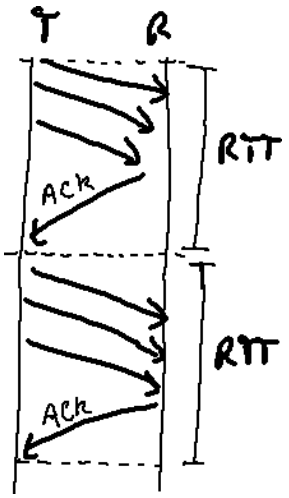


Prestazioni

In assenza di pacchetti persi, il throughput si calcola come:

$$\min\left\{\frac{\text{finestra trasmissione (bit)}}{\text{RTT (s)}}, \text{velocità di linea}\right\}$$

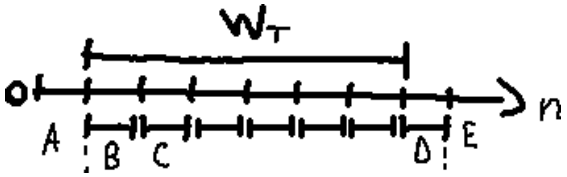
Pertanto accorciando la distanza tra il ricevitore e il trasmettitore, ovvero riducendo il RTT, si possono ottenere prestazioni migliori, anche se aumenta il throughput, cioè il traffico sulla rete.



3.2.3 Go back N

Nel selective repeat, la finestra di trasmissione e la finestra di ricezione sono entrambe maggiori di 1 e di solito di pari dimensione. Invece, nel **go back N** la finestra di ricezione W_R ha sempre dimensione pari a 1 $\Rightarrow$ il ricevitore può ricevere solamente pacchetti in sequenza, perché tutti i pacchetti fuori sequenza vengono scartati.

Fissata una finestra di trasmissione W_T:



A. configurazione vietata: la finestra di trasmissione non trasla finché tutti i pacchetti sono stati ricevuti;

B. nessuno dei pacchetti trasmessi è stato ricevuto;

C. è stato ricevuto il pacchetto 1, ma la sua ACK è andata persa;

D. tutti i pacchetti sono stati ricevuti, ma tutte le ACK sono andate perse;

E. configurazione vietata: il ricevitore aspetta un pacchetto che il trasmettitore non è abilitato a trasmettere.

Rispetto allo stop and wait, nel go back N il trasmettitore diventa più complesso perché sono necessari degli algoritmi per la gestione dei pacchetti. Al lato ricevitore:

- ACK individuali: la gestione è molto semplice: se un pacchetto viene perso, è solo il trasmettitore a dover accorgersi di non avere ricevuto il suo ACK;
- ACK cumulativi: siccome l'ACK viene mandato dopo la ricezione non di un singolo pacchetto ma di un gruppo di pacchetti, è necessario un clock che stabilisca il timeout quando l'attesa di uno dei pacchetti diventa troppo lunga.

Il go back N si comporta come il selective repeat in termini di velocità di trasmissione (throughput) e di occupazione del canale.

Capitolo 4

Livello 1 OSI

Il **livello fisico** (physical) è lo strato deputato al trasporto fisico dei bit di informazione tra un sistema e l'altro. Un livello fisico è definito in base a codifiche di linea, connettori, livelli di tensione, ecc.

4.1 Mezzi di trasmissione

4.1.1 Mezzi elettrici (cavi di rame)

Il mezzo ottimale è caratterizzato da:

- buone caratteristiche elettriche, cioè resistenza, capacità parassite e impedenza basse;
- buona resistenza alla trazione;
- flessibilità (le fibre ottiche non sono flessibili).

Le caratteristiche dei mezzi elettrici dipendono da:

- geometria;
- numero di conduttori e distanza reciproca;
- tipo di isolante;
- tipo di schermatura.

Parametri di merito

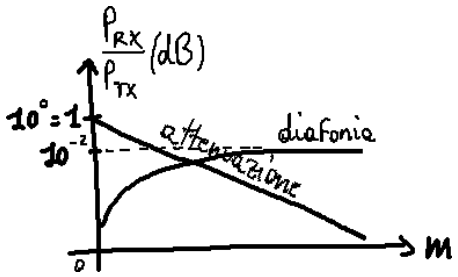
- impedenza;
- velocità di propagazione del segnale: 0,5c - 0,7c per cavi di rame, 0,6c per fibre ottiche;
- **attenuazione**, cioè la perdita di potenza del segnale trasmesso con l'aumentare della distanza: cresce linearmente in dB con la distanza;
- **diafonia** (o cross-talk), cioè la misura del disturbo indotto da un cavo vicino: vicino alla sorgente di trasmissione è poco disturbato, ma cresce con la distanza fino a stabilizzarsi.

dB

$$(x)_{\text{dB}} = 10 \log_{10} \left(\frac{x}{x_{\text{rif}}} \right)$$

$$(P)_{\text{dBm}} = 10 \log_{10} \left(\frac{P \text{ [W]}}{1 \text{ mW}} \right)$$

$$(P_{\text{RX}})_{\text{dBm}} = (P_{\text{TX}})_{\text{dBm}} - (\text{perdita})_{\text{dB}}$$



Doppino

Il **doppino** è costituito da due fili di rame intrecciati per ridurre l'effetto delle interferenze dall'esterno. Ha un costo molto basso.

La versione UTP del doppino non presenta schermature. Vi sono 7 categorie, di qualità e prezzo crescenti:

1. telefonia analogica
2. telefonia ISDN
3. reti locali fino a 10 Mb/s
4. reti locali fino a 16 Mb/s
5. reti locali fino a 100 Mb/s: il connettore si attacca a 4 doppini

5e. reti locali fino a 1 Gb/s: il connettore si attacca a 4 doppini

6. reti locali fino a 1 Gb/s: in fase di standardizzazione

Cavo coassiale

In un **cavo coassiale**, il filo centrale è circondato da una maglia che fa da massa e da un isolante per ridurre le interferenze dei disturbi esterni (gabbia di Faraday). Grazie al maggiore spessore del conduttore l'impedenza è molto bassa, ma non è molto flessibile. Consente velocità di trasmissione dell'ordine di centinaia di Mb/s, ma è costoso.

4.1.2 Mezzi ottici (fibre ottiche)

La **fibra ottica** è costituita da un filo molto sottile di vetro, con dei rivestimenti di protezione.

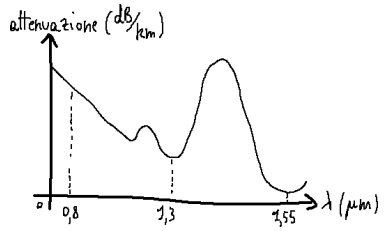
Vantaggi

- totale immunità dai disturbi elettromagnetici, anche se la probabilità di errore non è perfettamente nulla;
- alta capacità di trasmissione (fino a decine di Terabit/s);
- il segnale trasmesso viaggia "rimbalzando" sulle pareti e viene ricevuto con un'attenuazione molto bassa $\Rightarrow$ i cavi possono essere molto lunghi;
- dimensioni ridotte e costi contenuti.

Svantaggi

- bassa flessibilità: si spezza facilmente;
- difficili da collegare tra loro (giunzioni) e con connettori: serve il microscopio $\Rightarrow$ costo elevato;
- la comunicazione è unidirezionale, e le connessioni possono essere solo punto-punto.

Finestre di lavoro



Ci sono tre tipi di laser:

1. lunghezza d'onda 0,8 µm: costo molto basso ma attenuazione alta;
2. lunghezza d'onda 1,3 µm: si trova subito prima della frequenza a perdita massima (1,4 µm);
3. lunghezza d'onda 1,55 µm: perdita minima, ma costo massimo.

4.1.3 Mezzi radio (onde radio)

Una **trasmissione via radio** può subire delle perdite in ampiezza dovute a:

- presenza di ostacoli nell'ambiente:
  - fading (variazione veloce): il segnale si riflette sugli oggetti dell'ambiente che si muovono dinamicamente, e si creano delle interferenze costruttive o distruttive (perché in controfase);
  - shadowing (variazione lenta): la gran parte del segnale viene bloccata da un oggetto vicino al trasmettitore;
- interferenze co-canale: interferenze con altri segnali;
- attenuazione: a differenza dei cavi elettrici, il segnale si attenua con il quadrato della distanza.

4.2 Reti di trasporto

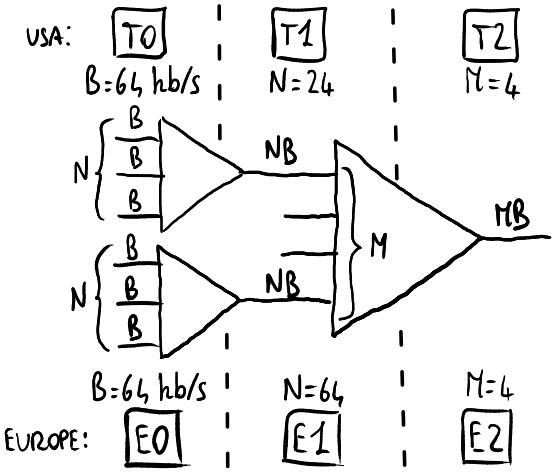
La **rete di trasporto** comprende gli apparati e i mezzi di trasmissione, appartenenti a uno o più gestori, che collegano due nodi di accesso.

4.2.1 PDH

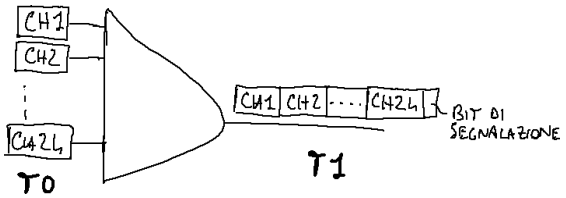
Il **Plesiochronous Digital Hierarchy** (PDH) è un vecchio standard progettato per il trasferimento di canali vocali numerici a 64 Kb/s (PCM) nelle reti telefoniche. Le reti PDH sono reti a circuito digitale basate sul TDM:[1] i dati non viaggiano con lo store and forward, ma in flussi organizzati secondo una trama temporale. Il sistema è detto "plesiocrono" perché occorre una stretta sincronizzazione tra trasmettitore e ricevitore. Esistono diversi standard nel mondo.

Vi è una **gerarchia TDM** se i flussi di dati vengono aggregati in flussi di livello via via superiore secondo delle regole stabilite; più si sale di livello gerarchico, più la velocità di trasferimento è elevata:

[1] Si veda la sezione 1.5.1.



Il PCM ha una frequenza di campionamento di 8 kHz, cioè 8000 campioni al secondo, e ogni campione è pari a 8 bit ⇒ al livello T0 arriva un campione ogni 1 / 8000 = 125 µs. La durata delle trame nei livelli successivi deve continuare a essere uguale a 125 µs perché il flusso di dati dev'essere continuo. Siccome vengono aggregati 24 canali di livello T0 più un bit di segnalazione, al livello T1 in un secondo vengono aggregati 8000 × (8 × 24 + 1) = 1,544 Mbit/s.



Per isolare una singola telefonata all'interno di un flusso bisogna demultiplare un livello per volta, utilizzando alcuni bit di segnalazione per la sincronizzazione (ogni apparato ha un proprio clock).

4.2.2 SONET/SDH

Nel **Synchronous Digital Hierarchy** (SDH) esiste un clock unico per l'intero sistema, grazie a una rete di sincronizzazione o tramite il GPS ⇒ ogni telefonata si trova in una definita posizione di bit ed è così facilmente rintracciabile. L'SDH è l'equivalente europeo dello standard internazionale SONET. La topologia è spesso ad anello per garantire l'affidabilità: un blocco nella rete deve essere recuperato istantaneamente, anche se a costo maggiore.

L'SDH funziona solo su fibra ottica perché le velocità sono incompatibili con i cavi di rame.

La multiplazione dei flussi non è banale come nel PDH (un canale a fianco dell'altro), ma è progettata per ottimizzare l'elaborazione via hardware. Ogni trama temporale include nella PCI:

- informazioni di sincronizzazione per distinguere l'inizio della trama;
- canali vocali di servizio;
- gestione guasti/errori.

Anche nell'SDH ogni trama deve durare 125 µs in tutti i livelli.

4.3 Reti di accesso

La **rete di accesso** (o local loop) comprende gli apparati e i mezzi di trasmissione che collegano l'utente con il nodo di accesso del gestore (es. centrale telefonica urbana).

4.3.1 Rete cellulare

L'area geografica è suddivisa in celle, e al centro di ogni cella c'è un'antenna. Le antenne sono collegate attraverso la rete dell'operatore. A differenza di una comune rete wireless, la **rete cellulare** supporta la mobilità: quando il telefonino si sposta da una cella all'altra cambia ponte radio in modo trasparente.

4.3.2 Plain Old Telephone Service (POTS)

Il **modem** è un dispositivo modulatore e demodulatore, che trasforma i bit in segnali acustici da trasmettere sulla rete telefonica analogica pubblica. Si distinguono il PC dell'utente (Data Terminal Equipment [DTE]) e il modem (Data Circuit-terminating Equipment [DCE]).

L'ultimo standard, il V.90, raggiunge (teoricamente) i 56 kb/s in ricezione e 33,6 kb/s in trasmissione. La velocità è limitata fisicamente dalla capacità di Shannon del doppino:

$$C_{\text{Shannon}} = B_{\text{anda}} \log (\text{rapporto segnale rumore (SNR)})$$

dove il segnale si attenua con la lunghezza del doppino, e la banda è limitata dalla massima frequenza di suoni possibile (4 kHz).

4.3.3 Integrated Services Digital Network (ISDN)

La **rete ISDN** è:

- **digitale:** sulla rete viaggiano i segnali digitali fino al terminale dell'utente ⇒ il telefono deve integrare un convertitore analogico/digitale;
- **integrata:** supporta la trasmissione di dati e voce su un'unica risorsa di rete.

Caratteristiche

- orientata alla connessione (tariffazione a tempo);
- pubblica e/o privata;
- numerica end-to-end: l'informazione viaggia tutta a livello numerico dalla sorgente alla destinazione;
- plesiocrona: usa trame TDM;
- offre servizi a circuito (telefonia, fax) e a pacchetto (trasmissione dati).

È organizzata in due tipi di flussi:

- canale B (Bearer): 64 kb/s (voce, dati, fax);
- canale D (Data): 16 kb/s (segnalazione, dati, telecontrollo);

combinati tra loro:

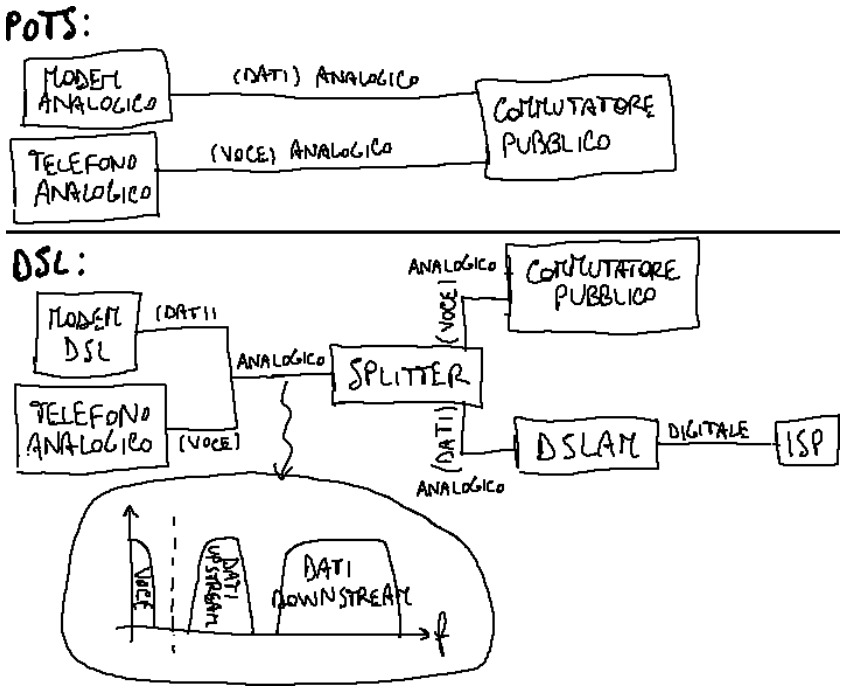
- Basic Rate Interface (BRI) (2B + D, 128 kb/s): destinata all'utenza domestica, il segnale numerico è distribuito tramite l'S-bus che ha una topologia a bus;
- Primary Rate Interface (PRI) (EU: 30B + D, USA: 23B + D): destinata alle imprese.

Gli apparati ISDN sono più costosi rispetto a quelli per l'ADSL ⇒ è una tecnologia abbandonata.

4.3.4 Digital Subscriber Line (DSL)

La rete DSL è una rete digitale che permette di trasmettere dei dati ad alta velocità sulla rete di accesso tradizionale. L'ADSL è una DSL asimmetrica: la velocità in upstream (max 640 Kbps) è molto inferiore a quella in downstream (max 9 Mbps).

Il **filtro splitter** ha il compito di separare il segnale vocale dai dati in base alla frequenza.



4.3.5 Hybrid Fiber Coax (HFC)

Le **reti HFC** sono le reti per la TV via cavo (CATV), diffuse negli Stati Uniti. Sono pensate per un funzionamento unidirezionale: l'operatore invia i dati ai nodi remoti via fibra ottica, e ogni nodo tramite un cavo coassiale manda il segnale in parallelo a tutti gli utenti (topologia ad albero); l'utente usa un cable modem che riceve il segnale analogico e lo converte in digitale. I dati e i segnali TV occupano porzioni diverse di banda ⇒ occorre un filtro presso l'utente. Siccome il canale è unidirezionale, l'upstream di dati viene effettuato su linea telefonica standard.

Confronto tra ADSL e HFC

- l'ADSL è punto-punto, l'HFC è un unico mezzo fisico condiviso tra tanti;
- l'ADSL si appoggia alla tecnologia telefonica standard presso l'utente, l'HFC richiede la posa di cavi ad hoc;
- nell'ADSL la qualità del segnale diminuisce con la distanza, l'HFC non risente della distanza.

4.3.6 Accesso radio mobile

Wireless GPRS, UMTS, IEEE 802.11 (wi-fi), IEEE 802.16 (Wi-Max)

Reti satellitari

- Geostationary Earch Orbit (GEO) (altitudine 35000 km, tempo di propagazione 270 ms, 3 satelliti necessari per la copertura globale): TV satellitare, non per upstream;
- Medium Earth Orbit (MEO) (altitudine 15000 km, tempo di propagazione 50 ms, >10 satelliti): GPS;
- Low Earth Orbit (LEO) (altitudine <1000 km, tempo di propagazione 5 ms, >50 satelliti): telefonia satellitare con antenne omnidirezionali e bassa latenza;

- piattaforme stratosferiche (in fase di studio): un drone vola all'altezza della stratosfera (sopra le nuvole), e funge da satellite a bassissima quota.

Capitolo 5

Livello 2 OSI

Il **livello di collegamento** (data link) permette il trasferimento di unità dati del livello rete e cerca di fronteggiare i malfunzionamenti dello strato fisico.

Funzioni

- delimitare le trame, cioè capire dove si trovano l'inizio e la fine di una PDU, in diversi modi:
  - vi sono bit che delimitano esplicitamente la PDU o ne indicano la lunghezza;
  - i pacchetti sono a lunghezza fissa;
  - la fine di un pacchetto e l'inizio di un altro corrisponde a un silenzio nella trasmissione;
- multiplare i flussi provenienti dagli strati superiori in un mezzo fisico;
- risolvere i problemi di indirizzamento;
- rilevare gli errori;
- utilizzare i protocolli a finestra[1] per:
  - fare il controllo di flusso, cioè evitare di inondare la destinazione se non è pronta;
  - controllare i numeri di sequenza;
  - correggere gli errori;
- regolare l'accesso multiplo nei canali condivisi;
- fare il controllo di flusso sull'interfaccia verso i livelli superiori.

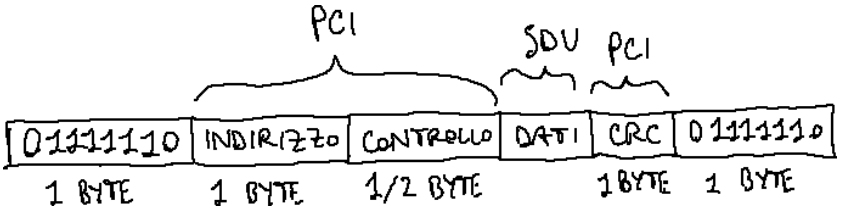
5.1 HDLC

L'**High-level Data Link Control** (HDLC) è lo standard ISO del protocollo Syncronous Data Link Control (SDLC), da cui derivano:

- LAP-B (sezione 5.2);
- LLC per reti locali (sezione 5.3);
- PPP per collegamenti punto-punto su linea commutata (sezione 5.4);
- LAP-F per reti Frame Relay (sezione 5.5).

[1] Si veda il capitolo 3.

Le PDU hanno il seguente formato:



5.1.1 Campi di delimitazione

01111110 è la sequenza che delimina l'inizio e la fine di una PDU, ma è la combinazione vietata nei dati. Nasce il problema se l'utente vuole trasferire proprio la sequenza vietata.

Si può implementare a livello hardware una regola di tipo **bit-stuffing**: se viene ricevuta una sequenza di 5 bit a 1:

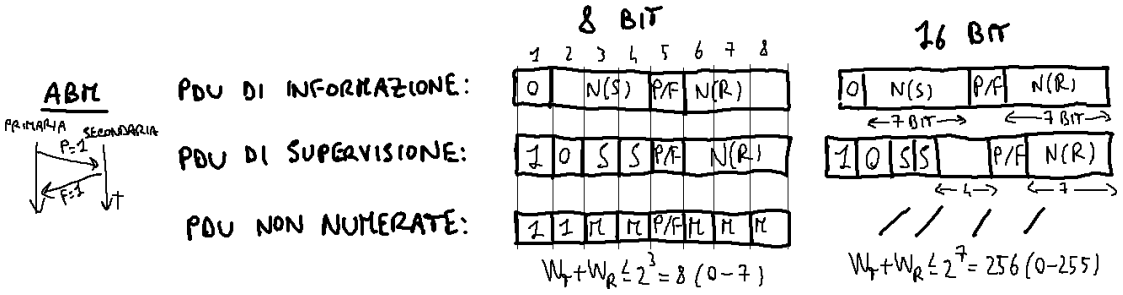
- se il sesto bit è un 1: la PDU è terminata;
- se il sesto bit è uno 0: il sesto bit viene ignorato e la lettura continua al settimo bit.

In questo modo se l'utente vuole inviare la sequenza 01111110, la PDU conterrà la sequenza 011111*0*10.

5.1.2 Campo CRC

Il codice CRC segue i dati perché così l'apparato di ricezione può iniziare a calcolare il CRC dei dati appena sono stati ricevuti mentre aspetta di ricevere anche il CRC di controllo.

5.1.3 Campo di controllo



Il campo di controllo differenzia le PDU:

- N(S): numero di sequenza;
- N(R): numero di riscontro;
- P/F: bit di poll (se inviato dalla stazione primaria) o bit di final (se inviato dalla stazione secondaria). L'invio di un poll specifica che il mittente attende obbligatoriamente una risposta; il final è la risposta a un poll. Il mittente non può emettere un altro P = 1 prima di aver ricevuto un F = 1, perché altrimenti non saprebbe a quale richiesta risponderebbe il final.

5.2 Link Access Procedure Balanced-B (LAP-B)

5.2.1 Campo di controllo

Trame di informazione (I)

Le **trame di informazione** permettono di trasferire i dati.

- N(S) = numero di sequenza della PDU trasmessa;
- N(R) = numero di sequenza della PDU attesa (ACK).

È quindi possibile il piggybacking:[2] il trasmettitore può mandare insieme ai dati anche l'ACK relativo al flusso di dati in direzione inversa.

Trame di supervisione (S)

Le **trame di supervisione** permettono di trasferire ACK senza mandare dei dati. I due bit SS possono comunicare tre informazioni di stato:

- RR (ricevitore pronto): fornisce riscontro positivo;
- RNR (ricevitore non pronto): fornisce riscontro positivo, ma dichiara che il ricevitore non è disponibile (controllo di flusso);
- REJ (reject): richiede di ritrasmettere tutte le PDU a partire da N(R).

Trame non numerate (U)

Le **trame non numerate** permettono il controllo della trasmissione, con comandi e risposte sui 5 bit M:

- Set Asynchronous Balanced Mode(E) (SABM(E)): (re)inizializza il collegamento, specificando anche se la numerazione delle PDU è su 3 o 7 bit;
- Disconnect (DISC): annuncia la chiusura del collegamento;
- Unnumbered Acknowledgment (UA): ACK alla PDU di tipo SABM o DISC.

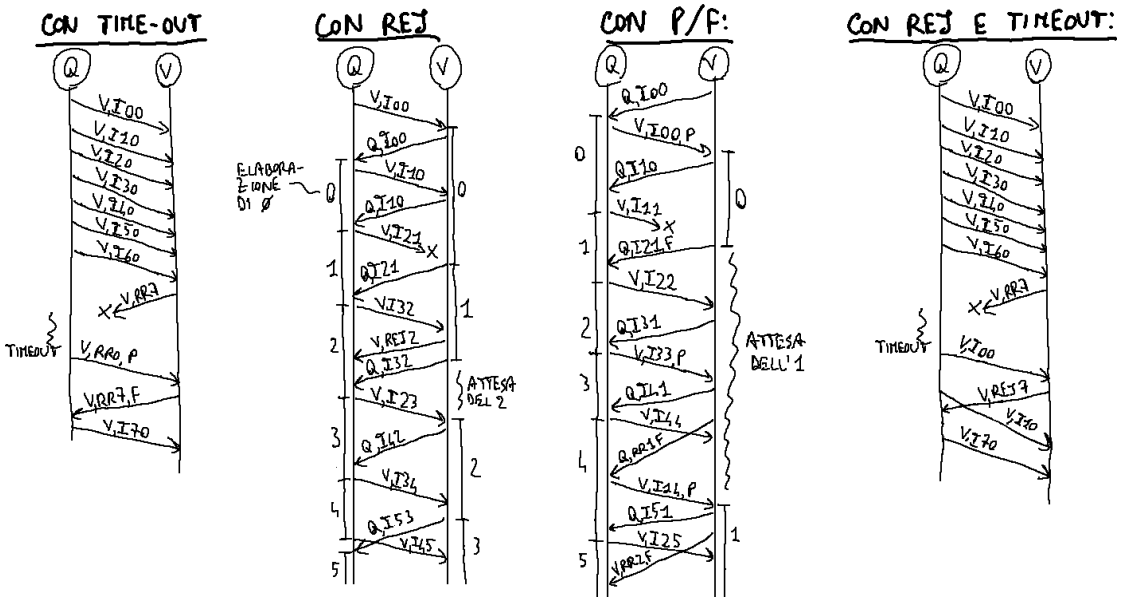
5.2.2 Campo di indirizzo

L'indirizzo non sempre è quello del destinatario: nel caso di una risposta viene specificato l'indirizzo del mittente.

5.2.3 Tecniche di recupero degli errori

Nel LAP-B sono implementabili varie tecniche di recupero degli errori:

[2] Si veda la sezione 3.2.1.

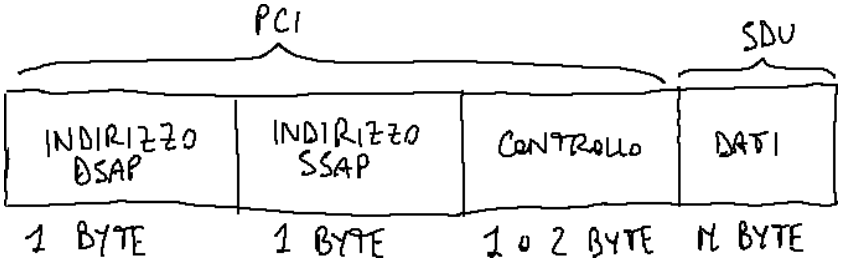


5.3 LLC

Nelle reti locali il livello 2 è diviso in due sottolivelli: **Logical Link Control** (LLC) e **Medium Access Control** (MAC).

IEEE 802.2 è lo standard ISO dell'LLC.

Il formato delle PDU LLC è:

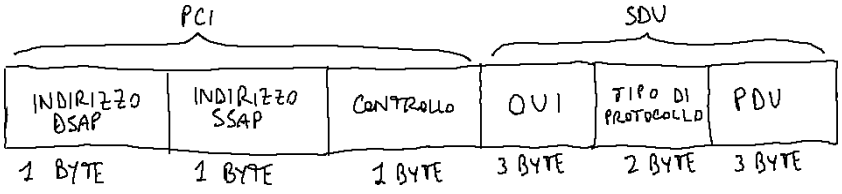


Mancano i campi di delimitazione e di CRC perché sono demandati al MAC. Il campo di controllo è di 1 byte nelle PDU non numerate, e di 2 nelle PDU numerate. Il campo dati ha una dimensione multipla del byte. Il campo di indirizzo contiene gli indirizzi delle SAP[3] sorgente e di destinazione.

Deve essere specificato anche quale protocollo usa il livello superiore.

5.3.1 SNAP

Il **SubNetwork Access Protocol** (SNAP) è una particolare implementazione dell'LLC. Il formato delle PDU è:



[3]Si veda la sezione 2.3.2.

Nel campo dati, 5 byte vengono usati per definire il protocollo di livello superiore usato dalla PDU contenuta. In particolare, il campo OUI è assegnato in modo univoco a un ente o società, che può definire i propri tipi di protocollo con i due byte a fianco.

5.4 PPP

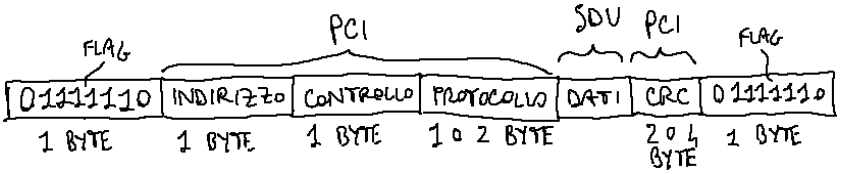
Il **Point to Point Protocol** (PPP) è utilizzato per collegamenti punto-punto cablati:

- collegamenti su linea telefonica tra il modem dell'utente e il provider Internet;
- connessioni SONET/SDH;
- circuiti ISDN.

Obiettivi

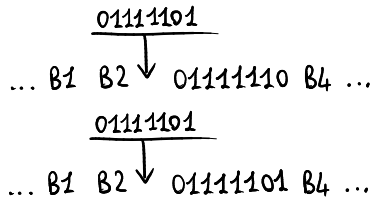
- sono presenti i campi di delimitazione delle PDU (detti flag);
- riconosce gli errori ma non li corregge;
- multipla più protocolli di livello rete;
- controlla l'attività del collegamento;
- permette la negoziazione dell'indirizzo di livello rete (tipicamente IP): i nodi ai due estremi del collegamento apprendono o configurano i propri indirizzi di rete.

Non si occupa né di controllo di flusso né di mantenimento della sequenza.
Il formato delle PDU è:



I campi di indirizzo e di controllo non hanno significato perché in una comunicazione punto-punto non serve l'indirizzo. Il campo di protocollo ha funzione analoga a quello dell'LLC SNAP.

Lo stuffing è a livello di byte: esiste un byte di escape (01111101) che interviene sia quando nei dati compare un byte uguale al byte di delimitazione (01111110), sia quando compare un byte uguale al byte di escape stesso:



Il protocollo PPP-Link Control Protocol (PPP-LCP) ha il compito di aprire e chiudere una connessione PPP, negoziando alcune opzioni (lunghezza massima dei frame, protocollo di autenticazione, ecc.).

5.5 LAP-F

Le **reti Frame Relay** sono reti a pacchetto con servizio di circuito virtuale (tipicamente permanenti):

- velocità: tra 64 kb/s e 2 Mb/s;
- dimensione pacchetti: variabile, max 4 KiB.

Le reti Frame Relay utilizzano il protocollo Link Access Procedure to Frame mode bearer services (LAP-F).

Il formato delle PDU è uguale a quello dell'HDLC, ma i campi si dividono in due categorie:

- DL-CORE (flag + indirizzo + CRC): utilizzato da tutti i nodi della rete;
- DL-CONTROL (controllo): utilizzato solo dal mittente e dal destinatario.

Il campo di indirizzo è composto da alcuni sottocampi, tra cui:

- DLCI: identifica il circuito virtuale;
- FECN e BECN: avvisano esplicitamente la sorgente e la destinazione che un nodo della rete è congestionato (il TCP/IP si accorge implicitamente di congestioni tenendo sotto controllo le perdite di pacchetti);
- DE: specifica la priorità del pacchetto;
- C/R: distingue se il pacchetto è un comando o una risposta.

5.6 ATM

Le **reti Asynchronous Transfer Mode** (ATM) sono reti B-ISDN[4] a pacchetto con servizio di circuito virtuale. La dimensione dei pacchetti è fissa: 53 byte, di cui 48 byte per i dati e 5 byte per l'intestazione.

Svantaggi

- la dimensione dei pacchetti è fissa → per trasportare un singolo byte di informazione servono altri 52 byte;
- la dimensione dei pacchetti è piccola → molta banda è usata per le intestazioni (quasi il 10%).

Vantaggi

- velocità elevate (min. 622 Mb/s);
- la regolarità del formato ATM velocizza il processo di elaborazione dei pacchetti e semplifica l'hardware;
- bassa latenza e basso ritardo di pacchettizzazione ⇒ adatto per il trasporto di voce (telefonia) e video.

Le reti ATM hanno un modello molto complesso, derivato da una mentalità "da operatore telefonico" per avere il controllo su tutta la rete e garantire un'alta affidabilità ai guasti.

[4] Si veda la sezione 4.3.3.

5.6.1 Strato di adattamento ad ATM (AAL)

Quando ATM era stato progettato, si pensava a PC dotati direttamente di schede ATM. Oggi in realtà i PC hanno implementato solo il protocollo IP, e l'ATM è usato solo in alcune parti della rete che richiedono una certa affidabilità, ma servono dei router che convertano in maniera trasparente i pacchetti ATM in pacchetti IP e viceversa.

Le reti ATM hanno un approccio core and edge: al di sopra dello strato ATM (core) vi è uno strato di adattamento (edge), detto ATM Adaptation Layer (AAL), che è presente solo nel terminale di sorgente e in quello di destinazione, ed è appunto usato per suddividere i pacchetti IP in pacchetti ATM.

Il livello AAL è in grado di gestire:

- gli errori di trasmissione;
- la pacchettizzazione;
- la perdita di pacchetti;
- il controllo di flusso.

Classi di servizio

Sono definite 4 classi di servizio AAL in base a 3 parametri legati alla qualità del servizio:

	Classe A	**Classe B**	**Classe C**	**Classe D**
Modalità di temporizzazione fra sorgente e destinazione	necessaria[a]		non necessaria[b]	
Velocità di trasmissione della sorgente	costante (CBR)	variabile (VBR)		
Modalità di connessione	orientato alla connessione[c]			non connesso[d]
Tipi di PDU AAL	AAL tipo 1	AAL tipo 2	AAL tipo 3/4 - 5	
Possibili applicazioni	voce 64 kbit/s video CBR	video/audio VBR	dati	

[a] Nelle applicazioni real-time (ad es. voce) ci sono dei vincoli temporali, cioè i pacchetti non devono arrivare con eccessivo ritardo.

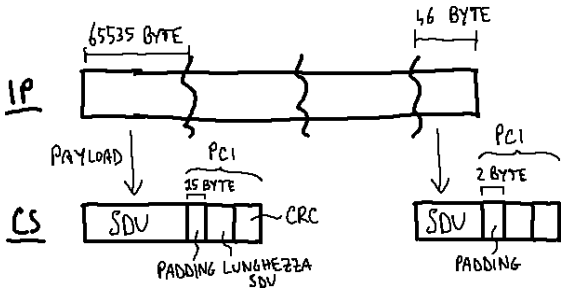
[b] Non vi sono particolari vincoli temporali, basta che i pacchetti arrivino.

[c] Sono sfruttate tutte le caratteristiche di un circuito virtuale.

[d] Il circuito virtuale non viene sfruttato a livello superiore.

Il livello AAL è ancora suddiviso in due sottolivelli: CS e SAR. Di seguito è trattato il processo che porta a una PDU ATM di tipo 5.

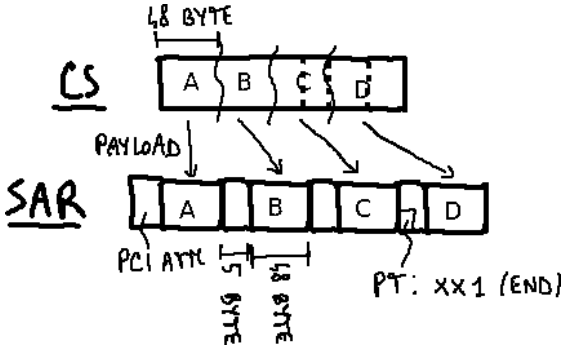
PDU CS



Il pacchetto di livello 3 (tipicamente IP) viene tagliato in parti di dimensione massima 65535 byte, che diventano il payload delle PDU Convergence Sublayer (CS). Il padding sono dei bit di

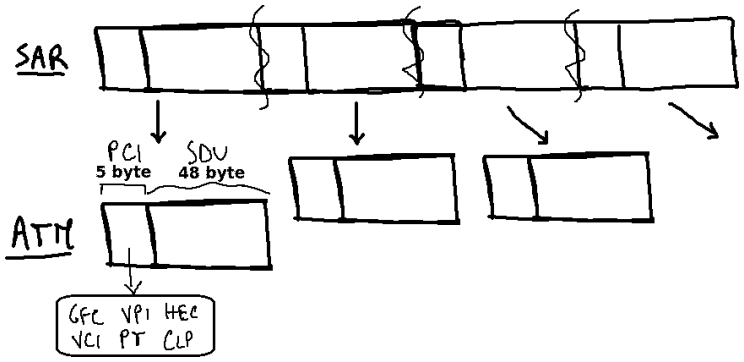
riempimento per adattare il payload CS ad un multiplo di 48, e quindi può andare da 0 a 47 byte. Siccome la lunghezza del payload è variabile, un campo di 2 byte ne definisce la lunghezza. Alla fine il codice CRC permette di verificare la correttezza dell'intera PDU CS al momento della ricostruzione, e quindi di rilevare la perdita di un pacchetto ATM.

PDU SAR



I blocchi da 48 byte vengono inframezzati dalle PCI ATM. Nell'ultima PCI ATM, il terzo bit del campo PT viene impostato a 1 per segnalare la fine del segmento.

5.6.2 PDU ATM



Il formato della PCI distingue due tipi di PDU ATM:

- User-to-Network Interface (UNI): per la comunicazione tra l'utente e un nodo della rete (VPI da 8 bit);
- Network-to-Network Interface (NNI): per la comunicazione tra due nodi interni alla rete (VPI da 12 bit).

Generic Flow Control (GFC) (4 bit)
Tramite il GFC la rete può informare l'utente quanti dati può ricevere (controllo di flusso).

Virtual Path Identifier (VPI) (8-12 bit)
Il VPI è l'identificativo del virtual path, che è un aggregato di circuiti virtuali.

Virtual Circuit Identifier (VCI) (16 bit)
Il VCI è l'identificativo del circuito virtuale all'interno di un virtual path.

Payload Type (PT) (3 bit)
Il PT identifica il tipo di dati trasportati:

- 4 codici sono riservati alle funzioni d'utente, e possono avvisare esplicitamente della presenza di congestioni per rallentare la sorgente;
- 4 codici sono riservati alle funzioni di rete (gestione delle risorse e manutenzione della rete).

Cell Loss Priority (CLP) (1 bit)
Il CLP assegna al pacchetto una priorità per differenziare il traffico: se è presente una congestione, il nodo inizia a buttare via i pacchetti a priorità bassa (CLP = 1) preservando quelli a priorità alta (CLP = 0). In alcuni casi tipo il traffico vocale, se ogni tanto viene perso qualche pacchetto non importa.

Header Error Code (HEC) (8 bit)
L'HEC permette di capire se l'intestazione è stata ricevuta correttamente, e permette di correggere un singolo errore oppure di rilevare due errori.

Tabella 5.1: Confronto tra protocolli di strato 2.

Protocollo	**Delimitazione pacchetti**	**Multiplazione protocolli di strato 3**	**Rilevazione errori**	**Correzione errori (protocollo a finestra)**
LAP-B	delimitatori	realizzato in strato superiore	sì	sì
LLC	demandato a MAC	sì	opzionale	opzionale
MAC (Ethernet)	silenzi	sì	sì	no
PPP	delimitatori	sì	sì	no
LAP-F	delimitatori	mediante circuiti virtuali	sì (DL-CORE)	opzionale (DL-CONTROL)
ATM	demandato al livello fisico	mediante circuiti virtuali	sì (AAL)	no

Capitolo 6

Reti locali

Una rete si dice **locale** se è diffusa su una piccola estensione geografica. Le prime reti locali sono nate con una topologia a bus, ma oggi si preferisce la topologia a stella.

6.1 Protocolli per reti locali

In una rete locale ad accesso multiplo,[1] il mezzo trasmissivo è condiviso ⇒ può trasmettere solo un nodo alla volta:

- vantaggio: un nodo quando trasmette ha a disposizione tutta la rete ⇒ massima velocità;
- vantaggio: comodo per il traffico broadcast e multicast, dove un nodo comunica con tutti gli altri;
- svantaggio: se un nodo vuole comunicare con un altro nodo specifico, deve inserire l'indirizzo del destinatario.

Parametri

- capacità e traffico smaltito (throughput);
- equità tra gli interlocutori;
- ritardi di accesso, propagazione, consegna;
- numero di stazioni, lunghezza della rete, ecc.

Classificazione

- **ad accesso casuale**;
- **ad accesso ordinato:** si basa sul passaggio di "testimoni" detti token ⇒ non ha avuto successo;
- **a slot con prenotazione:** chi vuole parlare deve aspettare che gli si venga data la risorsa.

[1] Si veda la sezione 1.5.1.

6.1.1 Protocolli ad accesso casuale

In un **protocollo ad accesso casuale**, ogni nodo che vuole trasmettere trasmette quando è necessario, e non c'è un determinismo:

- usa la massima velocità permessa dal canale;
- non si coordina con gli altri nodi.

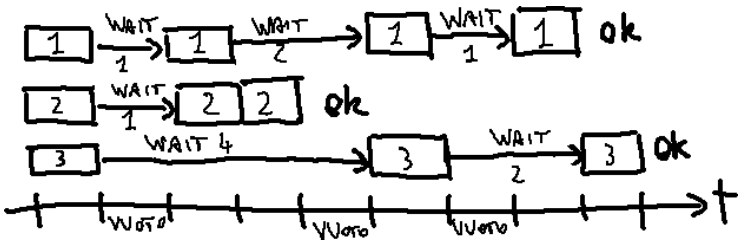
Si ha una **collisione** quando due o più nodi trasmettono contemporaneamente.

Un **dominio di collisione** è un insieme di nodi (schede di rete) che concorrono per accedere allo stesso mezzo trasmissivo ⇒ la trasmissione contemporanea provocherebbe collisione.

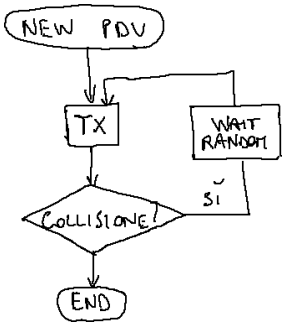
I protocolli ad accesso casuale specificano:

- come riconoscere una collisione;
- come recuperare una collisione (ritrasmissione).

6.2 Slotted ALOHA



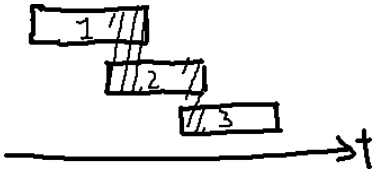
Il tempo è suddiviso in **slot**, e ogni nodo spezza la conversazione in slot. Quando un nodo si accorge che c'è stata una collisione in un certo slot (per esempio a causa della mancanza di un ACK), ritenta la trasmissione in uno slot scelto a caso, finché a forza di tentare la trasmissione non va a buon fine:



È fondamentale che lo slot sia scelto a caso, perché se tutti i collidenti ritrasmettessero aspettando tutti lo stesso numero di slot le collisioni sarebbero senza fine. Il costo di questo coordinamento distribuito è l'inefficienza: ci possono essere degli slot in cui nessuno trasmette.

6.3 Pure ALOHA

Non c'è la sincronizzazione tra gli interlocutori, ovvero non esistono degli slot di tempo ma ognuno può iniziare a comunicare in qualunque istante. Il problema è che aumentano le collisioni: se l'interlocutore A inizia a trasmettere quando l'interlocutore B deve ancora finire, tutto il pacchetto trasmesso da B dovrà essere buttato via.



Svantaggio

- le prestazioni non sono molto alte: sotto ipotesi di traffico uniforme, il throughput massimo è pari al 37% nello Slotted ALOHA, e al 18% nel Pure ALOHA;
- il canale si intasa prima nel Pure ALOHA, perché aumentano molto le collisioni con il carico;
- i ritardi di accesso non sono controllabili a priori in modo deterministico.

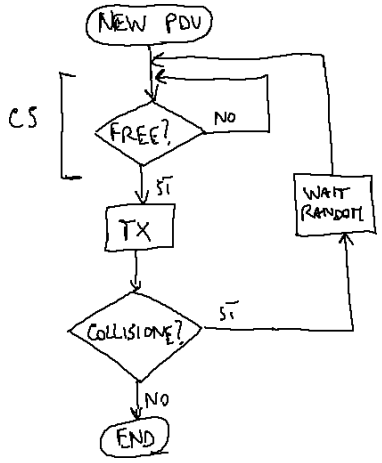
Vantaggi

- protocolli semplici;
- a basso carico, il ritardo di accesso è nullo o contenuto.

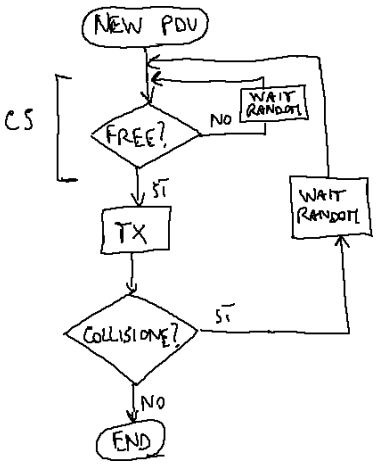
6.4 CSMA

Il **Carrier Sense Multiple Access** (CSMA) prevede l'ascolto del canale (CS) prima della trasmissione:

- se sente che il canale è libero: il nodo trasmette il pacchetto;
- se sente che il canale è occupato:
  - CSMA 1-persistente: il nodo continua a verificare se il canale è libero e trasmette appena si libera:



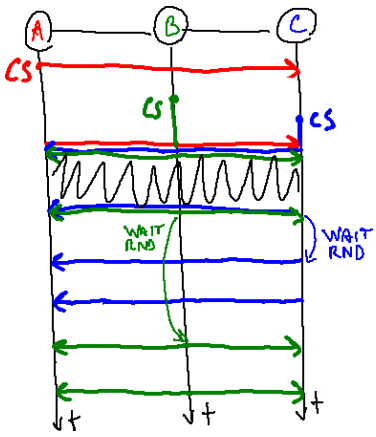
  - CSMA 0-persistente: il nodo riprova dopo un tempo casuale:



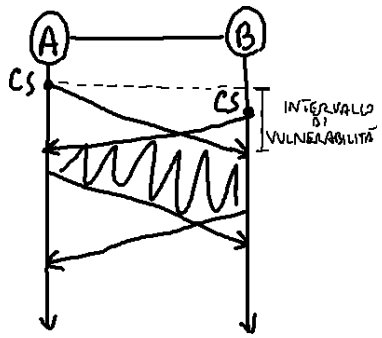
– CSMA p-persistente: il nodo con probabilità $1 - p$ aspetta un tempo casuale (0-persistente), con probabilità p riverifica subito (1-persistente).

Questo protocollo è più efficiente in termini di thoughput. Si possono però verificare ugualmente delle collisioni:

- due interlocutori possono decidere nello stesso istante di iniziare a trasmettere ⇒ si ha una **sincronizzazione delle collisioni**:



- se si tiene conto dei tempi di propagazione, un nodo lontano può sentire il canale libero, anche se in realtà è occupato ma la trasmissione non ha ancora raggiunto il nodo lontano ⇒ si dice **intervallo di vulnerabilità** l'intervallo di tempo in cui l'avvio di una trasmissione da parte del nodo lontano creerebbe una collisione (è pari al ritardo di propagazione sul canale), e questo intervallo è tanto grande quando la distanza è maggiore ⇒ questo protocollo funziona bene su reti piccole:



Pertanto il CS non risolve del tutto il problema delle collisioni.

6.5 CSMA/CD

Nel **Carrier Sense Multiple Access/Collision Detect** (CSMA/CD), anziché trasmettere l'intero pacchetto e soltanto alla fine verificare la collisione, il nodo durante la trasmissione ogni tanto cerca di capire se si è verificata una collisione, e in caso affermativo interrompe subito la trasmissione, evitando di sprecare il canale per una trasmissione inutile.

Nella comunicazione via radio non è possibile implementare la collision detection, perché è difficile per un interlocutore capire se altri stanno parlando in quel momento: in trasmissione non conviene ascoltare il canale, e in ricezione non conviene trasmettere, perché la ricezione sarebbe disturbata dalla trasmissione.

Siccome l'accesso alla rete è conteso, quando si riesce a ottenere l'accesso alla rete conviene trasmettere pacchetti grandi. Esiste un vincolo tra la dimensione minima del pacchetto e la dimensione della rete per riconoscere le collisioni: si veda la sezione 7.2.3.

Ethernet implementa il CSMA/CD 1-persistente perché è pensato per reti scariche.

Il **backoff** è esponenziale nelle ritrasmissioni:

- 1ª ritrasmissione: se c'è una collisione il nodo aspetta un tempo scelto a caso tra 0 e 1;
- 2ª ritrasmissione: se c'è una collisione il nodo aspetta un tempo scelto a caso da 0 a 3;
- 3ª ritrasmissione: se c'è una collisione il nodo aspetta un tempo scelto a caso da 0 a 7;

e così via.

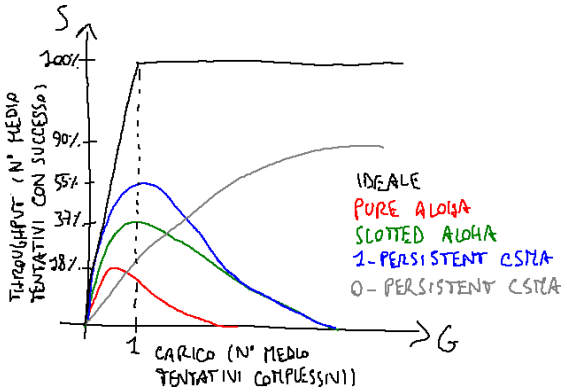


Figura 6.1: Prestazioni dei protocolli ad accesso casuale.

Capitolo 7

Standard LAN

802.3 è lo standard Ethernet cablata, 802.11 è lo standard per reti wireless: cambia il livello fisico, ma entrambi hanno un livello MAC.

Lo standard IEEE 802.3, basato su Ethernet, è CSMA/CD 1-persistente su topologia a bus.

Particolarità

- la sequenza di jamming è un segnale potente che viene mandato da chi si è accorto di una collisione per assicurarsi che tutti gli altri abbiano capito che è avvenuta una collisione;
- all'avvenuta ricezione non segue una conferma (ACK) alla stazione che ha trasmesso.

7.1 Livello fisico

A livello fisico viene usata la codifica Manchester[1] (transizioni) per mantenere il sincronismo indipendentemente dalla sequenza di bit, ma il segnale di clock è al doppio della velocità.

Mezzi trasmissivi L'Ethernet può usare cavi coassiali (10 BASE 2), doppini telefonici (10 BASE T) o fibre ottiche (10 BASE FL/FB/FP).

7.2 Livello MAC

L'**indirizzo MAC** permette di identificare la scheda, e si occupa di delimitare le trame (silenzi tra pacchetti o SFD) e di rilevare gli errori (il controllo degli errori è demandato al sottostrato LLC, ma è opzionale).

Gli indirizzi MAC sono composti da 6 byte, di cui i primi 3 byte sono lotti di indirizzi assegnati univocamente al costruttore (OUI). Un esempio di indirizzo MAC è 02-60-8C-07-9A-4D.

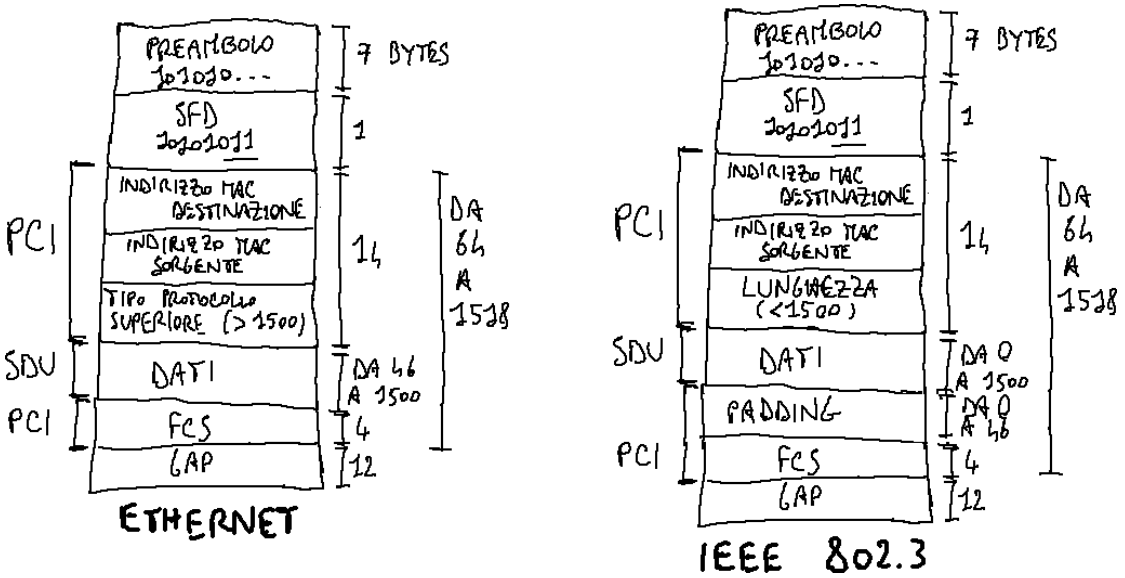
Un indirizzo MAC si dice di tipo **broadcast** quando tutti i bit sono a 1 (FF-FF-FF-FF-FF-FF).

Una scheda MAC quando riceve un pacchetto:

- se l'indirizzo MAC di destinazione coincide con quello di stazione, lo accetta;
- se l'indirizzo MAC di destinazione è di tipo broadcast, lo accetta;
- se l'indirizzo MAC di destinazione non coincide con quello di stazione, lo rifiuta.

La modalità promiscua accetta tutti i pacchetti ⇒ serve per lo sniffing di rete.

[1] Si veda la sezione "Codifica Manchester" nel capitolo "Collegamenti seriali sincroni" negli appunti di *Elettronica applicata e misure*.



7.2.1 PDU Ethernet

Nella PCI, oltre agli indirizzi MAC sorgente e di destinazione e al controllo CRC (chiamato FCS), vi è un campo per definire il tipo di protocollo di livello superiore trasportato, che deve contenere un numero maggiore di 1500.

Prima della PDU vi è:

- un preambolo costituito da 101010... per 7 byte che serve per recuperare la sincronizzazione;
- l'SFD corrispondente al byte 10101011 che indica l'inizio del pacchetto.

Alla fine della PDU vi è un silenzio minimo, detto Inter Packet GAP, che equivale a 12 byte, per lasciare spazio agli altri interlocutori.

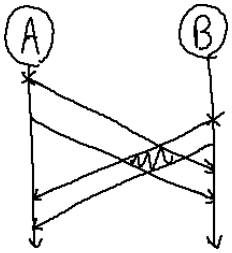
La dimensione minima della PDU è di 64 byte (SDU 46 byte), la massima è 1518 byte (SDU 1500 byte).

7.2.2 PDU IEEE 802.3

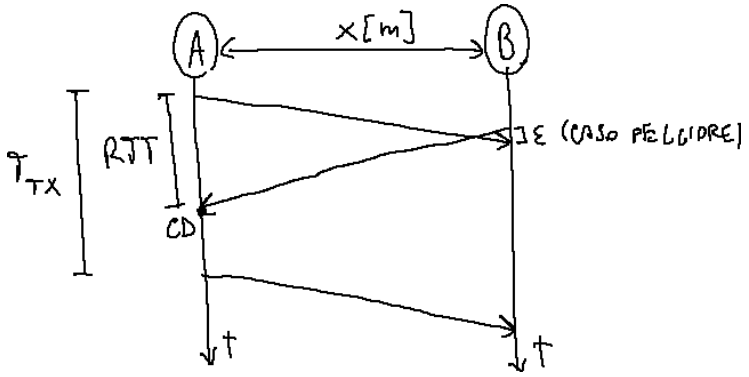
Il campo per il tipo di protocollo diventa superfluo perché il sottostrato di livello superiore è sempre LLC. Il campo viene comunque sfruttato per indicare la lunghezza della PDU di livello LLC trasportata, che ha una dimensione variabile tra 0 e 1500 byte. Per rendere compatibile il formato dei pacchetti IEEE 802.3 con il protocollo Ethernet occorre aggiungere fino a 46 byte di padding se la dimensione della PDU trasportata ha una dimensione inferiore a 46 byte. Siccome la lunghezza è obbligatoriamente inferiore al numero 1500, questo campo permette di distinguere i due tipi di PDU.

7.2.3 Dimensione minima delle PDU

È necessario stabilire una dimensione minima per le PDU perché se il pacchetto è troppo piccolo e la trasmissione collisa dura troppo poco può avvenire che nessuno si accorga della collisione:



La collision detection funziona solo se il round trip time RTT è minore (o uguale) del tempo di trasmissione T_{TX}:



$$\begin{cases} T_{\text{TX}} = \dfrac{L_{\text{PDU}}}{V_{\text{TX}}} \\ \text{RTT} = 2\dfrac{x}{c} \end{cases} \Rightarrow 2\frac{x}{c} \leq \frac{L_{\text{PDU}}}{V_{\text{TX}}} \Rightarrow L_{\text{PDU}} \geq \frac{V_{\text{TX}} \cdot 2x}{c}$$

Aumentare la velocità di trasmissione significa aumentare la dimensione minima dei pacchetti, oppure a parità di dimensione minima significa diminuire la distanza massima tra i nodi, ma pacchetti troppo grandi aumentano la probabilità di errore della trasmissione e intasano la rete.

7.3 Reti locali di nuova generazione

Le reti locali moderne adottano una topologia a stella.

Per realizzare una Ethernet a 100 Mb/s bisogna:

- aumentare la dimensione minima dei pacchetti;

o

- ridurre la dimensione della rete (soluzione utilizzata nel Fast Ethernet);

o

- modificare il protocollo di accesso.

Il Gigabit Ethernet aumenta ulteriormente la dimensione minima dei pacchetti.

Capitolo 8

Interconnessione LAN

Apparati di interconnessione

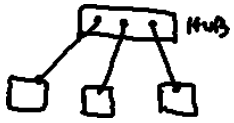
- repeater o hub[1]: è in grado di interconnettere solo reti di livello 1;
- bridge o switch[1]: interconnette reti di livello 2 ⇒ più complesso perché deve gestire il MAC e ha algoritmi di instradamento;
- router: interconnette reti di livello 3, e implementa IP;
- gateway: interconnette due reti completamente diverse e lavora a livello applicazione (livello 7).

8.1 Repeater o hub

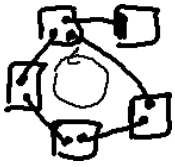
Il **repeater** si limita solo a trasmettere i bit. L'interconnessione di due domini di collisione crea un unico dominio di collisione ⇒ è un limite alla dimensione della rete.

Regeneration, reshaping, retiming (3R) L'unica intelligenza è a livello elettrico: si sincronizza con le onde quadre e le rigenera in modo da pulirle. La sincronizzazione richiede però un certo tempo ⇒ il preambolo che si trova all'inizio della PDU serve per la sincronizzazione.

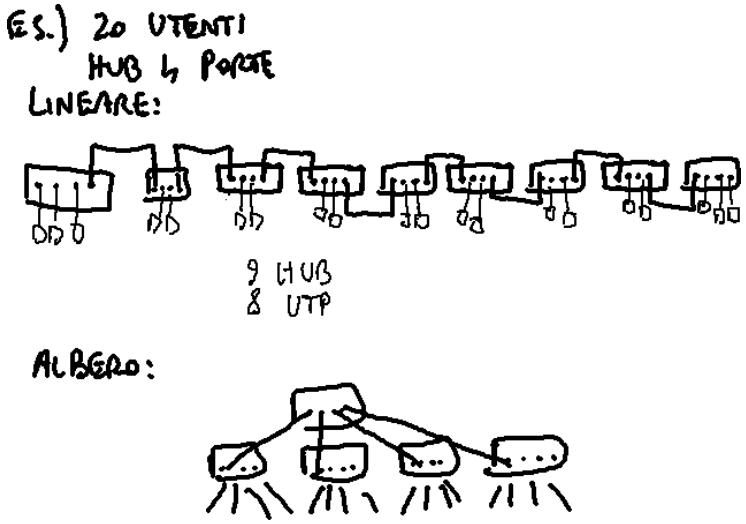
Topologia a stella Un hub si collega a più nodi, e il segnale elettrico proveniente da un nodo viene replicato su tutte le altre interfacce. Siccome il cavo UTP consente solo collegamenti punto-punto, ogni nodo deve essere collegato direttamente con l'hub:



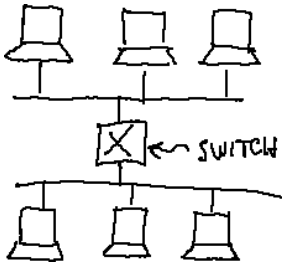
Non sono permessi anelli perché il segnale comincia a girare e inizia una collisione infinita:



[1] Questi due termini sono qui usati come sinonimi per semplicità, ma in realtà sono due oggetti differenti.



8.2 Bridge o switch

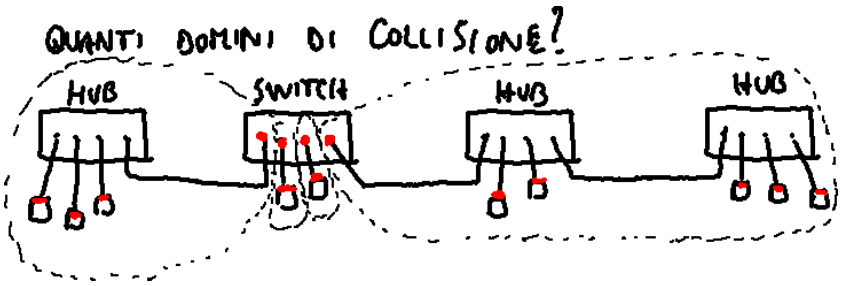


I **bridge** implementano il collision detect (CS), e si basano su store and forward. Un bridge può interconnettere anche livelli fisici e MAC diversi, perché è in grado di fare la traduzione delle intestazioni, purché i protocolli di livello superiore (LLC) siano uguali. I bridge non intervengono sul contenuto dei pacchetti.

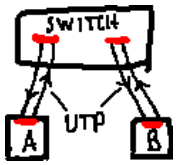
L'intelligenza di instradamento è molto semplice: quando arriva il segnale:

- lo memorizza (store);
- verifica se è valido;
- verifica la destinazione: se il pacchetto è destinato a un nodo in un altro dominio di collisione:
  - sente se la rete è occupata (CS);
  - lo trasmette all'altro dominio di collisione (forward).

L'interconnessione dev'essere trasparente all'utente. Un insieme di segmenti di LAN interconnessi mediante bridge è detto anche **LAN estesa**:



Collegare due PC, cioè due schede di rete, a due terminali separati di uno switch rende superfluo il collision detect:

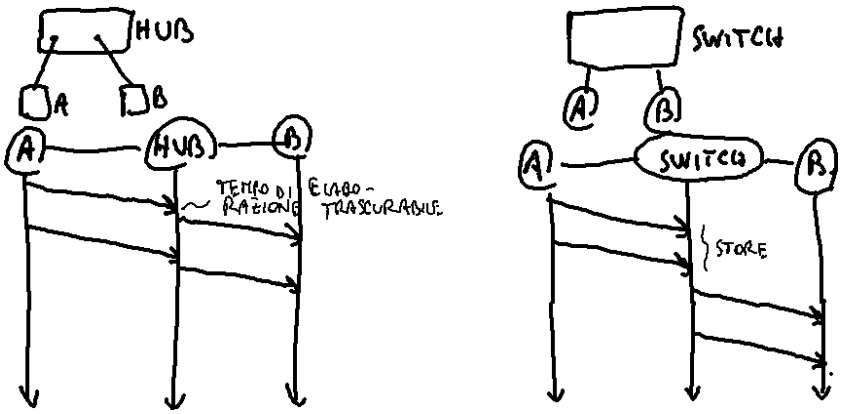


In questo caso conviene realizzare un collegamento **full-duplex**, cioè con CS disabilitato.

Il bridge interrompe la condivisione di risorse trasmissive tipica della LAN, creando due domini di collisione separati, perché la comunicazione tra due schede di un dominio non interferisce con la comunicazione tra due schede di un altro dominio. Nel caso il pacchetto abbia destinazione broadcast (FFFF...), il dominio invece è unico. Lo switch quindi permette di estendere l'estensione geografica della rete senza problemi. Anche lo switch rigenera il segnale. Lo switch introduce anche sicurezza perché un nodo non può sniffare comunicazioni tra altri nodi.[2]

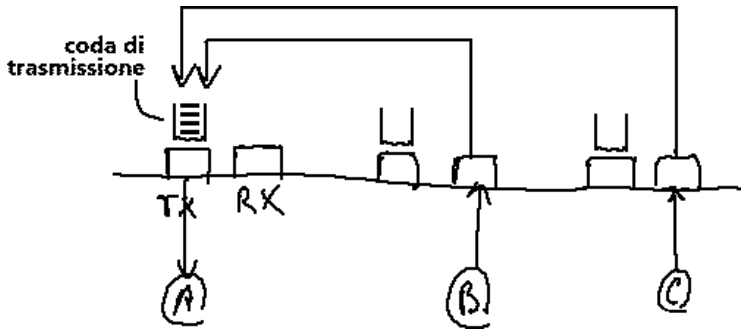
8.2.1 Svantaggi

- ritardi di store and forward rispetto all'hub:

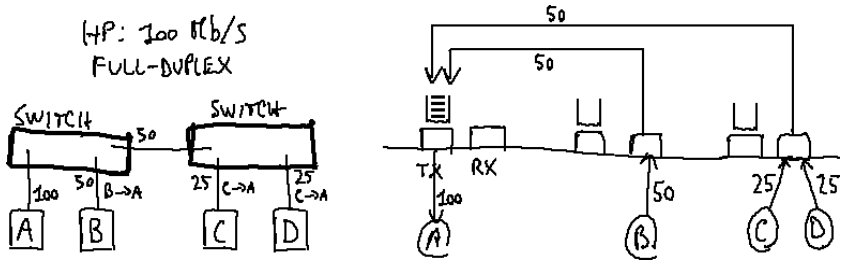


- possibilità di perdita di pacchetti per overflow/congestione delle code di trasmissione:

[2]Esistono in realtà dei metodi.



- problemi di equità nella condivisione della banda (gli hub invece garantiscono che la banda sia spartita equamente tra tutti):



8.2.2 Transparent bridge

Ogni porta dello switch deve avere un proprio indirizzo MAC. L'instradamento è svolto in maniera trasparente: lo switch cerca di imparare la posizione dei nodi ad esso collegati riempendo una forwarding table.

Se l'utente sposta il PC in un'altra porta, lo switch deve supportare lo spostamento e correggere la forwarding table. C'è un timer: oltre un certo tempo un MAC memorizzato si cancella.

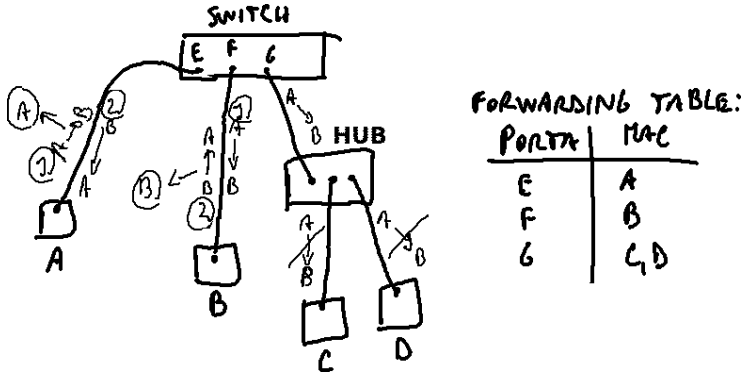


Figura 8.1: Address learning e frame forwarding.

Address learning

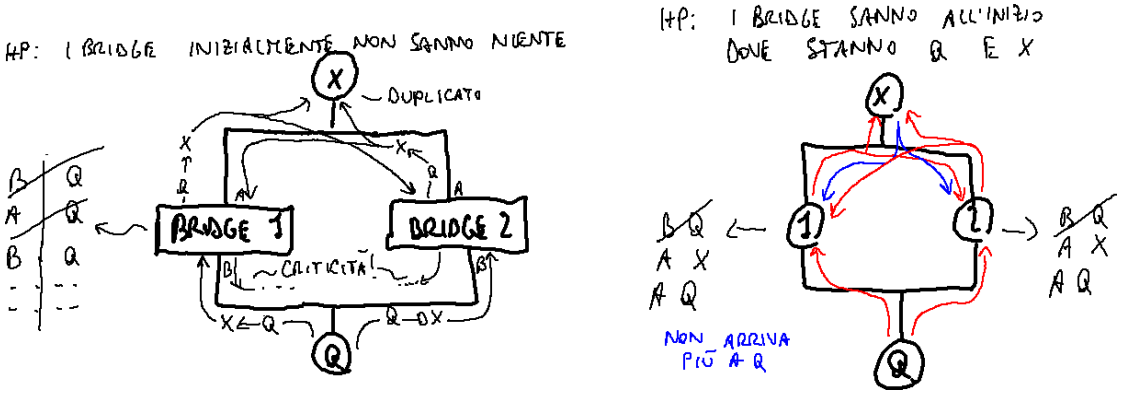
Nell'**address learning** l'apprendimento è basato sugli indirizzi MAC sorgenti dei pacchetti, tramite un algoritmo di **backward learning**.

Frame forwarding

Nel **frame forwarding** l'apprendimento è basato sull'indirizzo MAC di destinazione: quando arriva un pacchetto la cui destinazione non è ancora presente nella forwarding table, lo switch manda il pacchetto in broadcast (a tutti), e attende la risposta che molto probabilmente la destinazione invierà.

Spanning tree

L'algoritmo di backward learning funziona solo se in topologia non ci sono degli anelli, altrimenti i nodi continuano a ricevere pacchetti identici all'infinito, oppure l'invio di pacchetti si blocca:



L'**algoritmo spanning tree** serve per eliminare gli anelli logici dalla topologia fisica: quando si verifica un guasto lo switch ricalcola automaticamente uno spanning tree per evitare le criticità.

Limiti

- non si possono aggiungere degli switch per creare dei collegamenti alternativi che si spartiscano il traffico perché si creerebbero degli anelli in topologia;
- se la rete è molto grande, la forwarding table diventa molto grande;
- il traffico broadcast dev'essere confinato, cioè non dev'essere mandato indistintamente a tutti i nodi.

Parte II

Internet: architettura e protocolli

Capitolo 9

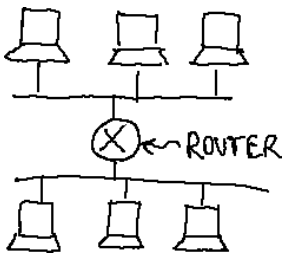
Protocollo IPv4

L'**Internet Protocol** (IP) è il livello di rete di TCP/IP:

- è un servizio non connesso: non c'è segnalazione di rete o utente;
- è un protocollo di tipo datagram: basato su pacchetti.

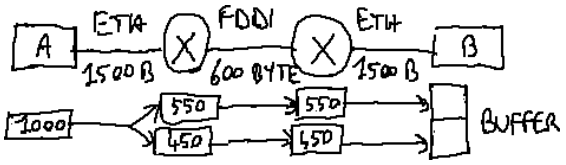
Funzionalità

- frammentazione e riassemblaggio dei pacchetti per il trasporto sul livello 2;
- gestione di indirizzi a 32 bit (IPv4);
- configurazione delle classi di servizio: i progettisti di IP avevano previsto una differenziazione del traffico per priorità, ma oggi le classi di servizio non sono sfruttate.



Gli apparati di interconnessione di livello 3 sono chiamati **router**:

- a differenza degli switch evitano l'intasamento della forwarding table;
- effettuano solo la frammentazione, mentre il riassemblaggio è delegato alla destinazione:



9.1 Pacchetti IP

9.1.1 Intestazione

Il formato dell'intestazione di un pacchetto IP è composto da vari campi:

- versione: IPv4 o IPv6;
- tipo di servizio (Type of Service [ToS]): per le classi di servizio;
- lunghezza totale del pacchetto prima della deframmentazione;
- identificatore: tutti i frammenti di uno stesso pacchetto condividono lo stesso identificatore;
- flag: indica se il pacchetto corrente è l'ultimo frammento;
- offset di frammento: per l'ordinamento del pacchetto nella sequenza;
- tempo di vita (Time To Live [TTL]): è un contatore di **hop**, cioè è un numero che viene decrementato di un'unità a ogni passaggio dentro un router ⇒ quando arriva a 0 il pacchetto viene scartato per evitare che pacchetti vaghino all'infinito;
- protocollo: identifica il protocollo di livello 4 del pacchetto payload (TCP, UDP, ICMP...);
- checksum: protegge solo l'intestazione;
- indirizzi IP della sorgente e della destinazione;
- alcune opzioni oggi abbandonate.

9.1.2 Indirizzi

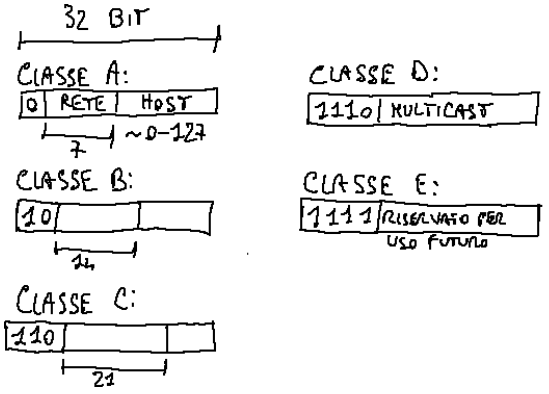


Figura 9.1: Classi di indirizzi IP.

Ogni indirizzo IP è ampio 4 byte:

- campo rete: interfacce vicine sono aggregate in un **prefisso di rete** comune che occupa i bit alti;
- campo host: identifica la macchina specifica.

Un indirizzo IP è umanamente rappresentabile suddividendo i 32 bit in gruppi da 8, convertendo ogni gruppo in decimale e separandoli con dei puntini ⇒ ogni numero è compreso tra 0 e 255.

Non tutti gli indirizzi possibili sono utilizzabili:

- indirizzi con tutti 0 nel campo host identificano l'intera rete e non possono essere utilizzati per un singolo host;
- indirizzi con tutti 1 nel campo host identificano il broadcast di tipo **directed** a tutti gli host nella rete;

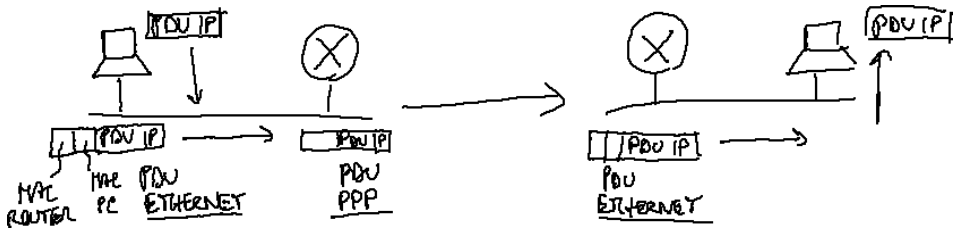
- indirizzi con tutti 1 identificano il broadcast di tipo **limited** a tutti gli host della rete locale;
- indirizzi che iniziano con 127 identificano la macchina stessa e bloccano il forwarding di traffico (**loopback**);
- indirizzi privati: gli indirizzi pubblici devono essere univoci in tutta Internet, gli indirizzi privati sono gestiti all'interno della rete privata che dall'esterno viene vista con un unico indirizzo IP (NAT[1]):
  - 1 rete privata di classe C, 2^{24} indirizzi: tra 10.0.0.0 e 10.255.255.255;
  - 15 reti private di classe B, 16 indirizzi: da 172.16.0.0 a 172.31.255.255;
  - 256 reti private di classe C, 256 indirizzi: da 192.168.0.0 a 192.168.255.255.

Gli indirizzi IP sono assegnati alle interfacce ⇒ un router non ha un unico indirizzo, ma ne ha uno per ogni interfaccia.

9.2 Reti fisiche e reti logiche

Una **rete fisica** è un insieme di host connessi in una rete senza router ⇒ il livello fisico non viene interrotto da un dispositivo di livello 3.

Una **rete logica** (Logical IP Subnet [LIS]) è un insieme di host con lo stesso prefisso di rete, e include la rete fisica e il router. Ogni rete fisica corrisponde a livello 3 a una rete logica.



La comunicazione tra due macchine appartenenti a una stessa rete fisica è basata sugli indirizzi MAC dei due host (CSMA/CD). In realtà il livello applicazione non sa che l'host di destinazione si trova nella stessa rete fisica, e fornisce l'indirizzo IP ⇒ **consegna diretta** al livello 2: è il livello 3 che deve capire che la destinazione è nella rete fisica e far partire i meccanismi di livello 2 senza passare dal router.

La consegna tra reti logiche differenti è affidata al router (**consegna indiretta**). Il livello 2 è comunque usato per le consegne dirette tra la sorgente e il router, e tra il router e la destinazione. L'host deve quindi conoscere almeno un **default gateway**, che è il router per uscire dalla rete logica.

Esiste la possibilità di emulare due reti logiche in una stessa rete fisica: una stessa interfaccia è vista da un host con un certo indirizzo IP e da un altro host con un altro indirizzo IP, e la comunicazione tra i due host passa sempre attraverso il router. In realtà esistono dei meccanismi che permettono di ottimizzare il percorso (routing table, ARP table): dopo la prima comunicazione, il router istruisce il primo host affinché la volta dopo svolga la consegna diretta a livello 2.

9.2.1 Subnetting

Il **subnetting** consente partendo da una rete di una certa classe di creare una rete a classe più piccola: il prefisso di rete si espande e una parte del campo host diventa il campo subnet.

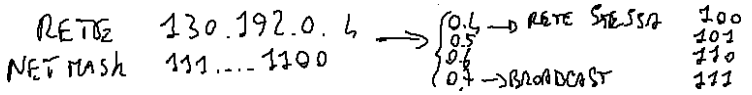
La **subnet mask** è una sequenza di bit:

[1] Si veda il capitolo 16.

- i byte 255 identificano il prefisso di rete, compreso il campo subnet;
- i byte 0 identificano il campo host ridotto.

Queste reti a classe più piccola sono delle reti logiche distinte ⇒ il subnetting si può usare per emulare più reti logiche in una stessa rete a classe maggiore.

9.2.2 Indirizzamento classless (CIDR)



La **netmask** permette di estendere il prefisso di rete a un qualsiasi bit dell'indirizzo:

- i bit 1 identificano il prefisso di rete;
- i bit 0 identificano il campo host.

In questo modo più reti piccole possono venire aggregate in un unico router ⇒ si definiscono delle gerarchie di indirizzamento.

Non sono ammesse le netmask 255.255.255.255 (/32) e 255.255.255.254 (/31) perché l'indirizzo con tutti 0 è riservato alla rete stessa, e quello con tutti 1 al traffico broadcast.

9.2.3 Routing

Dalla netmask il router può capire se la destinazione si trova nella stessa rete del mittente, confrontando i risultati di due operazioni AND bit a bit:

- il primo AND bit a bit è tra l'indirizzo del mittente e la netmask del mittente;
- il secondo AND bit a bit è tra l'indirizzo del destinatario e la netmask del mittente.

Quando al router arriva un datagramma:

- se la destinazione coincide con il router stesso: elaborazione locale (per configurare il router);
- se la destinazione è in uno degli address range del router: ARP,[2] invio diretto all'indirizzo MAC di destinazione, ecc. (se l'indirizzo di destinazione appartiene a più address range, viene preferito quello con la netmask più lunga (Longest Prefix Matching));
- se la destinazione è fuori da tutti gli address range del router: consulta la routing table, quindi invia il datagramma al "prossimo hop" indicato nella routing table (o sulla default route).

Tabelle di routing

Una **tabella di routing** può contenere:

- entry dirette: sono le coppie indirizzo di rete (con 0 finali) + netmask associate a ogni interfaccia del router;
- entry indirette: sono gli indirizzi IP delle interfacce degli altri router (next hop) a cui fare le consegne dirette:
  - statiche: sono configurate manualmente dal gestore;
  - dinamiche: sono determinate in modo automatico.

[2] Si veda il capitolo 10

Si scelgono sempre le route a minor costo.

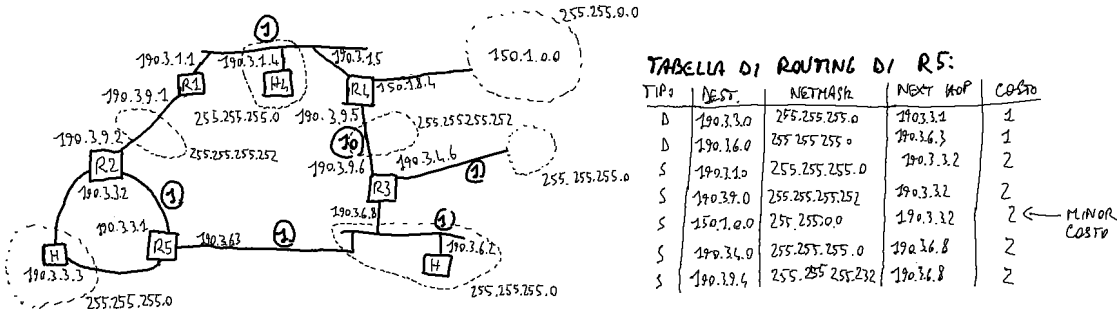


Figura 9.2: Esempio di routing table.

Più address range con uguali bit alti si posso aggregare, mettendo degli zeri al posto dei bit che differiscono:

Tabella 9.1: A sinistra la routing table non aggregata, a destra la routing table aggregata.

Destinazione	**Netmask**	**Next hop**
190.3.1.0	255.255.255.0	190.3.3.2
190.3.9.0	255.255.255.252	190.3.3.2
190.3.**9.4**	255.255.**255.252**	190.3.6.8
190.3.**4.0**	255.255.**255.0**	190.3.6.8
150.1.0.0	255.255.0.0	190.3.6.8

Destinazione	**Netmask**	**Next hop**
190.3.1.0	255.255.255.0	190.3.3.2
190.3.9.0	255.255.255.252	190.3.3.2
190.3.**0.0**	255.255.**0.0**	190.3.6.8
150.1.0.0	255.255.0.0	190.3.6.8

Il default gateway ha sempre netmask 0.0.0.0, perché viene scelto solo se non ci sono altri address range che risultano validi dai due AND bit a bit.

Capitolo 10

Protocollo ARP

I protocolli ARP e RARP si poggiano sul livello 2, ma non fanno propriamente parte del livello 3.

10.1 ARP

Il **protocollo ARP** serve per venire a sapere l'indirizzo MAC della destinazione partendo dal suo indirizzo IP:

1. **ARP request:** il mittente invia in broadcast un pacchetto in cui inserisce nel payload l'indirizzo IP di cui vuole conoscere l'indirizzo MAC;
2. **ARP reply:** la destinazione risponde comunicando il proprio indirizzo MAC;
3. il mittente memorizza nella propria **tabella ARP** la coppia indirizzo IP-indirizzo MAC (assegnando un TTL, cioè un tempo di vita oltre il quale la entry viene cancellata).

Se la destinazione si trova al di fuori della rete logica del mittente, l'ARP request va fatta al router della rete ⇒ nella consegna diretta, l'indirizzo MAC di destinazione è quello del router, mentre l'indirizzo IP contenuto nel payload rimane l'indirizzo della destinazione al di fuori della rete.

10.2 RARP

Il **protocollo Reverse ARP** (RARP) serve per venire a sapere il proprio indirizzo IP partendo dal proprio indirizzo MAC. Un nodo diskless, cioè privo di supporti di archiviazione, quando si connette ad una rete non è in grado di memorizzare l'indirizzo IP che ad esso è stato assegnato, e che sarà necessario inserire come indirizzo del mittente a ogni comunicazione ⇒ un server ARP, dotato di dischi, si occupa di salvare le associazioni indirizzi IP-indirizzi MAC di tutti i nodi connessi alla rete locale.

Quando il nodo diskless vuole sapere il proprio indirizzo IP:

1. **RARP request:** invia in broadcast un pacchetto in cui inserisce il proprio indirizzo MAC;
2. **RARP reply:** il server RARP risponde comunicando l'indirizzo IP che è stato assegnato a quell'indirizzo MAC.

Il protocollo RARP è stato soppiantato dal più flessibile DHCP.

Capitolo 11

Protocollo ICMP

L'**Internet Control Message Protocol** (ICMP) fornisce delle funzionalità di controllo sul livello IP:

- verifica lo stato della rete;
- riporta eventuali anomalie;
- scopre la netmask (compito oggi svolto dal DHCP);
- migliora il routing.

Non è in grado di rilevare errori che si verificano nei livelli sottostanti.

11.1 Intestazione dei pacchetti

Un pacchetto ICMP può essere il payload di un pacchetto IP.

L'intestazione di un pacchetto ICMP ha lo scopo di comunicare un errore avvenuto nella rete:

- tipo di messaggio;
- codice del messaggio (sottotipo);
- intestazione IP + i primi 64 bit del pacchetto che ha generato un errore.

11.2 Messaggi

11.2.1 (0) Echo Reply - (8) Echo Request

Verifica se un host è raggiungibile. Questo messaggio è usato dal comando `ping`.

11.2.2 (3) Destination Unreachable

Segnala che la destinazione non è raggiungibile, e il motivo è specificato nel codice:

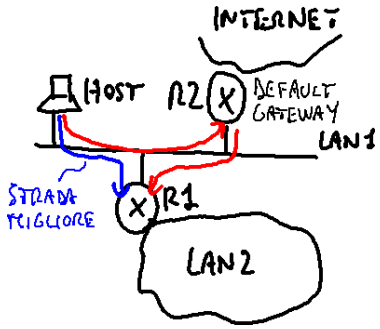
- (0) Network unreachable: il router non trova la destinazione nella sua tabella di routing;
- (1) Host unreachable: il router non trova la destinazione per la consegna diretta (per es., non ha risposto nessuno a una ARP Request);
- (2) Protocol unreachable: l'host non trova il protocol type nel pacchetto IP;
- (3) Port unreachable: la porta di livello 4 richiesta dal livello applicazione dell'host non è attiva;

- (4) Fragmentation needed and DF set: il router ha bisogno di frammentare il pacchetto, ma il mittente ha richiesto di non frammentare il pacchetto;
- (5) Source routed failed: il mittente può specificare manualmente il path all'interno del pacchetto, ma quel particolare path può non esistere.

11.2.3 (4) Source Quence

Può essere inviato da un router quando sta per esaurire il buffer, e chiede al mittente di abbassare il rate ⇒ oggi non viene usato perché il TCP si occupa anche di rallentare il rate di invio dei pacchetti.

11.2.4 (5) Redirect



Migliora il routing: si accorge se l'interfaccia di uscita di un pacchetto da un router corrisponde all'interfaccia di ingresso di quel pacchetto ⇒ esiste una strada migliore per raggiungere la destinazione.

11.2.5 (11) Time Exceeded for a Datagram

Viene generato quando il TTL del pacchetto diventa zero, cioè quando il pacchetto ha attraversato un numero di hop pari al valore iniziale del TTL ⇒ il TTL limita gli effetti di un loop imprevisto. Questo messaggio è usato dal comando `traceroute`.

Capitolo 12

Domain Name System

Il **Domain Name System** (DNS) serve per risolvere (convertire) i nomi in indirizzi IP e viceversa:

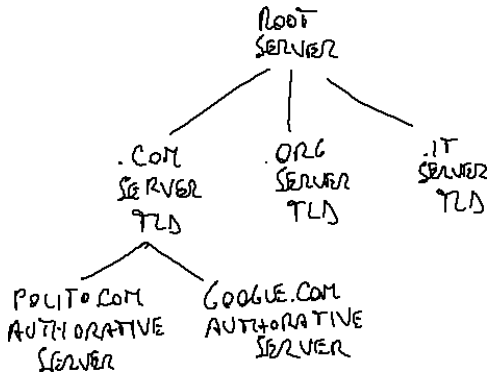
- database distribuito: contengono tutte le coppie nomi-indirizzi;
- protocollo di livello applicazione: per la gestione delle interrogazioni al database.

Funzionalità

- risoluzione di nomi in indirizzi;
- host aliasing: oltre al nome canonico, a un indirizzo IP possono venire associati altri nomi chiamati alias (soprannomi) (ad esempio mail.polito.it è associato a polito.it);
- mail server aliasing: riguarda gli alias di posta elettronica;
- load distribution: allo stesso nome possono essere associati più indirizzi IP per il load balancing tra vari server.

12.1 Gerarchia del database distribuito

Non esiste un unico server centralizzato, ma vari server organizzati in modo gerarchico:



Root server Vari root server gemelli sono distribuiti geograficamente nel mondo. La ricerca di un sito parte sempre dal root server e percorre un ramo dell'albero.

Top-level domain (TLD) server I TLD server sono responsabili per i domini .com, .org, .net, ecc. Per registrare un sito in un dominio .com, per esempio, bisogna contattare Network Solutions.

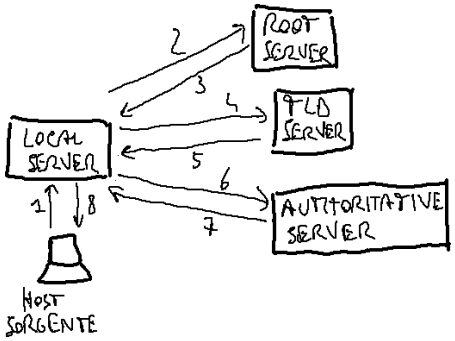
Authoritative DNS server Gli authoritative DNS server sono i server DNS veri e propri che individuano gli indirizzi IP dei vari siti.

Local DNS name server Un gateway domestico ha un nome associato al loro IP, a cui si può accedere dai PC collegati. Hanno funzioni di caching: memorizzano temporaneamente la risposta del server DNS, per migliorare le prestazioni se quel sito viene contattato dopo poco tempo.

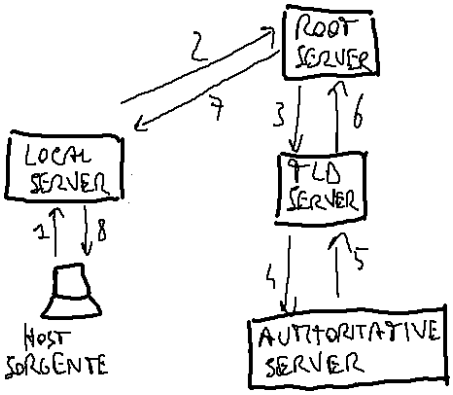
12.2 Query

Tipi di query

- **query iterativa:** un server risponde dicendo qual è il prossimo server da contattare;
- **query ricorsiva:** il server si occupa di forwardare la richiesta agli altri server.



(a) Query iterativa.



(b) Query ricorsiva.

I record del database (RR) hanno il seguente formato:

`(name, value, type, ttl)`

Tipo A

- `name` = nome dell'host
- `value` = indirizzo IP (tipo AAAA se IPv6)

Tipo CNAME

- `name` = alias
- `value` = nome canonico

Tipo Name Server (NS) È la entry di livello superiore che gestisce il puntatore al server autoritativo.

- `name` = dominio
- `value` = nome del server autoritativo per il dominio

Tipo MX

- `value` = nome del mailserver associato a `name`

Tipo PTR Gestisce i mapping da un indirizzo IP al suo nome, ad esempio per il traceroute.

Il database deve essere dinamico perché i mapping possono essere cambiati ⇒ le entry della cache hanno TTL.

12.3 Formato dei pacchetti

Nell'intestazione:

- identification: identifica se si tratta di una domanda o di una risposta;
- flag:
  - ricorsione desiderata/disponibile: servono per sapere se si può fare la query ricorsiva;
  - reply is authoritative: la query non è stata risolta da alcuna cache ma è arrivata fino al server autoritativo;
- altri campi: questions, answers, authority, additional info.

12.4 Registrazione dei domini

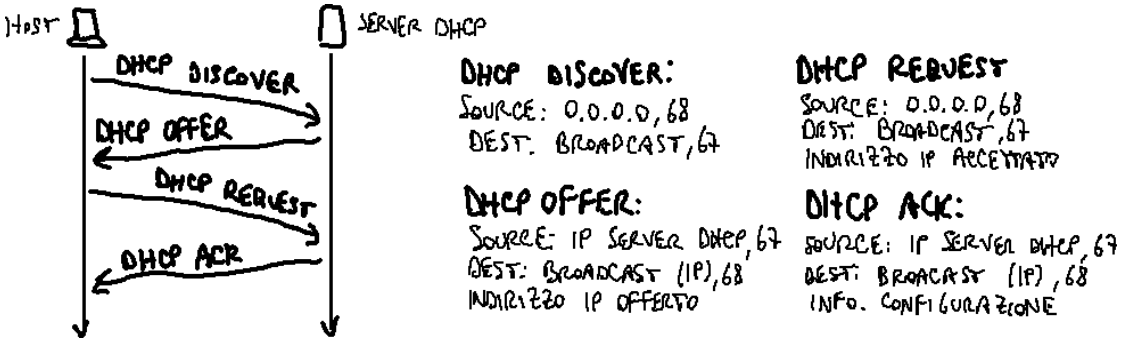
I **DNS registrar** sono gli enti a cui ci si deve rivolgere per registrare nuovi domini. Un DNS registrar fornisce anche:

- due server autoritativi (primario e secondario), perché se il primario smette di funzionare subentra il secondario;
- due entry: una di tipo NS per arrivare al server autoritativo, l'altra di tipo A per l'indirizzo IP.

Capitolo 13

Protocollo DHCP

Il **Dynamic Host Configuration Protocol** (DHCP) è un protocollo di livello applicazione che permette a un host di ottenere dinamicamente dal server DHCP vari parametri di configurazione, tra cui il proprio indirizzo IP:



1. **DHCP discover:** viene inviato dall'host in broadcast;
2. **DHCP offer:** il server DHCP offre all'host un indirizzo IP; l'indirizzo IP di destinazione è sempre broadcast perché l'host non ha ancora un indirizzo IP, ma l'indirizzo MAC di destinazione è unicast verso l'host;
3. **DHCP request:** l'host accetta l'indirizzo IP offerto dal server DHCP;
4. **DHCP ACK:** il server DHCP conferma la ricezione del DHCP request, e invia insieme all'indirizzo IP assegnato anche l'indirizzo IP del server DNS e l'indirizzo IP del router.

Il pacchetto DHCP diventa il payload del pacchetto UDP, a sua volta payload del pacchetto IP.

Ogni indirizzo IP assegnato tramite il DHCP ha un tempo di vita chiamato **lease**.

Ci possono essere più server DHCP che offrono indirizzi IP all'host, che ne sceglie uno.

Capitolo 14

Livello Trasporto: Protocolli TCP-UDP[1]

Il **Transmission Control Protocol** (TCP) e lo **User Datagram Protocol** (UDP) sono due protocolli di livello trasporto.

Ogni host ha un unico indirizzo IP, ma può avere in esecuzione più applicazioni. Ogni applicazione comunica attraverso una porta TCP e una porta UDP; il livello di trasporto effettua poi il **multiplexing** delle connessioni, cioè gestisce i flussi provenienti dalle varie porte verso il protocollo di rete sottostante.

14.1 TCP

14.1.1 Formato dell'intestazione dei pacchetti TCP

L'intestazione dei pacchetti TCP ha il formato seguente:

- **source port:** la porta TCP sorgente;
- **destination port:** la porta TCP di destinazione;
- **numero di sequenza**;
- **acknowledgment number:** la destinazione specifica il numero di sequenza del prossimo pacchetto da ricevere;
- **hlen:** la lunghezza dell'intestazione;
- **flag:** identificano l'informazione contenuta nel pacchetto:
  - il flag ACK identifica se il pacchetto è un ACK;
  - il flag SYN serve per l'apertura della connessione;
  - il flag FIN serve per la chiusura della connessione;
- **window size:** la dimensione della finestra di ricezione del mittente del pacchetto, per permettere l'uso dei protocolli a finestra;[2]
- **checksum**.

[1] Questo capitolo include contenuti CC BY-SA da Wikipedia in inglese: Transmission Control Protocol.
[2] Si veda il capitolo 3.

Numero di sequenza

Il TCP è un protocollo orientato ai byte: il primo pacchetto inizia con un numero casuale, il secondo pacchetto ha quel numero incrementato del numero di byte di cui è costituito il primo pacchetto, e così via.

I buffer del mittente (coda di trasmissione) e della destinazione (coda di ricezione) non sono necessariamente di uguale dimensione ⇒ i dati possono venire segmentati in un modo e riassemblati in un altro.

14.1.2 Circuito virtuale

Il TCP è un protocollo connesso: funziona tramite **circuito virtuale** tra una porta del mittente e una porta del destinatario.

Il circuito virtuale è **full-duplex**: consente la comunicazione in entrambi i sensi.

Esistono delle porte standard, chiamate **well-known port**, tramite cui può essere contattato l'applicativo di destinazione: ad ogni well-known port è assegnato un protocollo ben preciso ⇒ l'applicativo di destinazione riesce a sapere quale protocollo vuole usare l'applicativo sorgente per la comunicazione. Ad esempio, la porta 80 è riservata alla comunicazione tramite il protocollo HTTP. Siccome la comunicazione è full-duplex, la risposta uscirà dalla porta 80 dell'applicativo di destinazione e giungerà alla porta aperta prima dall'applicativo sorgente per l'invio della richiesta. Le well-known port sono 1024, una piccola parte rispetto a tutte le porte utilizzabili.

Apertura della connessione

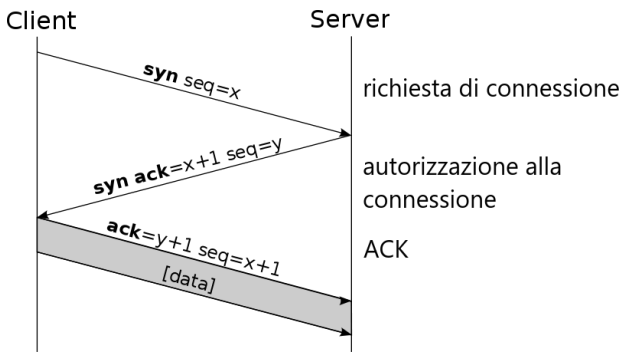


Figura 14.1: Three-way handshake di TCP.[3]

Per aprire una connessione, il TCP usa un three-way handshake:[4]

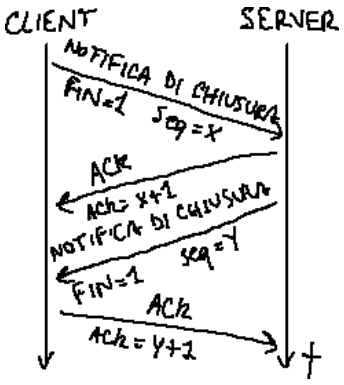
- **SYN:** il client invia un SYN al server, impostando il numero di sequenza del segmento ad un valore casuale x;
- **SYN-ACK:** il server risponde con un SYN-ACK, impostando il numero di acknowledgment al numero di sequenza ricevuto più uno $(x+1)$, e scegliendo come numero di sequenza del pacchetto un altro numero casuale y;
- **ACK:** infine il client invia indietro un ACK al server, impostando il numero di sequenza al valore di acknowledgement ricevuto $(x+1)$, e il numero di acknowledgement al numero di sequenza ricevuto più uno $(y+1)$.

A questo punto, sia il client sia il server hanno ricevuto un acknowledgment della connessione e la comunicazione full-duplex è stabilita.

[3] Questa immagine è derivata da un'immagine su Wikimedia Commons (Apertura connessione TCP.png), realizzata dall'utente snubcube, ed è concessa sotto la licenza Creative Commons Attribuzione - Condividi allo stesso modo 3.0 Unported.

[4] Si veda la voce Handshaking su Wikipedia in italiano.

Chiusura della connessione



La chiusura della connessione richiede l'invio di FIN e ACK in entrambe le direzioni. È una chiusura di tipo graceful leaving, perché gli interlocutori non abbandonano improvvisamente la conversazione ma prima la chiudono interagendo in maniera "educata".

14.1.3 Uso dei protocolli a finestra

Il TCP richiede più banda dell'UDP, a causa delle maggiori dimensioni dell'header dovute ai maggiori controlli: a differenza dell'UDP, il TCP effettua il **controllo di flusso** e **di congestione**, e garantisce che tutti i pacchetti vengano consegnati integri a destinazione. L'acknowledge è obbligatorio (normalmente tramite piggybacking[5]), e se un pacchetto non è giunto a destinazione va ritrasmesso.

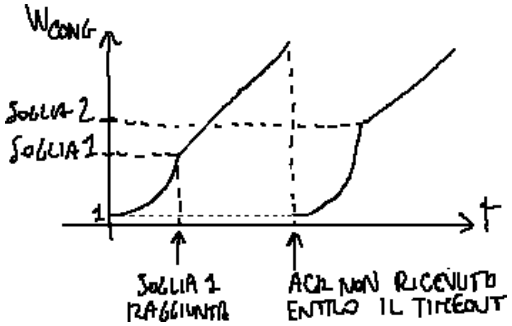
Se un segmento va perso e arrivano segmenti successivi nella sequenza, il ricevitore inizia ad inviare NACK duplicati riferiti al segmento mancante. Il trasmettitore capisce che deve ritrasmettere il segmento quando riceve 3 NACK duplicati.

La dimensione della finestra di trasmissione viene modificata dinamicamente in funzione della capacità della rete e del ricevitore:

$$\text{finestra di trasmissione} = \min(\text{finestra di ricezione, finestra di congestione})$$

Il trasmettitore conosce la finestra di ricezione grazie al campo "window size" nell'intestazione dei pacchetti TCP.

Finestra di congestione



La **finestra di congestione** serve per evitare di congestionare la rete limitando la finestra di trasmissione. All'apertura della connessione la finestra di congestione ha dimensione pari a 1, cioè viene mandato un solo pacchetto. Man mano che arrivano gli ACK, la finestra di congestione:

[5]Si veda la sezione 3.2.1.

- cresce esponenzialmente (2, 4, 8...) fino ad arrivare a un certo **valore di soglia**, che la prima volta è predefinito;
- superato il valore di soglia cresce linearmente.

Se un riscontro non arriva entro il timeout, il valore di soglia viene impostato alla metà del valore corrente della finestra di congestione e si riparte da 1.

14.2 UDP

Il protocollo UDP è privo dei controlli di flusso e di congestione ⇒ il trasmettitore continua a trasmettere alla massima velocità consentita dalla banda di trasmissione senza aspettare alcun ACK. Esegue solamente il controllo degli errori, cioè verifica il checksum del pacchetto per verificarne l'integrità.

L'UDP è un protocollo non connesso:

- non vi è alcun ritardo di apertura e di chiusura della connessione;
- è basso l'overhead, cioè il carico sul sistema operativo per tenere aperte le connessioni;
- i ritardi sono bassi e costanti, cioè la distanza tra un pacchetto e l'altro è uguale per tutti i pacchetti.

L'intestazione dei pacchetti UDP è molto più breve e semplice:

- source port;
- destination port;
- UDP length;
- UDP checksum.

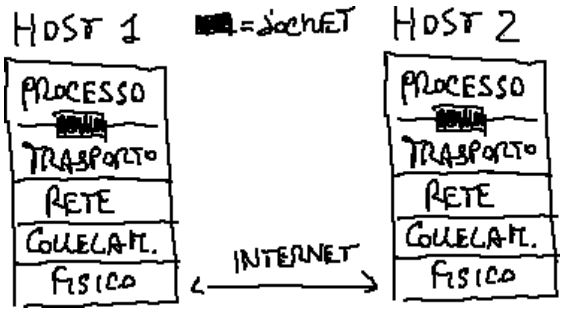
L'UDP è utile:

- quando la rete è affidabile ⇒ i controlli non servono;
- quando le prestazioni e la riduzione del ritardo sono più importanti dell'affidabilità (es. telefonia);
- quando l'applicazione mette tutti i dati in un singolo pacchetto ⇒ non servono meccanismi per regolare la dimensione della finestra.

Eventuali meccanismi di ritrasmissione, come l'invio di ACK, devono essere gestiti a livello applicazione.

Capitolo 15

Application Layer



Il **socket** di un processo applicativo è il canale di comunicazione con il livello trasporto: il processo applicativo di un host e il processo applicativo di un altro host si scambiano messaggi attraverso i loro socket.

La comunicazione tra due host può avvenire secondo il paradigma client-server o in maniera peer-to-peer (P2P).

Le applicazioni possono avere diverse esigenze a seconda della loro funzione:

- **integrità dei dati:** un'e-mail deve arrivare intera, mentre una telefonata può tollerare la perdita di qualche campione;
- **ritardi:** una telefonata è molto sensibile ai ritardi nella conversazione;
- **throughput:** le applicazioni multimediali non funzionano se la rete è troppo carica, mentre le **applicazioni elastiche** come il caricamento di una pagina Web sono in grado di adattarsi al throughput corrente della rete;
- **sicurezza:** il protocollo Secure Sockets Layer (SSL) fornisce una connessione TCP criptata per motivi di sicurezza.

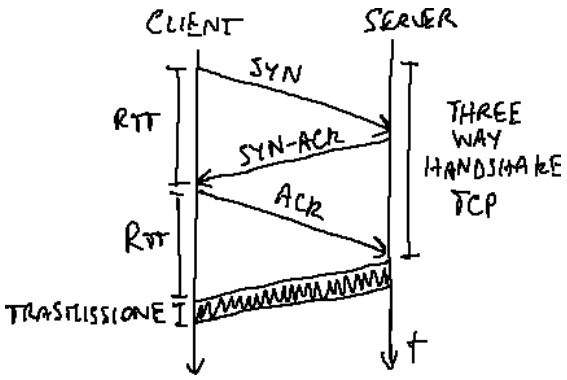
Le applicazioni usano vari **protocolli di livello applicazione**: SMTP per la posta elettronica, HTTP per le pagine Web, ecc.

15.1 Protocollo HTTP

Il **protocollo Hypertext Transfer Protocol** (HTTP) serve per il recupero di pagine Web dal server. Utilizza il TCP (tipicamente sulla well-known port 80) perché non sono ammesse perdite di pacchetti.

15.1.1 Persistenza

La versione 1.0 di HTTP è di tipo **non persistente**: viene aperta una connessione TCP per ogni oggetto (immagini) che costituisce la pagina Web.



La ricezione di ogni oggetto richiede l'attesa di 2 RTT + il tempo di trasmissione del file:

1. viene aperta la connessione TCP tramite three-way handshake;[1]
2. viene trasferito l'oggetto richiesto dal client.

La versione 1.1 di HTTP è di tipo **persistente:** tutti gli oggetti della pagina Web vengono scaricati entro un'unica connessione TCP ⇒ risparmio di tempo.

15.1.2 Messaggi HTTP

Messaggi di richiesta Il client invia dei **messaggi di richiesta**, nella cui intestazione specifica alcuni campi ASCII tra cui il tipo di comando:

- `GET`: richiesta di un oggetto, ad esempio:
  ```
  GET /index.html HTTP/1.1
  ```
- `POST`: invio di input da form, contenuto nel corpo del messaggio;
- `HEAD`: uguale a GET, ma il server invia nel messaggio di risposta solo l'intestazione e non il contenuto dell'oggetto;
- `PUT`: invio di un file nel corpo del messaggio (solo HTTP/1.1);
- `DELETE`: eliminazione di un file (solo HTTP/1.1).

L'intestazione contiene anche il campo `Connection`:

- nell'HTTP non persistente il valore predefinito è `Close`, perché bisogna chiudere la connessione TCP corrente prima di richiedere l'oggetto successivo;
- nell'HTTP persistente il valore predefinito è `Keep-Alive`, perché la connessione va chiusa solo dopo che è stato ricevuto l'ultimo oggetto.

Messaggi di risposta Il server invia dei **messaggi di risposta**, nella cui intestazione specifica alcuni campi ASCII tra cui il codice di stato:

- `200 OK`: la richiesta è stata completata con successo, ad esempio:
  ```
  HTTP/1.1 200 OK
  ```
- `301 Mover Permanently`: l'oggetto richiesto è stato spostato in una nuova posizione;

[1] Si veda la sezione 14.1.2.

- `400 Bad Request`: il formato del messaggio non è stato riconosciuto;
- `404 Not Found`: l'oggetto richiesto non è stato trovato;
- `505 HTTP Version Not Supported`: la versione HTTP non è supportata.

15.1.3 Cookie

L'HTTP è **stateless**: il server non conserva alcuna informazione relativa alle passate richieste da parte del client.

Un cookie è una stringa di testo salvato dal browser sull'host client che contiene alcune informazioni relative alla sessione utente (nome utente, lingua personalizzata), in modo che a ogni successiva visita alla stessa sezione del web il client la rimandi al server e quest'ultimo possa ricordare le operazioni svolte in passato dall'utente senza dover ricominciare da zero.

15.1.4 Server proxy

Il traffico di una rete locale può passare attraverso un **server proxy**,[2] che è dotato di una cache in cui salvare le ultime risposte HTTP avvenute $\Rightarrow$ si riduce il tempo di risposta nel caso in cui più client collegati al server proxy richiedano lo stesso oggetto, evitando di sovraccaricare il server di origine. Il server proxy, prima di consegnare la copia in cache al client, verifica con una richiesta HEAD al server di origine che l'oggetto non sia stato modificato nel frattempo: l'intestazione del messaggio di risposta contiene il campo `Last-Modified`, che informa sulla data di ultimo aggiornamento dell'oggetto. Anche il browser dell'utente può avere una cache locale.

15.2 Protocollo FTP

Il **protocollo File Transfer Protocol** (FTP) è ottimizzato per il trasferimento di file. Usa il TCP. L'host lato server non deve essere una macchina attrezzata appositamente a fare da server, ma può essere una macchina qualunque.

Al contrario dell'HTTP, l'FTP è **out of band**:[3] esiste una connessione dedicata alle **informazioni di controllo**, e il trasferimento di ogni file richiede l'apertura e la chiusura di una connessione TCP separata dalla connessione di controllo.

Anche nell'FTP i comandi e le risposte sono di tipo ASCII. Tra i comandi:

- `LIST` restituisce l'elenco dei file nel direttorio corrente;
- `RETR nome_file`: recupera un file;
- `STOR nome_file`: archivia un file in un host remoto.

15.3 Posta elettronica

1. l'**utente mittente** contatta il suo server di posta e gli invia l'e-mail;
2. l'e-mail, salvata nella mailbox del **server mittente**, entra in una coda[4] di messaggi da inviare non appena il server mittente riesce a contattare il server destinatario;
3. l'e-mail viene ricevuta dal **server destinatario** e salvata nella mailbox del server;
4. quando l'**utente destinatario** accederà alla posta troverà l'e-mail.

[2] L'utente deve specificare manualmente al browser l'indirizzo del server proxy.
[3] Si veda la sezione 1.6.1.
[4] Non è una vera e propria coda.

15.3.1 Protocollo SMTP

Il **protocollo Simple Mail Transfer Protocol** (SMTP) serve per l'invio di posta elettronica al server di posta. Utilizza il TCP (tipicamente sulla well-known port 25).

Un **relay mail server** è un server di posta configurato per accettare la posta proveniente da qualsiasi indirizzo IP, in modo che l'utente titolare dell'account di posta possa accedere alla propria posta da qualsiasi postazione Internet. L'RFC del protocollo SMTP tuttavia sconsiglia i relay server come misura anti-spam; inoltre il contenuto del comando HELO viene spesso confrontato con una blacklist.

Esempio di conversazione

I server comunicano tra di loro tramite comandi e risposte testuali ASCII (caratteri ASCI a 7 bit) ⇒ vantaggioso per il debug, ma svantaggioso per lo spreco di byte.

```
# Handshaking (presentazione)
C:  <apre la connessione>
S:  220 nome_server_B   # la conversazione inizia sempre con il codice 220
C:  HELO nome_server_A   # comunica il proprio nome
S:  250   # il nome è stato ricevuto
C:  MAIL FROM: <indirizzo_email_mittente>
S:  250   # conferma del mittente
C:  RCPT TO: <indirizzo_email_destinatario>
S:  250   # conferma del destinatario

# Trasferimento del messaggio
C:  DATA   # voglio iniziare a scrivere il messaggio
S:  354   # scrivi pure
C:  To: indirizzo_email_destinatario
C:  From: indirizzo_email_mittente
C:  Subject: oggetto_messaggio
C:  # una riga vuota, e più precisamente il carattere CRLF, separa
                                        l'intestazione dal corpo del messaggio
C:  corpo_messaggio
C:  .   # il punto sta a significare la fine del messaggio
S:  250   # messaggio accettato

# Chiusura della sessione
C:  QUIT   # voglio chiudere la connessione
S:  221
C:  <chiude la connessione>
```

Si noti che, poiché le informazioni inviate nella fase di handshaking non vengono mantenute, l'indirizzo del mittente e quello del destinatario vengono riportati anche nel testo stesso del messaggio.

Allegati

L'RFC originale del protocollo SMTP prevedeva solamente lo scambio di testo. L'evoluzione di questo RFC permette anche lo scambio di file grazie al **Multimedia Mail Extension** (MIME), cioè delle righe aggiuntive nell'intestazione del messaggio che dichiarano il tipo di contenuto

MIME del file.

Esempio: invio di un'immagine JPEG

```
...
C:  To: indirizzo_email_destinatario
C:  From: indirizzo_email_mittente
C:  Subject: oggetto_messaggio
C:  MIME-Version: 1.0
C:  Content-Transfer-Encoding: base64
C:  Content-Type: image/jpeg
C:
C:  bit_immagine   # corpo del messaggio
```

15.3.2 Protocollo POP3

Il **protocollo Post Office Protocol version 3** (POP3) serve per il recupero della posta elettronica dal server di posta. Usa il TCP.

Il protocollo Internet Mail Access Protocol (IMAP) migliora il protocollo POP3, che è stateless.

Esempio di conversazione

Ipotesi

- ci sono 2 messaggi di posta non letti sul server;
- non si utilizza il protocollo di sicurezza SSL.

```
# Autorizzazione
C:  <apre la connessione>
S:  +OK   # server POP3 pronto
C:  user nome_utente
S:  +OK   # il server può rispondere +OK o -ERR
C:  pass password   # la password è in chiaro ⇒ serve l'SSL
S:  +OK   # accesso con successo

# Transazione
C:  list   # quanti nuovi messaggi?
S:  1 498
S:  2 912
...
S:  .   # messaggi finiti

C:  retr 1   # chiede il contenuto del primo messaggio
S:  contenuto_messaggio
S:  .   # il contenuto del messaggio è terminato
C:  dele 1   # ordina di eliminare il messaggio dal server

C:  retr 2   # chiede il contenuto del secondo messaggio
S:  contenuto_messaggio
S:  .   # il contenuto del messaggio è terminato
C:  dele 2   # ordina di eliminare il messaggio dal server

# Chiusura della sessione
C:  quit   # voglio chiudere la connessione
```

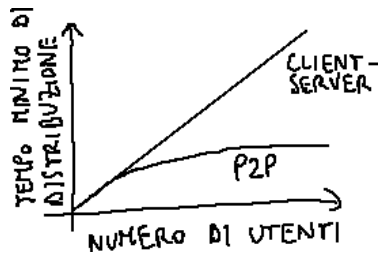
```
S:  +OK
C:  <chiude la connessione>
```

Il comportamento predefinito del client di posta, previsto dall'RFC del POP3, è quello di ordinare l'eliminazione del messaggio subito dopo averlo ricevuto dal server; l'utente può configurare il proprio client di posta affinché il messaggio sia mantenuto sul server.

15.4 Reti P2P

Nelle **reti P2P** non esiste un server sempre acceso al quale tutti si collegano per ricevere i file: i nodi di una rete P2P, chiamati **peer**, fungono sia da client sia da server, e sono connessi in modo intermittente.

Più peer stanno condividendo un file, più le prestazioni di distribuzione migliorano, perché un peer mentre sta scaricando il file può nel frattempo condividere la parte scaricata con un altro peer, mentre nel mondo client-server la banda dell'unico server è limitata:



15.4.1 Protocollo BitTorrent

I file sono suddivisi in parti (chunk) da 256 KB. Il **torrent** è il gruppo di peer che sta condividendo i chunk di un file; la rete P2P è chiamata **rete overlay**. Il **tracker** è il server che gestisce la distribuzione di un certo file tra i peer. Un host si collega periodicamente al tracker, il cui indirizzo è specificato in un file .torrent, e il tracker fornisce all'host una lista composta da un numero massimo di indirizzi IP di peer a cui l'host potrà collegarsi per recuperare il file.

Tra i peer si distinguono i **seed**, cioè gli host che possiedono l'intero file. I seed possono altruisticamente rimanere collegati alla rete oppure egoisticamente abbandonarla: il **churn**, cioè il tasso di abbandono, influisce molto sulle prestazioni della rete.

L'host può scambiare i chunk del file con i peer che il tracker gli ha comunicato.

Richiesta di chunk L'host chiede periodicamente a ogni peer una lista dei chunk che possiedono, quindi applica un algoritmo **local rarest first**: richiede il chuck mancante più raro nel vicinato, cioè quello che è posseduto da meno peer.

Invio di chunk Tra quelli che stanno richiedendo chunk, periodicamente l'host sceglie (choke) i 4 peer da cui è riuscito a scaricare a una velocità più alta nel passato, e inizia a mandare loro dei chunk. Questo meccanismo, chiamato **tit-for-tat**, serve per svantaggiare i freerider, cioè gli host che scaricano senza condividere. Per evitare di contattare sempre gli stessi peer, si applica l'**optimistically unchoke**: ogni 30 secondi l'host sceglie in modo casuale un altro peer e inizia a inviargli dei chunk, con la speranza di essere in futuro scelto da quel peer in modo da poter ricevere dei chunk che i soliti peer non hanno.

15.4.2 DHT

Napster, la prima rete P2P, era costituito da un unico tracker che serviva l'intera rete ⇒ scollegando il server fu facile smantellare la rete per motivi legali. Le reti P2P si stanno sempre più orientando verso il **Distributed Hash Table** (DHT): il database P2P, anziché essere concentrato in singoli tracker, è distribuito tra i peer stessi. Il database è organizzato per coppie (*chiave*,

valore) (ad esempio *chiave* = titolo del film, *valore* = indirizzo IP). Un peer può interrogare il database distribuito facendo una ricerca per chiave, e può anche inserire una coppia (*chiave*, *valore*).

Le coppie (*chiave*, *valore*) vengono assegnate ai peer tramite una tabella di hash:[5]

1. ogni chiave viene convertita in un id intero da una funzione di hash;
2. ad ogni peer viene assegnato un id intero (per esempio una funzione di hash converte l'indirizzo IP del peer);
3. la coppia viene assegnata al peer il cui id è pari o, nel caso non sia esistente, immediatamente successivo all'id della chiave; se viene superato il massimo id presente nella rete, la coppia viene assegnata al peer di id più basso.

Una rete P2P con DHT è **strutturata**, cioè il grafo della rete non è costruito in modo casuale.

Chord

La rete **Chord** ha una topologia circolare: ogni peer, oltre a conoscere il proprio id, conosce solo gli indirizzi IP del suo immediato precedessore e del suo immediato successore $\Rightarrow$ la ricerca di una chiave percorre la catena di peer fino ad arrivare al peer il cui id è pari o immediatamente successivo all'id della chiave cercata.

La complessità di questo meccanismo di ricerca è alta ($O(n)$) rispetto al sistema con il tracker ($O(1)$).

Grazie ai **shortcut**, cioè delle "scorciatoie" verso i peer distanti un numero di hop potenza del 2, le prestazioni migliorano a $O(\log n)$.

Kademlia

La rete Kademlia di eMule è invece basata su una topologia ad albero binario:[6] ha prestazioni $O(\log n)$ senza dover ricorrere a shortcut.

[5] Si veda il capitolo "Le tabelle di hash" negli appunti di *Algoritmi e programmazione*.
[6] Si veda il capitolo "Alberi binari di ricerca (BST)" negli appunti di *Algoritmi e programmazione*.

Capitolo 16

NAT

Il **Network Address Translation** (NAT) è un modo per connettere ad Internet più host utilizzando un unico indirizzo IP. A ogni host viene assegnato un **indirizzo privato**, e tutti gli host sono collegati a un router che ha invece un indirizzo pubblico.

Il router modifica a livello 3 ogni pacchetto che vi transita, sia in invio sia in ricezione:

- invio verso l'esterno: il router converte l'indirizzo IP sorgente, privato, nell'indirizzo pubblico, e memorizza nella NAT translation table la coppia indirizzo IP sorgente privato e indirizzo IP di destinazione;
- ricezione dall'esterno: il router cerca nella NAT translation table l'indirizzo IP sorgente del pacchetto per trovare quale host interno aveva contattato quella destinazione, e quindi converte l'indirizzo IP di destinazione del pacchetto nell'indirizzo interno privato.

Per consentire a più host interni di contattare la stessa destinazione, il router può implementare il **Port Address Translation** (PAT): gli host aprono le connessioni alla destinazione su porte TCP/UDP diverse, e la NAT translation table memorizza anche la coppia di porte, in modo che alla ricezione di pacchetti il router possa distinguere tra i vari host in base alla porta utilizzata. Un riconoscimento basato su indirizzo MAC non permetterebbe a un host di aprire più connessioni TCP/UDP verso la stessa destinazione.

16.1 NAT traversal

In presenza di NAT alcune applicazioni possono presentare delle problematiche legate alla limitata raggiungibilità dall'esterno:

- **port forwarding** (usato per le applicazioni P2P): tutto il traffico che giunge a una certa porta viene inviato (forwarding) a un indirizzo IP privato configurato staticamente;
- **relaying** (usato da Skype): il client esterno si connette a un relay server di proprietà Skype, che a sua volta invia tutto il traffico al client dentro il NAT $\Rightarrow$ l'utente non deve configurare manualmente la porta sul router.