# Appunti del corso di
# Reti di elaboratori

## PROF. G. BONGIOVANNI

## *Premessa*

Questi appunti sono basati sul libro "Computer Networks" di A. Tanenbaum, terza edizione, ed. Prentice-Hall, adottato quale libro di testo del corso.

Essi rispecchiano piuttosto fedelmente il livello di dettaglio che viene seguito durante le lezioni, e costituiscono un ausilio didattico allo studio.

Tuttavia, è importante chiarire che gli appunti non vanno intesi come sostitutivi né del libro di testo né della frequenza alle lezioni, che rimangono fattori importanti per una buona preparazione dell'esame.

La realizzazione di questi appunti è stata resa possibile dalla collaborazione dello studente Federico Neri, che ha avuto la pazienza di convertire in forma elettronica il contenuto testuale dei manoscritti da me preparati per il corso. La realizzazione delle figure, la formattazione e la rifinitura del testo sono opera mia.

# 1) Introduzione

Gli ultimi tre secoli sono stati dominati ciascuno da una diversa tecnologia che lo ha caratterizzato ed ha avuto profonde influenze sulla vita dell'uomo:

- 18º secolo: sistemi meccanici, rivoluzione industriale;
- 19º secolo: motori a vapore;
- 20º secolo: tecnologie dell'informazione:
    - raccolta e memorizzazione;
    - elaborazione;
    - distribuzione.

Nel nostro secolo si sono via via diffusi:

- il sistema telefonico, a livello mondiale;
- la radio e la televisione;
- i computer;
- i satelliti per telecomunicazioni.

Queste tecnologie stanno rapidamente convergendo. In particolare, la combinazione di elaboratori e sistemi di telecomunicazione ha avuto una profonda influenza sull'organizzazione dei sistemi di calcolo.

Si è passati dal vecchio modello ***mainframe - terminali*** , in cui la potenza di calcolo è concentrata in un unico grande elaboratore a cui si accede per mezzo di un certo numero di terminali, a quello attuale in cui vi è un grande numero di elaboratori ***autonomi***, ***interconnessi*** fra loro:

- autonomi: significa che non deve esserci fra loro una relazione tipo master/slave (ad es., l'uno non può forzare lo spegnimento dell'altro);
- interconnessi: significa che devono essere capaci di scambiare informazioni (sfruttando un opportuno mezzo fisico).

Un sistema di calcolo siffatto è detto ***rete di elaboratori*** (***computer network***).

Si noti che rete di elaboratori non è sinonimo di ***sistema distribuito*****.** Infatti:

- in un sistema distribuito l'esistenza di più elaboratori è invisibile all'utente, che ha l'impressione di avere a che fare con un unico sistema di calcolo;
- in una rete di elaboratori, l'utente è conscio dell'esistenza di molteplici elaboratori, che devono essere esplicitamente riferiti.

In effetti, si può dire che:

**Rete di Elaboratori + Sistema software di gestione = Sistema distribuito**

dove il sistema software di gestione altro non è che un particolare tipo di sistema operativo, ossia un ***sistema operativo distribuito***.

## *1.1) Usi delle reti di elaboratori*

Moltissimi sono gli usi delle reti di elaboratori, sia per le organizzazioni che per i singoli individui.

- Per le organizzazioni:
  - condivisione risorse: si possono rendere disponibili a chiunque programmi e informazioni anche distanti migliaia di km;
  - affidabilità: si ottiene mettendo in rete sorgenti alternative delle risorse (ad es. duplicando le applicazioni e i dati su più computer). E' importante in sistemi che devono funzionare a tutti i costi (traffico aereo, centrali nucleari, sistemi militari, ecc.);
  - diminuzione dei costi: una rete di personal computer costa molto meno di un mainframe. A volte alcuni elaboratori sono più potenti ed offrono agli altri dei servizi (modello ***client-server***, vedi figura sottostante);
  - scalabilità: si possono aumentare le prestazioni del sistema aumentando il numero di elaboratori (entro certi limiti);
  - comunicazione fra persone: è possibile inviarsi messaggi, scambiarsi file, ecc.

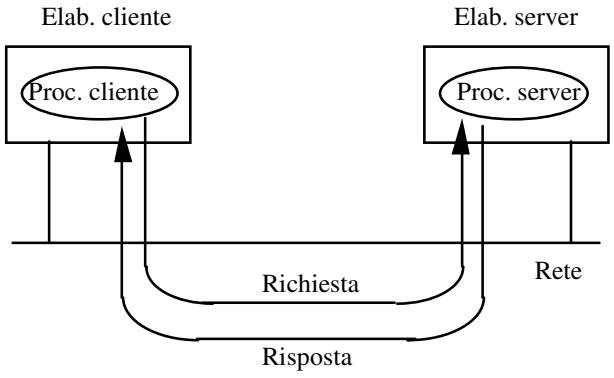


**Figura 1-1**: Il modello client-server

- Per i singoli individui: (di solito da casa propria tramite "fornitori di accesso"):
  - accesso ad informazioni remote, ad es.:

    - accesso a servizi bancari;
    - acquisti da casa;
    - navigazione sul World Wide Web;
- comunicazioni fra persone:
    - posta elettronica;
    - videoconferenza;
    - gruppi di discussione;
- divertimento:
    - video on demand (selezione e ricezione via rete di un qualunque spettacolo tratto da un catalogo);
    - giochi interattivi (contro macchine o avversari umani).

## *1.2) Aspetti hardware delle reti*

Due parametri sono utili per definire le caratteristiche di una rete, anche se non esiste una tassonomia universalmente accettata:

- tecnologia trasmissiva;
- scala dimensionale.

### 1.2.1) Tecnologia trasmissiva

Ci sono due tipologie per quanto riguarda la tecnologia trasmissiva:

- ***reti broadcast***;
- ***reti punto a punto***.

Le ***reti broadcast*** sono dotate di un unico "canale" di comunicazione che è condiviso da tutti gli elaboratori. Brevi messaggi (spesso chiamati ***pacchetti***) inviati da un elaboratore sono ricevuti da tutti gli altri elaboratori. Un indirizzo all'interno del pacchetto specifica il destinatario.

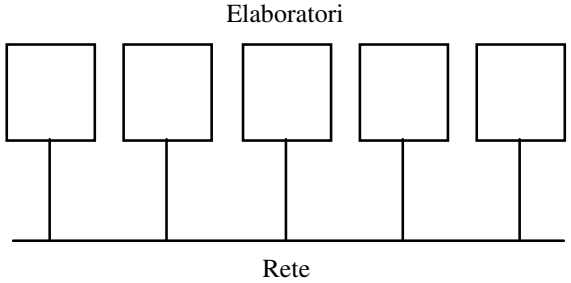


**Figura 1-2**: una rete broadcast

Quando un elaboratore riceve un pacchetto, esamina l'indirizzo di destinazione; se questo coincide col proprio indirizzo il pacchetto viene elaborato, altrimenti viene ignorato.

Le reti broadcast, in genere, consentono anche di inviare un pacchetto a tutti gli altri elaboratori, usando un opportuno indirizzo (***broadcasting***). In tal caso tutti prendono in considerazione il pacchetto.

Un' altra possibilità è inviare il pacchetto ad un sottoinsieme degli elaboratori (***multicasting***). In tal caso solo gli elaboratori di tale sottoinsieme lo prendono in considerazione, gli altri lo ignorano. In questo caso, un bit dell'indirizzo indica che si tratta di una trasmissione in multicasting. I rimanenti (n-1) bit dell' indirizzo rappresentano l'indirizzo del gruppo destinatario.

Le ***reti punto a punto*** consistono invece di un insieme di connessioni fra coppie di elaboratori.

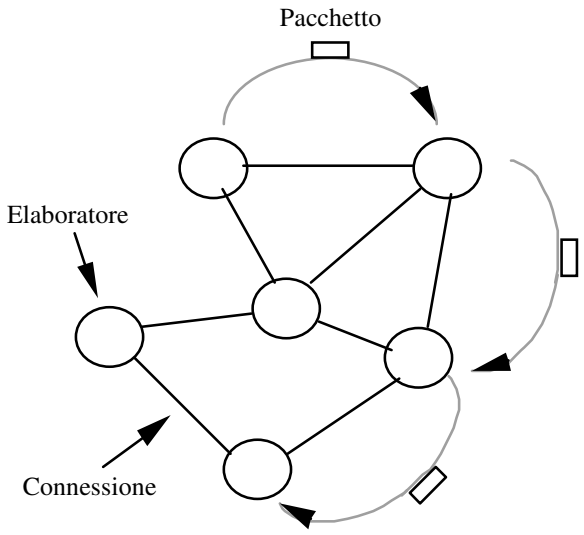


**Figura 1-3**: una rete punto a punto

Per arrivare dalla sorgente alla destinazione, un pacchetto può dover attraversare uno o più elaboratori intermedi. Spesso esistono più cammini alternativi, per cui gli algoritmi di ***instradamento*** (***routing***) hanno un ruolo molto importante.

In generale (ma con molte eccezioni):
- le reti geograficamente localizzate tendono ad essere broadcast;
- le reti geograficamente molto estese tendono ad essere punto a punto.

Alcune eccezioni:
- rete geografica realizzata via satellite (e quindi broadcast);
- rete locale basata su ATM (e quindi punto a punto).

### 1.2.2) Scala dimensionale

Un criterio alternativo di classificazione è la scala dimensionale delle reti. In questo contesto si distingue fra ***reti locali***, ***reti metropolitane*** e ***reti geografiche***.

| Distanza fra processori | Ambito | Tipo di rete |
|---|---|---|
| | | |
| 10 m. | Stanza | Rete locale |
| 100 m. | Edificio | Rete locale |
| 1 km. | Campus | Rete locale |
| 10 km. | Città | Rete metropolitana |
| 100 km. | Nazione | Rete geografica |
| 1000 km. | Continente | Rete geografica |
| 10.000 km. | Pianeta | Internet (Rete geografica) |

La distanza è un fattore molto importante, poiché a differenti scale dimensionali si usano differenti tecniche.

### 1.2.3) Reti locali

Le reti locali (***Local Area Network, LAN***), in genere:
- sono possedute da una organizzazione (reti private);
- hanno un'estensione che arriva fino a qualche km;
- si distendono nell'ambito di un singolo edificio o campus (non si possono, di norma, posare cavi sul suolo pubblico);
- sono usatissime per connettere PC o workstation.

Esse si distinguono dagli altri tipi di rete per tre caratteristiche:
- ***dimensione***: la dimensione non può andare oltre un certo limite, per cui è noto a priori il tempo di trasmissione nel caso peggiore. Questa conoscenza permette di utilizzare delle tecniche particolari per la gestione del canale di comunicazione;

- ***tecnologia trasmissiva***: come già accennato, le LAN sono in generale reti broadcast. Velocità di trasmissione tipiche sono da 10 a 100 Mbps (megabit al secondo, cioé milioni di bit al secondo), con basso ritardo di propagazione del segnale da un capo all'altro del canale (qualche decina di microsecondi) e basso tasso di errore;
- ***topologia***: sono possibili diverse topologie, le più diffuse sono il ***bus*** ed il ***ring***;
    - topologia bus:
        - in ogni istante solo un elaboratore può trasmettere, gli altri devono astenersi;
        - è necessario un meccanismo di ***arbitraggio*** per risolvere i conflitti quando due o più elaboratori vogliono trasmettere contemporaneamente;
        - l'arbitraggio può essere centralizzato o distribuito;
        - lo standard ***IEEE 802.3*** (chiamato impropriamente ***Ethernet***) è per una rete broadcast, basata su un bus, con arbitraggio distribuito, operante a 10 oppure 100 Mbps;
        - gli elaboratori trasmettono quando vogliono; se c'è una collisione aspettano un tempo casuale e riprovano;
    - topologia ring:
        - in un ring ogni bit circumnaviga l'anello in un tempo tipicamente inferiore a quello di trasmissione di un pacchetto;
        - anche qui è necessario un meccanismo di arbitraggio (spesso basato sul possesso si un ***gettone*** (***token***) che abilita alla trasmissione);
        - lo standard ***IEEE 802.5*** (derivante dalla rete IBM ***Token Ring***) è una rete broadcast basata su ring, con arbitraggio distribuito, operante a 4 o 16 Mbps.

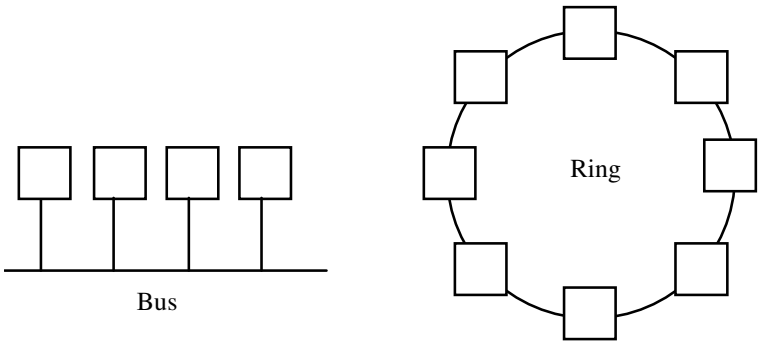


**Figura 1-4**: topologie bus e ring

Infine le reti broadcast possono essere classificate a seconda del meccanismo scelto per l'arbitraggio:

- ***Allocazione statica***: le regole per decidere chi sarà il prossimo a trasmettere sono fissate a priori, ad esempio assegnando un time slot ad ogni elaboratore con un algoritmo round-robin. Lo svantaggio è rappresentato dallo spreco dei time slot assegnati a stazioni che non devono trasmettere.

- ***Allocazione dinamica***: si decide di volta in volta chi sarà il prossimo a trasmettere; è necessario un meccanismo di arbitraggio delle contese, che può essere:
  - ***arbitraggio centralizzato***: un apposita apparecchiatura, ad esempio, una bus arbitration unit, accetta richieste di trasmissione e decide chi abilitare;
  - ***arbitraggio distribuito***: ognuno decide per conto proprio (come in 802.3); vedremo come si può evitare un prevedibile caos.

### 1.2.4) Reti metropolitane

Le reti metropolitane (***Metropolitan Area Network, MAN***)hanno un'estensione tipicamente urbana (quindi anche molto superiore a quella di una LAN) e sono generalmente pubbliche (cioé un'azienda, ad es. Telecom Italia, mette la rete a disposizione di chiunque desideri, previo pagamento di una opportuna tariffa).

Fino a qualche anno fa erano basate essenzialmente sulle tecnologie delle reti geografiche, utilizzate su scala urbana. Recentemente pero' e' stato definito un apposito standard, lo ***IEEE 802.6*** o ***DQDB*** (***Distributed Queue Dual Bus***), che e' effettivamente utilizzato in varie realizzazioni, molto piu' vicino alla tecnologia LAN che WAN.

Esiste un mezzo trasmissivo di tipo broadcast (due bus in 802.6) a cui tutti i computer sono attaccati.

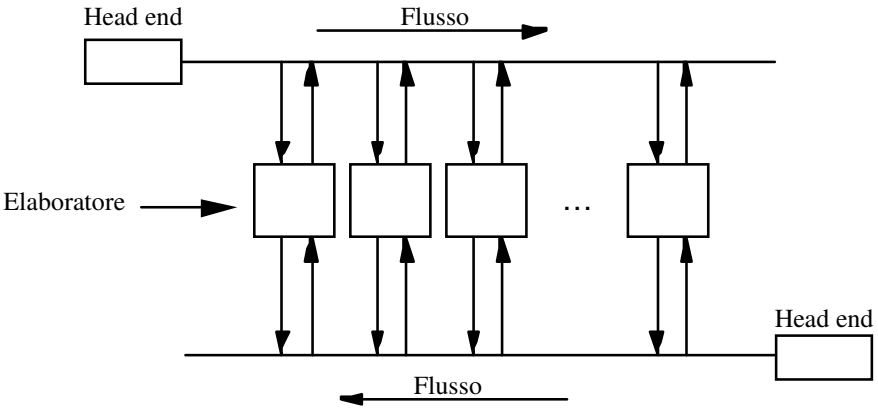


**Figura 1-5**: Distributed Queue Dual Bus

Ogni bus (cavo coassiale o fibra ottica) e' unidirezionale, ed ha una ***head-end*** che cadenza l'attivita' di trasmissione.

### 1.2.5) Reti geografiche

Le reti geografiche (***Wide Area Network, WAN***) si estendono a livello di una nazione, di un continente o dell'intero pianeta. Una WAN è tipicamente costituita di due componenti distinte:

- un insieme di elaboratori (***host*** oppure ***end system***) sui quali girano i programmi usati dagli utenti;
- una ***comunication subnet*** (o ***subnet***), che connette gli end system fra loro. Il suo compito è trasportare messaggi da un end system all'altro, così come il sistema telefonico trasporta parole da chi parla a chi ascolta.

Di norma la subnet consiste, a sua volta, di due componenti:

- ***linee di trasmissione*** (dette anche ***circuiti***, ***canali***, ***trunk***):
- ***elementi di commutazione*** (***switching element***): gli elementi di commutazione sono elaboratori specializzati utilizzati per connettere fra loro due o più linee di trasmissione. Quando arrivano dati su una linea, l'elemento di commutazione deve scegliere una linea in uscita sul quale instradarli. Non esiste una terminologia standard per identificare gli elementi di commutazione. Termini usati sono:
  - ***sistemi intermedi***;
  - ***nodi di commutazione pacchetti***;
  - ***router*** (quello che utilizzeremo noi).

Una tipica WAN è utilizzata per connettere più LAN fra loro:

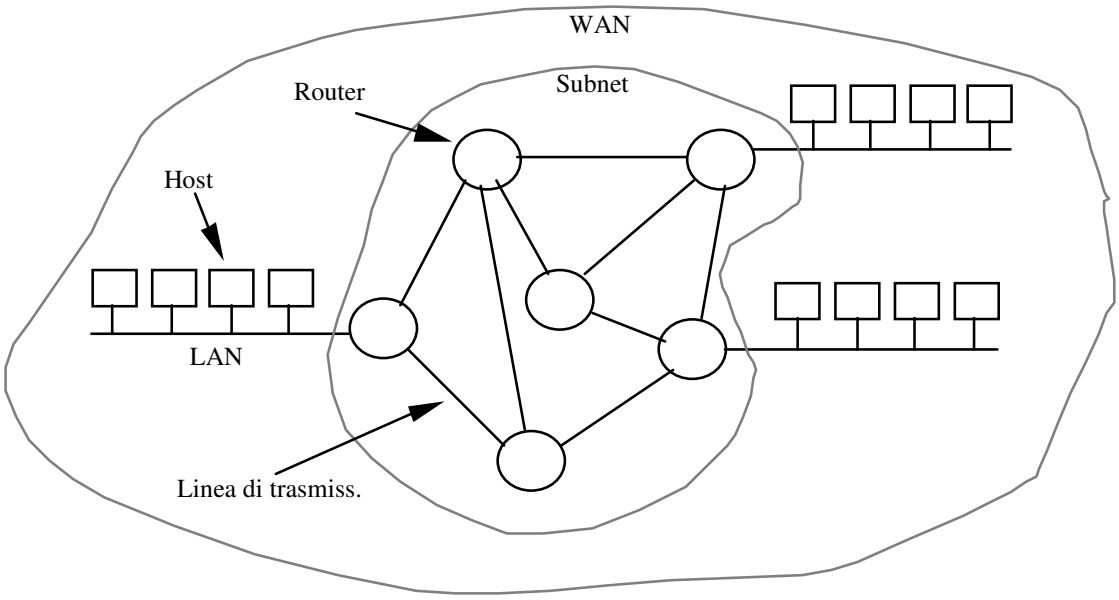


**Figura 1-6**: struttura tipica di una WAN

In generale una WAN contiene numerose linee (spesso telefoniche) che congiungono coppie di router.

Ogni router, in generale, deve:
1. ricevere un pacchetto da una linea in ingresso;
2. memorizzarlo per intero in un buffer interno;
3. appena la necessaria linea in uscita è libera, instradare il pacchetto su essa.

Una subnet basata su questo principio si chiama:
- ***punto a punto***;
- ***store and forward***;
- ***a commutazione di pacchetto (packet switched)***.

Molte topologie di interconnessione possono essere impiegate fra i router:
- ***a stella*** (ridondanza zero);
- ***ad anello*** (ridondanza zero);
- ***ad albero*** (ridondanza zero);
- ***magliata*** (ridondanza media);
- ***completamente connessa*** (ridondanza massima).

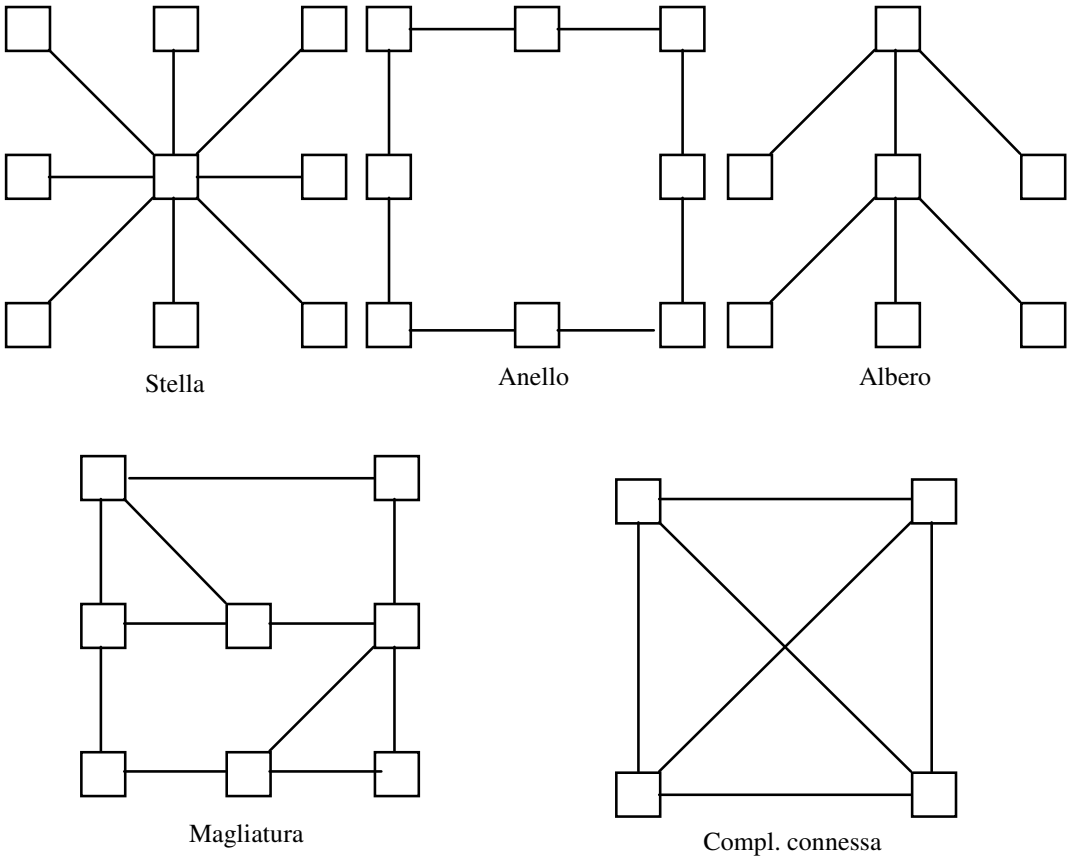


**Figura 1-7**: topologie di interconnessione

Un'altra possibilità è una WAN basata su satellite oppure radio al suolo.

- ***Satellite***: ogni router sente l'output del satellite e si fa sentire dal satellite. Dunque, in generale si ha:
    - broadcast downlink (cioé dal satellite a terra);
    - broadcast uplink (cioé da terra al satellite) se i router possono "sentire" quelli vicini, point to point altrimenti.

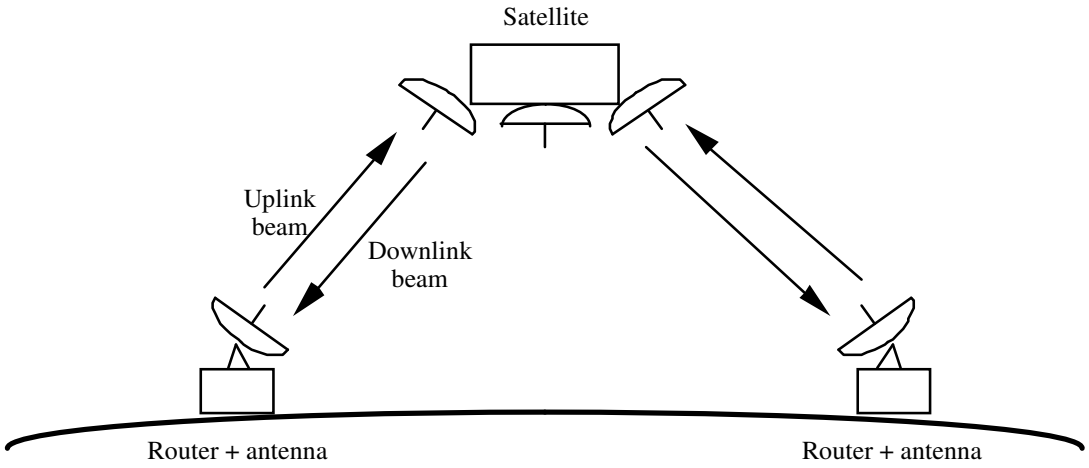


**Figura 1-8**: interconnessione di router via satellite

- ***Radio al suolo***: ogni router sente l'output dei propri vicini (entro una certa distanza massima):
    - anche qui siamo in presenza di una rete broadcast.

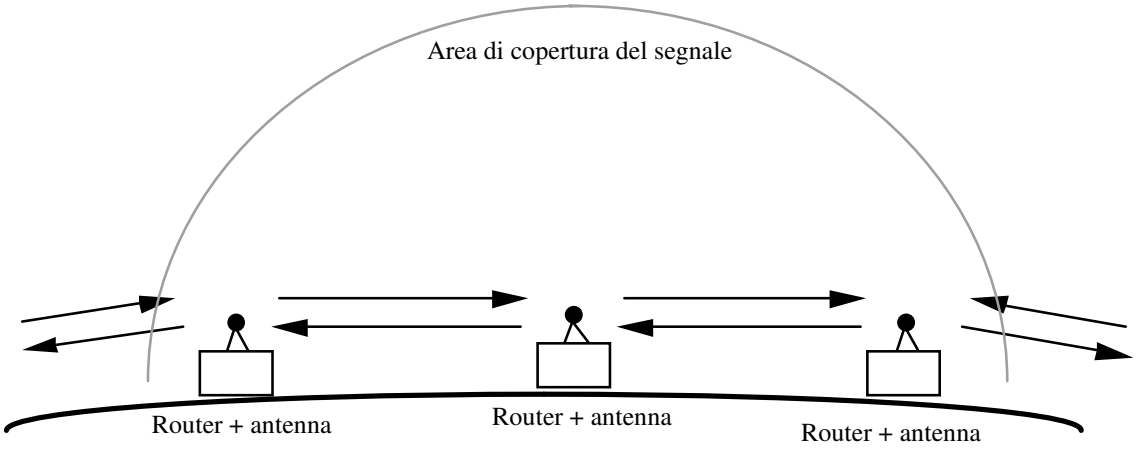


**Figura 1-9**: interconnessione di router via radio al suolo

Una WAN può essere anche realizzata in maniera mista: in parte cablata, in parte basata su radio o satellite.

### 1.2.6) Interconnessione di reti (Internetwork)

Una ***internetwork*** è formata quando reti diverse (sia LAN che MAN o WAN) sono collegate fra loro.

A prima vista, almeno in alcuni casi, la cosa è apparentemente uguale alla definizione di WAN vista precedentemente (se al posto di subnet si scrive WAN, abbiamo una internetwork costituita da una WAN e quattro LAN).

Alcuni problemi però sorgono quando si vogliono connettere fra di loro reti progettualmente diverse (spesso incompatibili fra loro). In questo caso si deve ricorrere a speciali attrezzature, dette ***gateway*** (o ***router multiprotocollo***), che oltre ad instradare i pacchetti da una rete all'altra, effettuano le operazioni necessarie per rendere possibili tali trasferimenti.

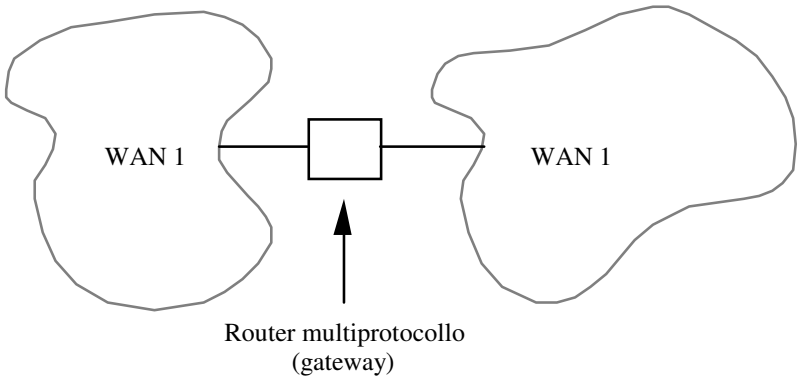


**Figura 1-10**: interconnessione di reti

Nel contesto del corso utilizzeremo:
- ***internet*** come sinonimo di internetwork, cioé la interconnessione di più reti generiche;
- ***Internet*** (con la I maiuscola) per riferirci alla specifica internetwork, basata su TCP/IP, che ormai tutti conoscono.

C'è molta confusione sui termini:
- ***sottorete (subnet)***, che nel contesto di una WAN è l'insieme dei router e delle linee di trasmissione;
- ***rete (network)***, che altro non è che una subnet più tutti gli host collegati;
- ***internetwork***, che è una collezione di più network, anche non omogenee, collegate per mezzo di gateway.

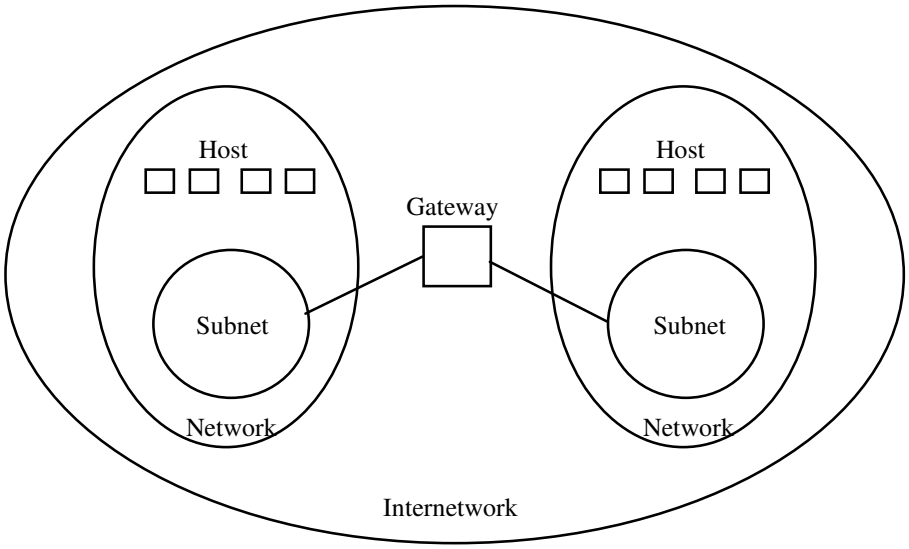


**Figura 1-11**: relazioni fra subnet, network e internetwork

## *1.3) Aspetti software delle reti*

Le prime reti furono progettate cominciando dall'hardware e sviluppando il software solo successivamente, quasi come se esso fosse un'accessoria appendice dell'hardware.

Questo approccio non funziona più. Il SW di rete è oggi altamente strutturato. Esaminiamo ora, a grandi linee, tale strutturazione, che servirà da guida per l'intero corso e sulla quale torneremo spesso.

### 1.3.1) Gerarchie di protocollo

Per ridurre la complessità di progetto, le reti sono in generale organizzate a ***livelli***, ciascuno costruito sopra il precedente. Fra un tipo di rete ed un altra, possono essere diversi:

- il numero di livelli;
- i nomi dei livelli;
- il contenuto dei livelli;
- le funzioni dei livelli.

Comunque un principio generale è sempre rispettato:

- lo scopo di un livello è offrire certi ***servizi*** ai livelli più alti, nascondendo i dettagli sul come tali servizi siano implementati.

Il livello n su un host porta avanti una conversazione col livello n su di un'altro host. Le regole e le convenzioni che governano la conversazione sono collettivamente indicate col termine di ***protocollo di livello n***.

Le entità (processi) che effettuano tale conversazione si chiamano ***peer entitiy (entità di pari livello)***.

Il dialogo fra due peer entity di livello n viene materialmente realizzato tramite i servizi offerti dal livello (n-1).

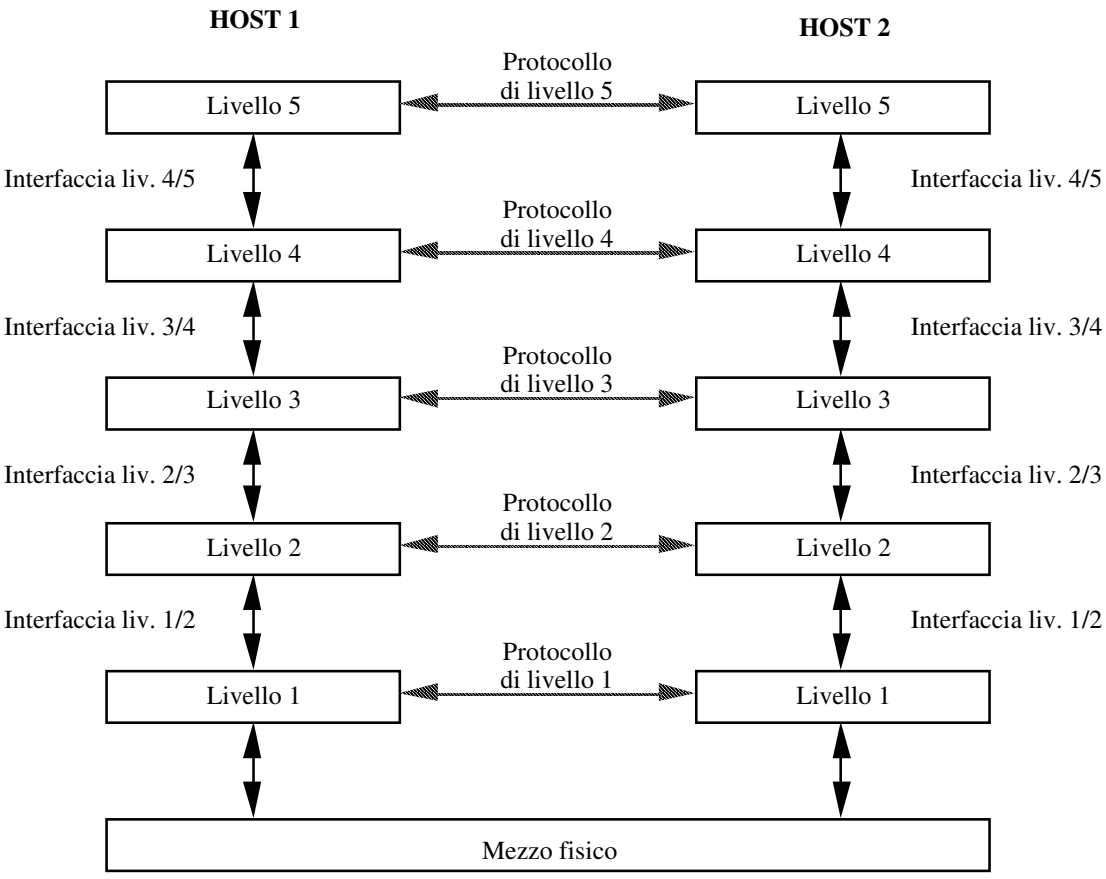


**Figura 1-12**: Dialogo fra peer entity

- In realtà non c'è un trasferimento diretto dal livello n di host 1 al livello n di host 2. Ogni livello di host 1 passa i ***dati***, assieme a delle ***informazioni di controllo***, al livello sottostante.
- Al di sotto del livello 1 c'è il mezzo fisico, attraverso il quale i dati vengono trasferiti da host 1 ad host 2.
- Quando arrivano a host 2, i dati vengono passati da ogni livello (a partire dal livello 1) a quello superiore, fino a raggiungere il livello n.

Fra ogni coppia di livelli adiacenti è definita una ***interfaccia***, che caratterizza:

- le operazioni primitive che possono essere richieste al livello sottostante;
- i servizi che possono essere offerti dal livello sottostante.

I vantaggi di una buona progettazione delle interfacce sono:

- minimizzazione delle informazioni da trasferire;
- possibilità di modificare l'implementazione del livello (ad es., ove le linee telefoniche venissero sostituite da canali satellitari) con una più attuale che offra gli stessi servizi.

### 1.3.2) Architettura di rete

L'insieme dei livelli e dei relativi protocolli è detto ***architettura di rete***.

La specifica dell' architettura deve essere abbastanza dettagliata da consentire la realizzazione di SW e/o HW che, per ogni livello, rispetti il relativo protocollo.

Viceversa, i dettagli implementativi di ogni livello e le interfacce fra livelli non sono parte dell'architettura, in quanto sono nascosti all'interno di un singolo host.

E' quindi possibile che sui vari host della rete ci siano implementazioni che differiscono fra di loro anche in termini di interfacce fra livelli, purché ogni host implementi correttamente i protocolli previsti dall'architettura. In questo caso possono dialogare fra loro anche host aventi caratteristiche (processore, sistema operativo, costruttore) diverse.

Dunque, nell'ambito di una specifica architettura di rete, si ha che:

- tutti gli host devono contenere implementazioni conformi in termini di livelli e di protocolli;
- gli host possono contenere implementazioni che differiscono in termini di dettagli implementativi e di interfacce fra livelli;

Un'architettura di rete può essere:

- ***proprietaria***;
- ***standard de facto***;
- ***standard de iure***.

Un'architettura proprietaria è basata su scelte indipendenti ed arbitrarie del costruttore, ed è generalmente incompatibile con architetture diverse. Nel senso più stretto del termine è un'architettura per la quale il costruttore non rende pubbliche le specifiche, per cui nessun altro può produrre apparati compatibili.

Esempi:

- IBM SNA (System Network Architecture)
- Digital Decnet Phase IV;
- Novell IPX;
- Appletalk.

Un'architettura standard de facto è un'architettura basata su specifiche di pubblico dominio (per cui diversi costruttori possono proporre la propria implementazione) che ha conosciuto una larghissima diffusione.

Esempi:

- Internet Protocol Suite (detta anche architettura TCP/IP).

Un'architettura standard de iure è un'architettura basata su specifiche (ovviamente di pubblico dominio) approvate da enti internazionali che si occupano di standardizzazione. Anche in questo caso ogni costruttore può proporne una propria implementazione.

Esempi:

- standard IEEE 802 per le reti locali;
- architettura OSI (Open Systems Interconnection);
- Decnet Phase V (conforme allo standard OSI).

L'insieme dei protocolli utilizzati su un host e relativi ad una specifica architettura di rete va sotto il nome di ***pila di protocolli (protocol stack)***. Si noti che un host può avere contemporaneamente attive più pile di protocolli.

### 1.3.3) Funzionamento del software di rete

Per comprendere i meccanismi basilari di funzionamento del software di rete si può pensare alla seguente analogia umana, nella quale un filosofo indiano vuole conversare con uno stregone africano:

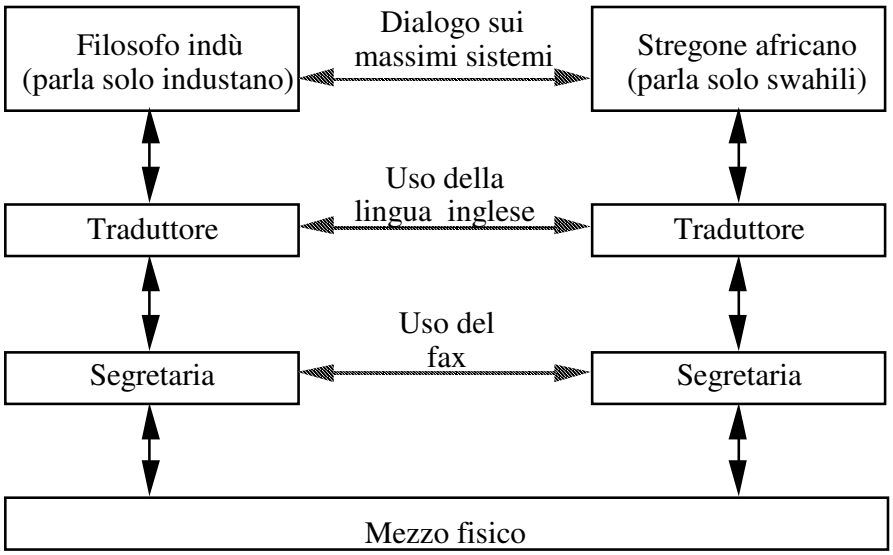


**Figura 1-13**: Dialogo fra grandi menti

Nel caso delle reti, la comunicazione fra le due entità di livello superiore avviene con una modalità che, almeno in linea di principio, è uguale in tutte le architetture di rete:

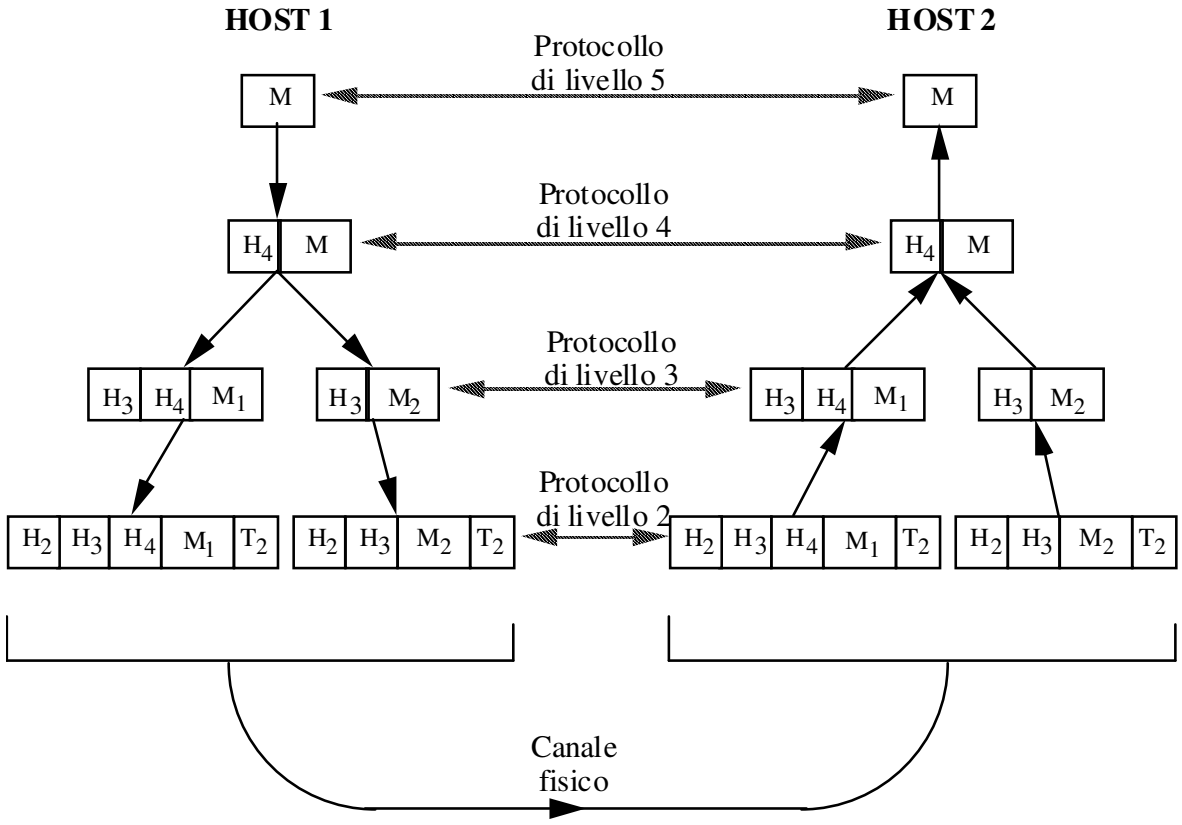


**Figura 1-14**: Flusso dell'informazione fra peer entity

Vediamo cosa accade:

1. il programma applicativo (livello 5) deve mandare un messaggio M alla sua peer entity;
2. il livello 5 consegna M al livello 4 per la trasmissione;
3. il livello 4 aggiunge un suo ***header*** in testa al messaggio (talvolta si dice che il messaggio è inserito nella ***busta*** di livello 4); questo header contiene informazioni di controllo, tra le quali:
    - numero di sequenza del messaggio;
    - dimensione del messaggio;
    - time stamp;
    - priorità;
4. il livello 4 consegna il risultato al livello 3;
5. il livello 3 può trovarsi nella necessità di frammentare i dati da trasmettere in unità più piccole, (***pacchetti***) a ciascuna delle quali aggiunge il suo header;

6. il livello 3 passa i pacchetti al livello 2;
7. il livello 2 aggiunge ad ogni pacchetto il proprio header (e magari un ***trailer***) e lo spedisce sul canale fisico;
8. nella macchina di destinazione i pacchetti fanno il percorso inverso, con ogni livello che elimina (elaborandoli) l'header ed il trailer di propria competenza, e passa il resto al livello superiore.

Aspetti importanti sono i seguenti:

- le peer entity pensano concettualmente ad una comunicazione orizzontale fra loro, basata sul protocollo del proprio livello, mentre in realtà comunicano ciascuna col livello sottostante attraverso l'interfaccia fra i due livelli;
- spesso i livelli bassi sono implementati in hardware o firmware (per ragioni di efficienza). Nonostante questo, spesso gli algoritmi di gestione sono complessi.

### 1.3.4) Interfacce e servizi

La funzione di ogni livello è di offrire servizi al livello superiore. Il livello inferiore è il ***service provider***, quello superiore è il ***service user***.

Un livello n che usufruisce dei servizi di livello (n-1) può, per mezzo di questi, a sua volta offrire al livello (n+1) i propri servizi.

I servizi sono disponibili ai ***SAP (Service Access Point)***. I SAP del livello n, o ***n-SAP***, sono i punti di accesso nei quali il livello (n+1) può accedere ai servizi del livello n. Ogni n-SAP ha un ***indirizzo*** che lo identifica univocamente.

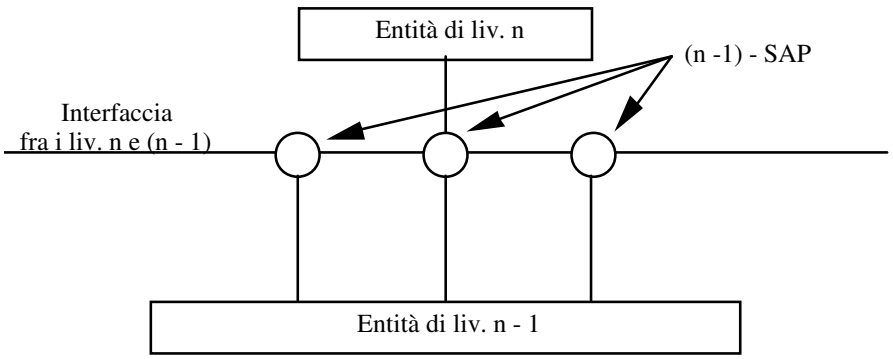

**Figura 1-15**: Livelli adiacenti e service access point

Analogia col telefono:

- SAP: presa a muro del telefono;
- SAP address: numero telefonico che identifica quella presa.

L'informazione passata dal livello n al livello (n-1), attraverso il (n-1)-SAP, si dice ***PDU (Protocol Data Unit) di livello n***, o ***n-PDU***.

Essa, entrando nel livello (n-1), diventa una ***SDU (Service Data Unit) di livello (n-1)***, o ***(n-1)-SDU***.

Entro il livello (n-1) viene aggiunta alla (n-1)-SDU una ***PCI (Protocol Control Information) di livello (n-1)***.

Il tutto diventa una (n-1)-PDU, che verrà passata al livello (n-2) attraverso un (n-2)-SAP.

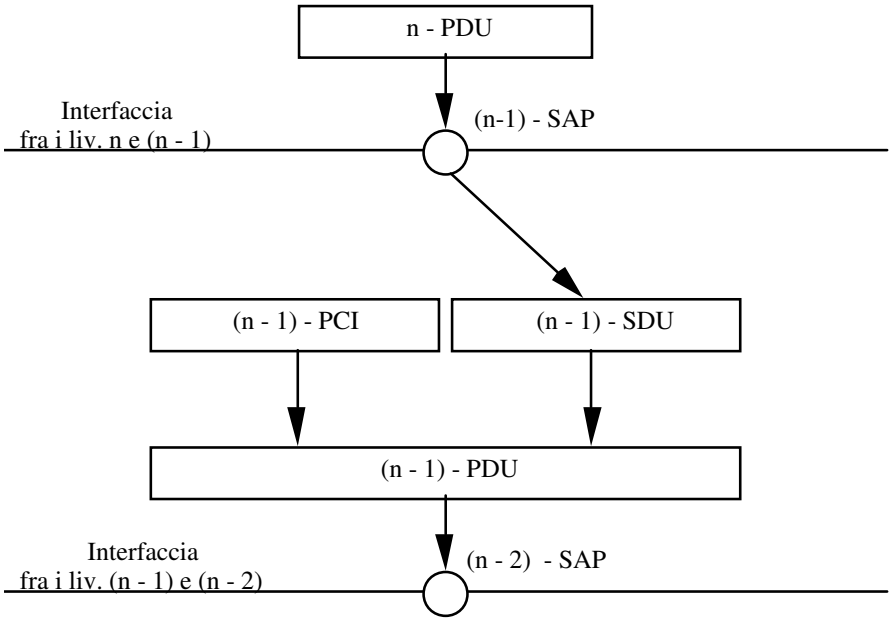


**Figura 1-16**: Passaggio dell'informazione fra livelli

Nomi spesso usati per i PDU:

- ***segmento*** (o***TPDU, Transport PDU***) a livello transport
- ***pacchetto*** (***packet***) a livello network
- ***trama*** (***frame***) a livello data link

Nome per il PCI:

- ***busta***

## 1.3.5) Servizi connection-oriented e connectionless

Ci sono due principali classi di servizi offerti da un livello a quello superiore:

- ***servizi connection-oriented***;
- ***servizi connectionless***.

**Servizi connection-oriented**

I servizi connection-oriented sono modellati secondo il sistema telefonico, dove per parlare con qualcuno si alza il telefono, si chiama, si parla e poi si riattacca. Ovvero:

1. si stabilisce una connessione;
2. si scambiano informazioni;
3. si rilascia la connessione.

Analogamente, un servizio connection-oriented si sviluppa in 3 fasi:

1. si stabilisce una connessione, cioé si crea con opportuni mezzi un "canale di comunicazione" fra la sorgente e la destinazione. La relativa attività tipicamente coinvolge un certo numero di elaboratori nel cammino fra sorgente e destinazione;
2. la connessione, una volta stabilita, agisce come un tubo digitale lungo il quale scorrono tutti i dati trasmessi, che arrivano nello steso ordine in cui sono partiti;
3. si rilascia la connessione (attività che coinvolge di nuovo tutti gli elaboratori sul cammino).

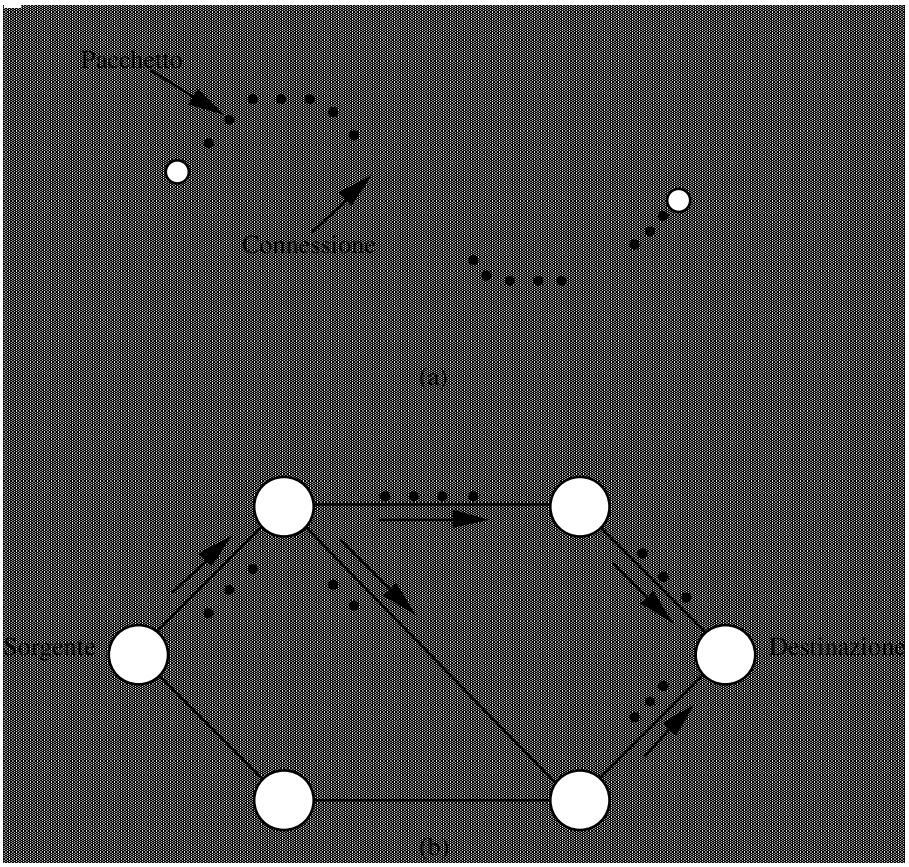


**Figura 1-17**: Servizi connection-oriented (a) e connectionless (b)

**Servizi connectionless**

I servizi connectionless sono modellati secondo il sistema postale: ogni lettera viaggia indipendentemente dalle altre; arriva quando arriva, e forse non arriva. Inoltre, due lettere con uguale mittente e destinatario possono viaggiare per strade diverse.

Analogamente, in un servizio connectionless, i pacchetti (PDU) viaggiano indipendentemente gli uni dagli altri, possono prendere strade diverse ed arrivare in ordine diverso da quello di partenza o non arrivare affatto.

La fase è una sola:

- invio del pacchetto (corrisponde all'immissione della lettera nella buca).

### 1.3.6) Affidabilità del servizio

Un servizio è generalmente caratterizzato dall'essere o no ***affidabile*** (***reliable***).

Un ***servizio affidabile*** non perde mai dati, cioé assicura che tutti i dati spediti verranno consegnati al destinatario. Ciò generalmente richiede che il ricevente invii un ***acknowledgement (conferma)*** alla sorgente per ogni pacchetto ricevuto. Si introduce ovviamente overhead, che in certe situazioni può non essere desiderabile.

Viceversa, un ***servizio non affidabile*** non offre la certezza che i dati spediti arrivino effettivamente a destinazione.

Si noti che se un certo livello non offre nessun servizio affidabile, qualora tale funzionalità sia desiderata dovrà essere fornita da almeno uno dei livelli superiori (vedremo che ciò accade spesso).

Esempi:

- ***reliable connection oriented***: trasferimento di file (non devono mancare pezzi e il file non deve essere "rimescolato");
- ***non reliable connection oriented***: nelle trasmissioni isocrone (quali voce e video) le relazioni temporali fra i bit del flusso devono essere mantenute. E' meglio qualche disturbo ogni tanto, piuttosto che interruzioni momentanee, ma avvertibili, del flusso di dati;
- ***non reliable connectionless*** (detto anche ***datagram service***, da telegram): distribuzione di posta elettronica pubblicitaria, non importa se qualche messaggio si perde.
- ***reliable connectionless*** (detto anche ***acknowledged datagram service***): si invia un breve messaggio e si vuole essere assolutamente sicuri che è arrivato.

### 1.3.7) Primitive di definizione del servizio

Un servizio di livello n è formalmente specificato da un insieme di ***primitive*** (cioé operazioni) che un'entità di livello (n+1) può adoperare per accedere al servizio. Esse possono indicare al servizio:

- l'azione da compiere (l'informazione viagga da livello n al livello (n-1));
- cosa riportare in merito ad una azione effettuata dalla peer entity di livello n (l'informazione viagga dal livello (n-1) al livello n).

Un esempio di primitive può essere il seguente:

| Primitiva | Significato |
|---|---|
| | |
| `request()` | si chiede al servizio di fare qualcosa |
| `indication()` | si viene avvertiti, dal servizio, di qualche evento |
| `response()` | si vuole rispondere ad un evento |
| `confirm()` | la risposta che si attendeva è arrivata |

Per stabilire una connessione fra le peer entity A a B si avrà che:

| Entity | Azione | Flusso informazione | Significato |
|---|---|---|---|
| | | | |
| A | invia una `connect.request()` | da livello n a livello (n-1) | A desidera connettersi |
| B | riceve una `connect.indication()` | da livello (n-1) a livello n | qualcuno vuole connettersi |
| B | invia una | da livello n | B accetta (oppure no) |

| | | | |
|---|---|---|---|
| | `connect.response()` | a livello (n-1) | |
| A | riceve una<br>`connect.confirm()` | da livello (n-1)<br>a livello n | B ha accettato (o no) |

Le primitive hanno vari parametri (mittente, destinatario, tipo del servizio richiesto, ecc.), che possono essere usati dalle peer entity per negoziare le caratteristiche della connessione. I dettagli della negoziazione fanno parte del protocollo.

Ad esempio, un servizio può essere richiesto in modalità confermata oppure non confermato.

Per il servizio confermato avremo:

- `request();`
- `indication();`
- `response();`
- `confirm().`

Mentre per il servizio non confermato:

- `request();`
- `indication().`

Connect è sempre confermato (ovviamente), ma altri servizi no.

Vediamo ora un esempio di servizio connection oriented con 8 primitive:

1. `connect.request();`
2. `connect.indication();`
3. `connect.response();`
4. `connect.confirm();`
5. `data.request()`: si cerca di inviare dati;
6. `data.indication()`: sono arrivati dei dati;
7. `disconnect.request()`: si vuole terminare la connessione;
8. `disconnect.indication()`: l'altra entity vuole terminare.

La sequenza di primitive che entrano in gioco successivamente nel corso della gestione di una connessione è la seguente:

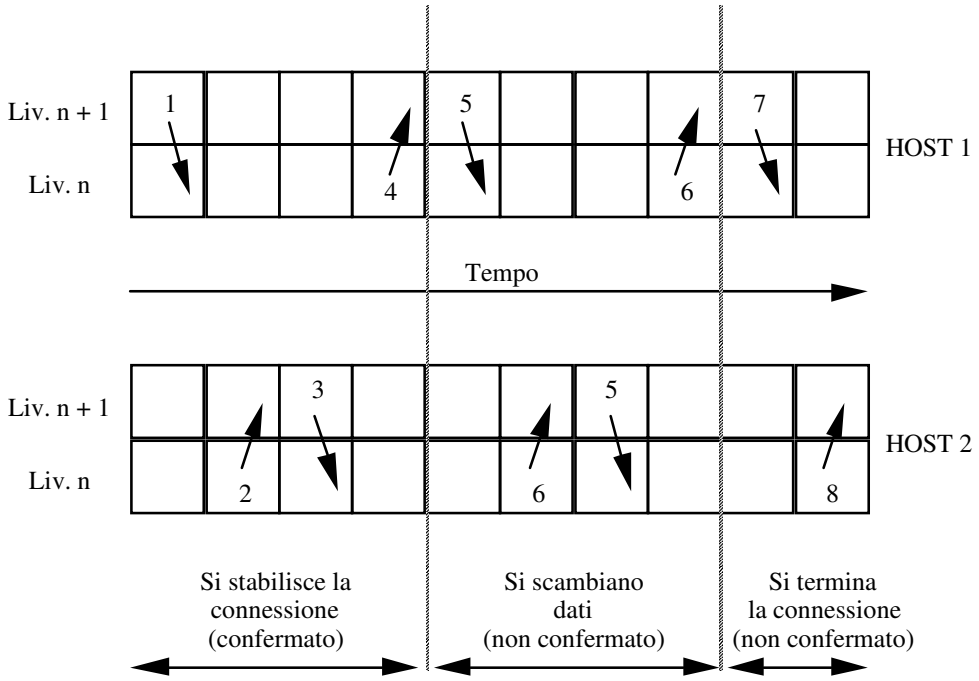


**Figura 1-18**: Esempio di attivazione, uso e rilascio di una connessione

## 1.3.8) Servizi vs. protocolli

Servizi e protocolli sono spesso confusi, ma sono concetti ben distinti.

| | |
|---|---|
| ***Servizio*** | insieme di operazioni primitive che un livello offre al livello superiore. Come tali operazioni siano implementate non riguarda il livello superiore. |
| ***Protocollo*** | insieme di regole che governano il formato ed il significato delle informazioni (messaggi, frame, pacchetti) che le peer entity si scambiano fra loro. Le entità usano i protocolli per implementare i propri servizi. |

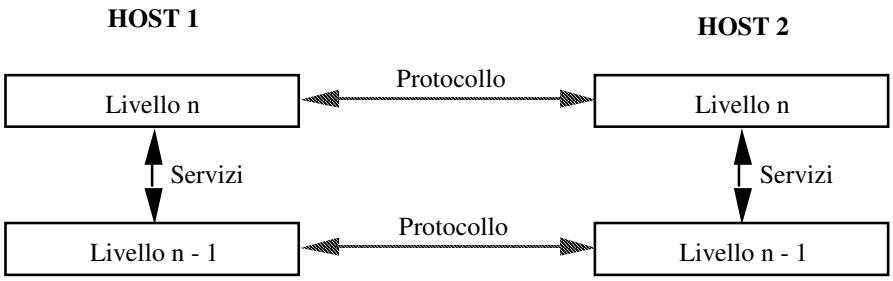


**Figura 1-19**: Relazione fra protocolli e servizi

Nota: una ipotetica primitiva `send_packet()`, alla quale l'utente fornisce il puntatore a un pacchetto già assemblato, non è conforme a questa filosofia.

## 1.3.9) Aspetti di progetto dei livelli

Decisioni di progetto vanno prese, nei vari livelli, in merito a diverse problematiche. Le principali sono:

1. Meccanismi di identificazione di mittente e destinatario (cioé ***indirizzamento***), in ogni livello.
2. Regole per il ***trasferimento dati*** (livelli bassi):
    - in una sola direzione (***simplex connection***);
    - in due direzioni ma non contemporaneamente (***half-duplex connection***).
    - in due direzioni contemporaneamente (***full-duplex connection***);
3. Meccanismi per il controllo degli ***errori di trasmissione***; è possibile:
    - rilevarli oppure no;
    - correggerli oppure no;
    - avvertire il mittente oppure no.
4. Meccanismi per il mantenimento (o la ricostruzione) dell'***ordine originario*** dei dati.
5. Meccanismi per regolare le ***velocità*** di sorgente e destinazione.
6. Decisioni sulla ***dimensione*** (minima o massima) dei messaggi da inviare, e su come eventualmente frammentarli.
7. Meccanismi di ***multiplexing*** di varie "conversazioni" su di un'unica connessione (se stabilire la connessione è costoso).

8. Meccanismi di ***routing*** dei messaggi se esistono più strade alternative, ed eventualmente di suddivisione di una "conversazione" su più connessioni contemporaneamente (per aumentare la velocità di trasferimento dei dati).

## *1.4) La realtà nel mondo delle reti*

Iniziamo ad esaminare due importanti realtà nel mondo delle reti:

1. ***OSI Reference Model***;
2. ***Internet Protocol Suite*** (detta anche ***architettura TCP/IP*** o, piuttosto impropriamente, ***TCP/IP reference model***).

Un modello di riferimento è cosa diversa da un'architettura di rete:

| | |
|---|---|
| ***Modello di riferimento*** | definisce il numero, le relazioni e le caratteristiche funzionali dei livelli, ma non definisce i protocolli effettivi |
| ***Architettura di rete*** | definisce, livello per livello, i protocolli effettivi |

### 1.4.1) Modello OSI

L'OSI (Open Systems Interconnection) Reference Model è il frutto del lavoro della ISO (International Standard Organization), ed ha lo scopo di:

- fornire uno standard per la connessione di sistemi aperti, cioé in grado di colloquiare gli uni con gli altri;
- fornire una base comune per lo sviluppo di standard per l'interconnessione di sistemi;
- fornire un modello rispetto a cui confrontare le varie architetture di rete.

Esso non include di per se la definizione di protocolli specifici (che sono stati definiti successivamente, in documenti separati).

Principi di progetto seguiti durante lo sviluppo del modello OSI:

- ogni livello deve avere un diverso livello di astrazione;
- ogni livello deve avere una funzione ben definita;
- la scelta dei livelli deve:
    - minimizzare il passaggio delle informazioni fra livelli;
    - evitare:
        - troppe funzioni in un livello;
        - troppi livelli.

Il modello OSI consiste di 7 livelli (i maligni dicono che ciò fu dettato dal desiderio di rendere il modello compatibile con l'architettura SNA dell'IBM).

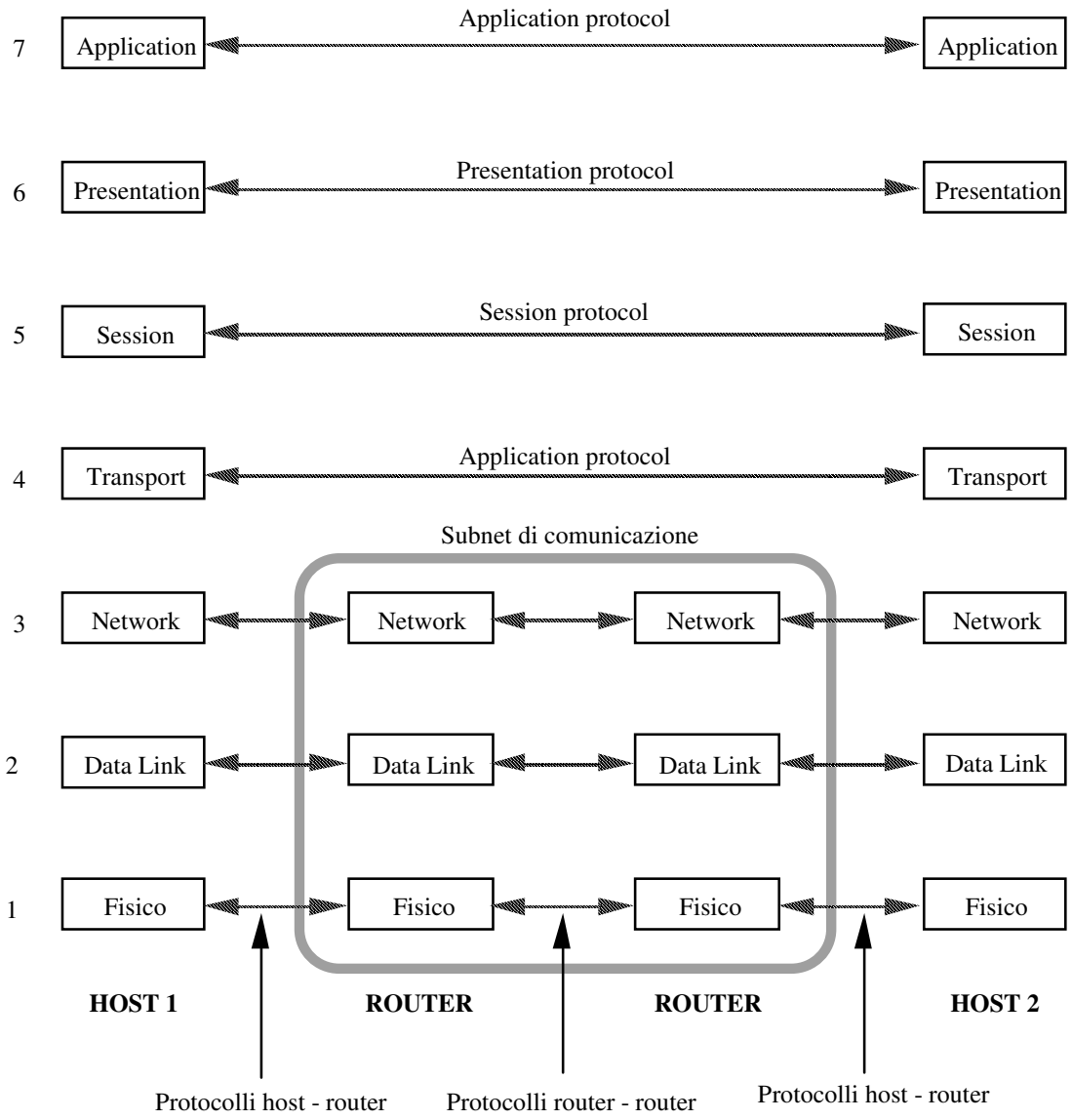


**Figura 1-20**: Il modello OSI

Spesso, per visualizzare le competenze (in termini di livelli gestiti) dei vari elaboratori sul cammino, si usano diagrammi simili al seguente:

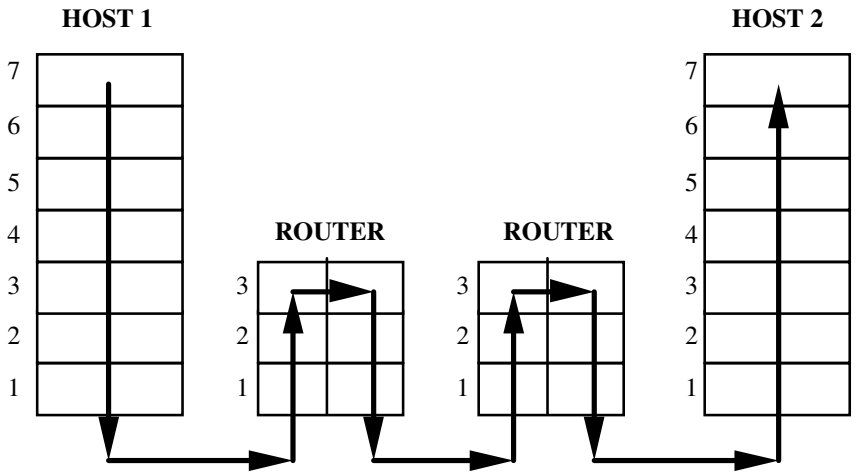


**Figura 1-21**: Rappresentazione schematica dei livelli gestiti lungo un cammino

Si noti che il modello OSI non è un'architettura di rete, perché dice solo cosa devono fare i livelli, ma non definisce né i servizi né i protocolli. Per questo ci sono separati documenti di definizione degli standard.

### Livello fisico

Ha a che fare con la trasmissione di bit "grezzi" su un canale di comunicazione.

Gli aspetti di progetto sono:

- volti a garantire che se parte un 1, arrivi effettivamente un 1 e non uno zero;
- largamente riguardanti le caratteristiche meccaniche, elettriche e procedurali delle ***interfacce di rete*** (componenti che connettono l'elaboratore al mezzo fisico) e le caratteristiche del mezzo fisico.

Si caratterizzano, tra gli altri:

- tensioni scelte per rappresentare 0 ed 1;
- durata (in microsecondi) di un bit;
- trasmissione simultanea in due direzioni oppure no;
- forma dei connettori.

**Livello Data Link**

Lo scopo di questo livello è far si che un mezzo fisico trasmissivo appaia, al livello superiore, come una linea di trasmissione esente da errori di trasmissione non rilevati.

Normalmente funziona così:

- spezzetta i dati provenienti dal livello superiore in frame (da qualche centinaia a qualche migliaia di byte);
- invia i frame in sequenza;
- aspetta un ***acknowledgement frame*** (***ack***) per ogni frame inviato.

Incombenze:

- aggiunta di delimitatori (***framing***) all'inizio ed alla fine del frame (che succede se il delimitatore è presente dentro il frame ?);
- ***gestione di errori*** di trasmissione causati da:
    - errori in ricezione;
    - perdita di frame;
    - duplicazione di frame (da perdita di ack);
- ***regolazione del traffico*** (per impedire che il ricevente sia "sommerso" di dati);
- meccanismi per l'invio degli ack:
    - ***frame separati*** (che però competono col regolare traffico nella stessa direzione);
    - ***piggybacking*** (da pickaback, cioé trasportare sulle spalle).

Le reti broadcast hanno un'ulteriore problema: il controllo dell'accesso al canale trasmissivo, che è condiviso. Per questo hanno uno speciale sottolivello del livello data link, il ***sottolivello MAC (Medium Access Control)***.

**Livello Network**

Lo scopo del livello è controllare il funzionamento della subnet di comunicazione.

Inizialmente tale livello offriva solamente servizi connection oriented; successivamente fu aggiunta la modalità connectionless.

Incombenze:

- ***routing***, cioé scelta del cammino da utilizzare. Può essere:
    - statico (fissato ogni tanto e raramente variabile);
    - dinamico (continuamente aggiornato, anche da un pacchetto all'altro);
- ***gestione della congestione***: a volte troppi pacchetti arrivano ad un router (es.: da molte linee in ingresso ad un unica linea di uscita);
- ***accounting***: gli operatori della rete possono far pagare l'uso agli utenti sulla base del traffico generato;
- ***conversione di dati*** nel passaggio fra una rete ed un'altra (diversa):
    - indirizzi da rimappare;
    - pacchetti da frammentare;
    - protocolli diversi da gestire.

**Livello Transport**

Lo scopo di questo livello è accettare dati dal livello superiore, spezzettarli in pacchetti, passarli al livello network ed assicurarsi che arrivino alla peer entity che si trova all'altra estremità della connessione. In più, fare ciò efficientemente, isolando i livelli superiori dai cambiamenti della tecnologia di rete sottostante.

Il livello transport è il primo livello realmente ***end-to-end***, cioé da host sorgente a host destinatario: le peer entity di questo livello portano avanti una conversazione senza intermediari.

Si noterà che certe problematiche sono, in ambito end-to-end, le stesse che il livello data link ha nell'ambito di una singola linea di comunicazione; le soluzioni però sono alquanto diverse per la presenza della subnet di comunicazione.

Incombenze:

- ***creazione di connessioni di livello network*** (attraverso i servizi del livello network) per ogni connessione di livello transport richiesta:
    - normalmente, una connessione network per ciascuna connessione transport;
    - per ottenere un alto throughput: molte connessioni network per una singola connessione transport;

    - se è alto il costo di una connessione network: una singola connessione network viene usata per molte connessioni transport, con meccanismi di multiplexing;
- *offerta di vari servizi* al livello superiore:
    - canale punto a punto affidabile, che consegna i dati in ordine e senza errori (il servizio più diffuso, connection oriented);
    - invio di messaggi isolati, con o senza garanzia di consegna (connectionless);
    - broadcasting di messaggi a molti destinatari (connectionless).

**Livello Session**

Ha a che fare con servizi più raffinati che non quelli del transport layer, come ad es.:

- *token management*: autorizza le due parti, a turno, alla trasmissione.

Come vedremo nel seguito, questo livello non ha avuto un grande successo.

**Livello Presentation**

E' interessato alla sintassi ed alla semantica delle informazioni da trasferire. Ad esempio, si occupa di convertire tipi di dati standard (caratteri, interi) da rappresentazioni specifiche della piattaforma HW di partenza in una rappresentazione "on the wire" e poi in quella specifica dell' HW di arrivo.

Anche questo livello non ha avuto molto successo.

**Livello Application**

Prevede che qui risieda tutta la varietà di protocolli che sono necessari per offrire i vari servizi agli utenti, quali ad esempio:

- *terminale virtuale*;
- *transferimento file*;
- *posta elettronica*.

Attraverso l'uso di questi protocolli si possono scrivere applicazioni che offrono i suddetti servizi agli utenti finali.

## 1.4.2) Internet Protocol Suite

La "madre di tutte le reti" fu ***Arpanet***, originata da un progetto di ricerca finanziato dal DoD (Department of Defense) americano. Lo scopo era creare una rete estremamente affidabile anche in caso di catastrofi (o eventi bellici) che ne eliminassero una parte. Arpanet, attraverso varie evoluzioni, ha dato origine alla attuale Internet.

Nel corso dello sviluppo, per integrare via via tipi diversi di reti, si vide la necessità di una nuova architettura, mirata fin dall'inizio a consentire l'interconnessione di molteplici reti (internetwork).

L'architettura divenne, più tardi, nota coi nomi di ***Internet Protocol Suite***, ***architettura TCP/IP*** e ***TCP/IP reference model***, dal nome dei suoi due protocolli principali. Essa non è un modello nel senso stretto del termine, in quanto include i protocolli effettivi, che sono specificati per mezzo di documenti detti ***RFC*** (***Request For Comments***).

I livelli TCP/IP hanno questa relazione con quelli OSI:

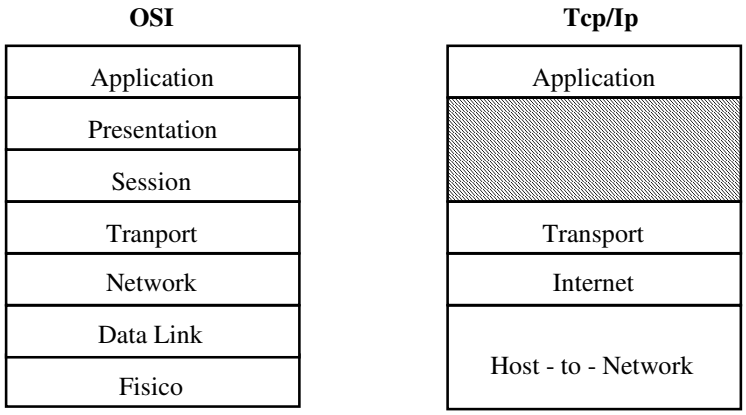


**Figura 1-22**: Relazione fra i livelli OSI e TCP/IP

I requisiti di progetto stabiliti fin dall'inizio (estrema affidabilità e tolleranza ai guasti, possibilità di interconnessione di più reti) portarono alla scelta di una rete:

- packet-switched;
- basata su un livello connectionless di internetwork.

**Livello host-to-network**

Il livello più basso non è specificato nell'architettura, che prevede di utilizzare quelli disponibili per le varie piattaforme HW e conformi agli standard.

Tutto ciò che si assume è la capacità dell'host di inviare pacchetti IP sulla rete.

**Livello Internet**

E' il livello che tiene insieme l'intera architettura. Il suo ruolo è permettere ad un host di iniettare pacchetti in una qualunque rete e fare il possibile per farli viaggiare, indipendentemente gli uni dagli altri e magari per strade diverse, fino alla destinazione, che può essere situata anche in un'altra rete. Dunque è ***connectionless***. E' un servizio ***best-effort datagram***. E' definito un formato ufficiale dei pacchetti ed un protocollo, ***IP (Internet Protocol)***.

Incombenze:

- routing;
- controllo congestione.

**Livello Transport**

E' progettato per consentire la conversazione delle peer entity sugli host sorgente e destinazione (end-to-end). Sono definiti due protocolli in questo livello:

- ***TCP (Transmission Control Protocol)***: è un protocollo connesso ed affidabile (ossia tutti i pacchetti arrivano, e nell'ordine giusto). Frammenta il flusso in arrivo dal livello superiore in messaggi separati che vengono passati al livello Internet. In arrivo, i pacchetti vengono riassemblati in un flusso di output per il livello superiore.
- ***UDP (User Datagram Protocol)***: è un protocollo non connesso e non affidabile, i pacchetti possono arrivare in ordine diverso o non arrivare affatto.

**Livello Application**

Nell'architettura TCP/IP non ci sono i livelli session e presentation (non furono ritenuti necessari; l'esperienza col modello OSI ha mostrato che questa visione è condivisibile).

Sopra il livello transport c'è direttamente il livello application, che contiene tutti i protocolli di alto livello vengono usati dalle applicazioni reali.

I primi protocolli furono:

- ***Telnet***: terminale virtuale;
- ***FTP (File Transfer Protocol)***: file transfer;
- ***SMTP (Simple Mail Transfer Protocol) e POP (Post Office Protocol)***: posta elettronica.

Successivamente se ne sono aggiunti altri, fra cui:

***DNS (Domain Name Service)***: mapping fra nomi di host e indirizzi IP numerici;

***NNTP (Network News Transfer Protocol)***: trasferimento di articoli per i newsgroup;

***HTTP (HyperText Transfer Protocol)***: alla base del Word Wide Web.

I vari protocolli nell'architettura TCP/IP si collocano come segue:

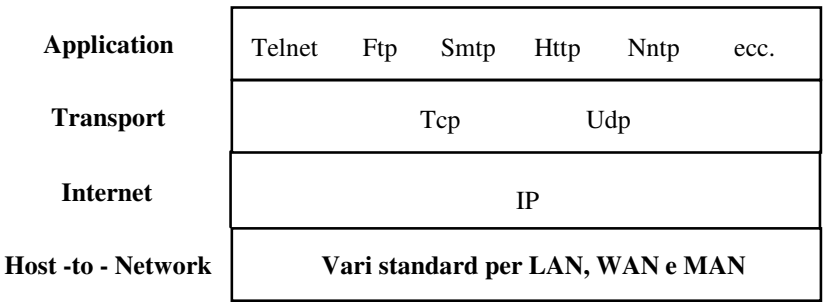


**Figura 1-23**: Relazione fra i livelli e i protocolli dell'architettura TCP/IP

### 1.4.3) Confronto fra modello di riferimento OSI e architettura TCP/IP

**Somiglianze**:

- basati entrambi sul concetto di pila di protocolli indipendenti;
- funzionalità simili in entrambi per i vari livelli.

**Differenze di fondo:**

- OSI nasce come modello di riferimento (utilissimo per le discussioni generali), i protocolli vengono solo successivamente;
- TCP/IP nasce coi protocolli, il modello di riferimento viene a posteriori.

**Conseguenze:**

essendo il modello OSI nato prima dei relativi protocolli, successe che:

- il modello era, ed è tuttora, molto generale (punto a favore);
- vi era insufficiente esperienza nella progettazione dei livelli (punto a sfavore). Ad esempio:
    - il livello data-link (pensato all'origine per linee punto-punto) ha dovuto essere sdoppiato per gestire reti broadcast;
    - mancò del tutto l'idea di internetworking: si pensava ad una rete separata, gestita dallo stato, per ogni nazione.

I protocolli dell'architettura TCP/IP sono invece il punto di partenza del progetto, per cui:

- l'architettura è molto efficiente (punto a favore);
- il reference model non è generale, in quanto descrive solo questa particolare architettura (punto a sfavore);
- è difficile rimpiazzare i protocolli se necessario (punto a sfavore).

**Confronto fra pile di protocolli OSI e TCP/IP**

I protocolli OSI non sono riusciti ad affermarsi sul mercato per una serie di ragioni:

- infelice scelta di tempo: la definizione dei protocolli è arrivata troppo tardi, quando cioé quelli TCP/IP si erano già considerevolmente diffusi. Le aziende non se la sono sentite di investire risorse nello sviluppo di una ulteriore architettura di rete;

- infelici scelte tecnologiche: i sette livelli (e i relativi protocolli) sono stati dettati in realtà dalla architettura SNA dell' IBM, più che da considerazioni di progetto. Per cui il progetto soffre di vari difetti:
    - grande complessità e conseguente difficoltà di implementazione;
    - inutili i livelli session e presentation;
    - non ottimali attribuzioni di funzioni ai vari livelli:
        - alcune funzioni appaiono in molti livelli (es. controllo errore e flusso in tutti i livelli);
        - altre funzioni mancano del tutto (ad es. sicurezza e gestione rete);
- infelice implementazione: le prime realizzazioni erano lente ed inefficienti, mentre contemporaneamente TCP/IP era molto ben implementato (e per di più gratis!). In effetti i protocolli dell'architettura TCP/IP invece sono stati implementati efficientemente fin dall'inizio, per cui si sono affermati sempre più, e quindi hanno goduto di un crescente supporto che li ha resi ancora migliori.

Ad ogni modo, neanche l'architettura TCP/IP è priva di problemi:

- l'architettura TCP/IP non ha utilità come modello (non serve ad altro che a descrivere se stessa);
- non c'è una chiara distinzione fra protocolli, servizi e interfacce, il che rende più difficile l'evoluzione dell'architettura;
- alcune scelte di progetto cominciano a pesare (ad es., indirizzi IP a soli 32 bit).

In conclusione:

- OSI è ottimo come modello, mentre i suoi protocolli hanno avuto poco successo;
- TCP/IP è ottima (per ora) come architettura di rete, ma inutile come modello.

Nel resto del corso ci concentreremo su un modello di riferimento OSI modificato, nonché sui protocolli principali dell'architettura TCP/IP.

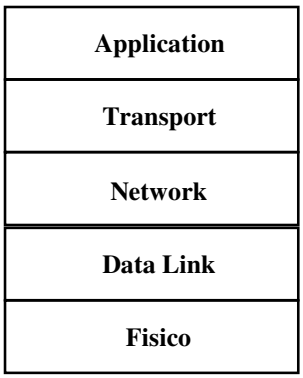


**Figura 1-24**: Il modello OSI modificato

## 1.4.4) Esempi di architetture di rete

### Netware (Novell)

E' l'architettura di rete più diffusa nel mondo PC. E' precedente a OSI, e assomiglia molto a TCP/IP.

| | |
|---|---|
| **Application** | **File server, ecc.** |
| **Transport** | **NCP, SPX, Tcp** |
| **Network** | **IPX** |
| **Data Link** | **Ethernet, Token Ring, ecc.** |
| **Fisico** | **Ethernet, Token Ring, ecc.** |

**Figura 1-25**: L'architettura Novell Netware

- IPX: best-effort datagram (connectionless);
- NCP: reliable connection oriented;
- SPX: unreliable connectionless

Nota: la pratica di avere un livello network best-effort e connectionless, e sopra esso un livello transport reliable e connected, è molto diffusa. Sono così anche:

- Appletalk;
- TCP/IP.

**Arpanet**

Nacque a metà degli anni '60 (ai tempi della guerra fredda); il DoD volle una rete robustissima anche in caso di catastrofi, in grado di non interrompere le connessioni in atto anche se alcune sue componenti fossero state distrutte.

Si scelse quindi una rete packed-switched, formata dagli host e da una subnet di comunicazione costituita da vari IMP (Interface Message Processor) collegati da linee di trasmissione.

Il software venne diviso in due parti:

- SW per gli host
- SW per la subnet (ossia per gli IMP)

e furono definiti alcuni protocolli:

- Host-IMP protocol;
- IMP-IMP protocol;
- Source IMP-destination IMP protocol.

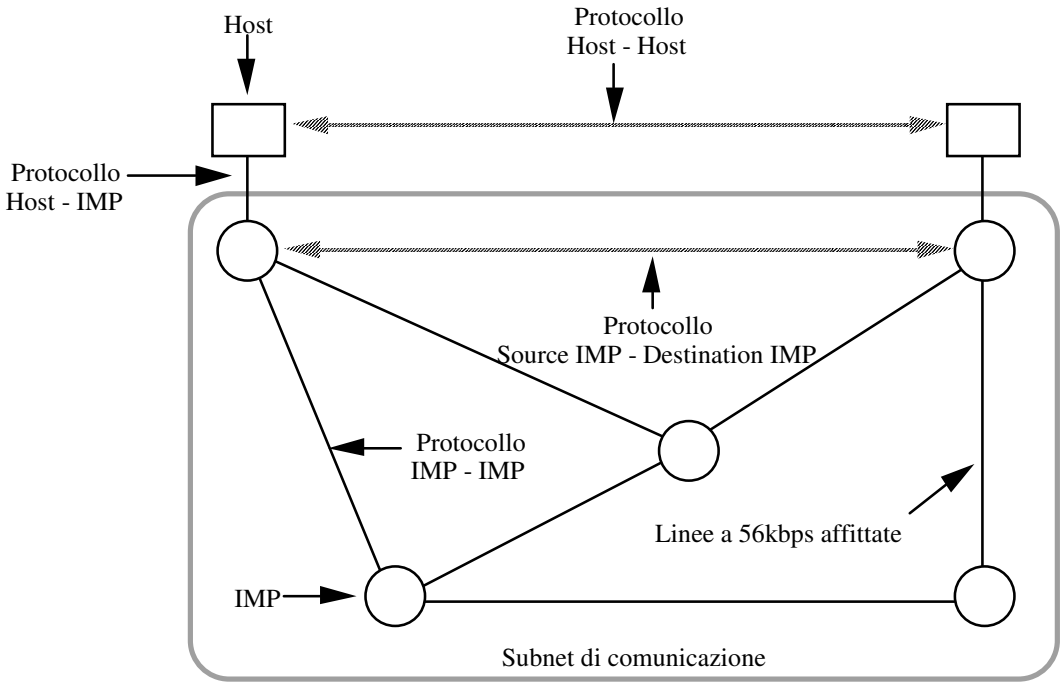


**Figura 1-26**: L'architettura di Arpanet

Successivamente Arpanet si sviluppò incorporando altre reti, il che mostrò l'inadeguatezza dei protocolli originari rispetto alle problematiche di internetworking.

Nacque di conseguenza, verso la metà degli anni '70, l'architettura TCP/IP, che il giorno 1/1/1983 divenne lo standard di Arpanet. TCP/IP fu poi mantenuta anche per l'evoluzione di Arpanet, National Science Foundation Net (NSFNET, metà degli anni '80), basata su linee da 56Kbps all'inizio e poi più veloci (1.5 Mbps nel 1990, 34 Mbps sulle linee principali oggi).

Il continuo aggiungersi di reti, ad Arpanet prima e ad NSFNET poi, ha creato quella che oggi viene comunemente chiamata Internet, costituita da milioni di host e utilizzata da decine di milioni di utenti. Internet si raddoppia all'incirca ogni anno, ed ha suscitato grande interesse sia per i privati cittadini che per le aziende.

**Decnet Phase V (Digital)**

Come già detto, Decnet Phase V è un'architettura di rete totalmente conforme al modello OSI. Decnet ha costruito una piattaforma comprendente i livelli 1, 2 e 3 del modello OSI, sulla quale possono appoggiarsi due diverse pile di protocolli:

- pila OSI;
- pila Decnet Phase IV (proprietaria).

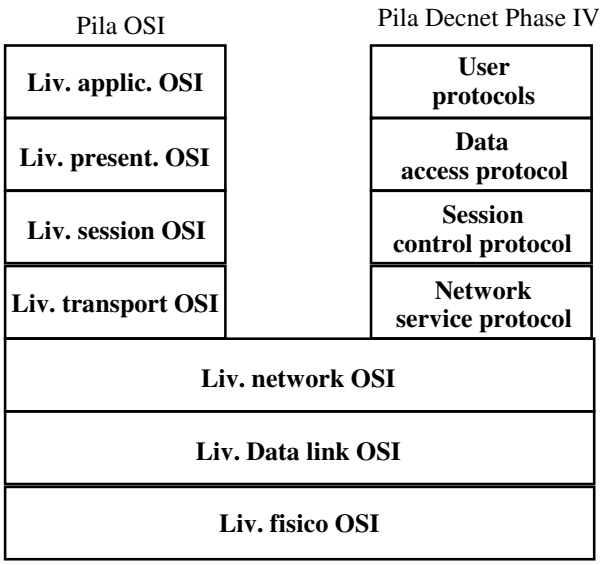


**Figura 1-27**: L'architettura Decnet Phase V

**SNA (IBM)**

SNA (System Network Architecture) è un'architettura di rete ancora molto diffusa, sopratutto nelle grandi aziende dotate di sistemi informativi IBM. Nacque a metà degli anni '70, periodo in cui i sistemi informativi erano basati sull'uso di mainframe e terminali.

Fu pensata per connettere fra loro più mainframe, ai quali dovevano poter essere connessi moltissimi terminali, anche geograficamente lontani.

E' un'architettura estremamente complessa e poco adatta all'attuale impostazione dei sistemi di calcolo (reti locali e applicazioni tipo client-server) per cui IBM sta migrando verso una strategia di networking più ampia, nella quale viene fornito il supporto ad SNA, APPN (successore di SNA: architettura proprietaria ma di dominio pubblico), OSI, TCP/IP ed altri (fra cui Novell IPX).

L'architettura SNA è la seguente:

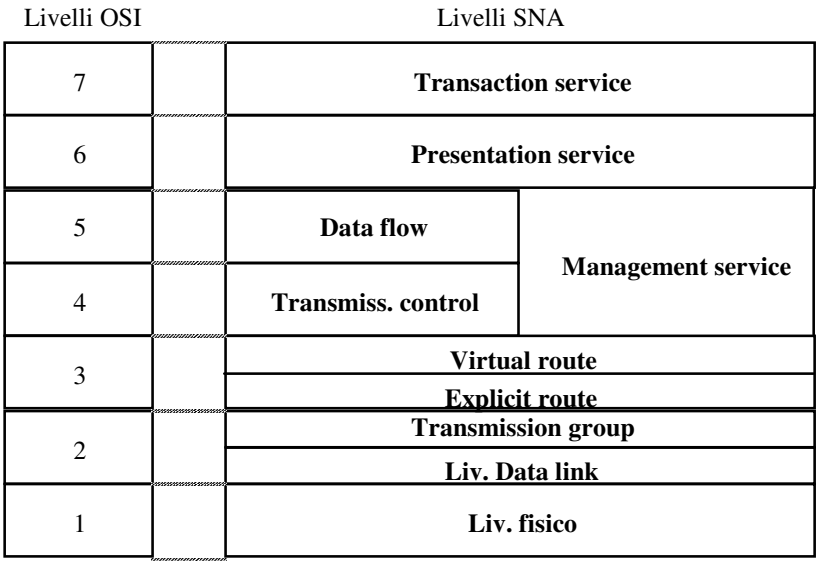


**Figura 1-28**: L'architettura IBM SNA

## 1.4.5) Autorità nel mondo degli standard

Queste sono le principali autorità nel mondo degli standard:

- ***PTT (Post, Telephone and Telegraph)***: amministrazione statale che gestisce i servizi trasmissivi (in Italia è il Ministero delle Poste);
- ***CCITT (Comité Consultatif International de Telegraphie et Telephonie)***: organismo internazionale che emette le specifiche tecniche che devono essere adottate dalle PTT. E' entrato da poco a far parte dell'***ITU (International Telecomunication Union)***;
- ***ISO (International Standard Organization)***: il principale ente di standardizzazione internazionale, che si occupa fra l'altro anche di reti;
- ***ANSI (American National Standards Institution)***: rappresentante USA nell' ISO;
- ***UNINFO***: rappresentante italiano, per le reti, nell'ISO;
- ***IEEE (Institute of Electrical and Electronic Engineers)***: organizzazione professionale mondiale degli ingegneri elettrici ed elettronici; ha gruppi di standardizzazione sulle reti;
- ***IRTF (Internet Research Task Force)***: comitato rivolto agli aspetti di ricerca a lungo termine in merito alla rete Internet;

- ***IETF (Internet Engineering Task Force)***: comitato rivolto agli aspetti di ingegnerizzazione a breve termine della rete Internet;
- ***IAF (Internet Architecture Board)***: comitato che prende le decisioni finali su nuovi standard da adottare per Internet, di solito proposti da IETF o IRTF.