

Reti di computer - Capitolo 3

Il modello architetturale OSI

| | |
|---|----|
| Introduzione | 1 |
| <i>Principali autorità nel mondo degli standard</i> | 6 |
| I compiti di base dei 7 strati funzionali..... | 6 |
| Il modello di riferimento ISO/OSI | 7 |
| Descrizione dei singoli livelli del modello ISO/OSI | 9 |
| Primitive OSI | 14 |
| Note sul modello OSI | 17 |
| OSI: stato attuale | 18 |
| Architettura di rete | 19 |

INTRODUZIONE

Il collegamento e la cooperazione tra sistemi informatici che utilizzano sistemi operativi incompatibili tra loro è una delle principali esigenze del mercato attuale. I sistemi capaci di interagire tra loro, pur basandosi su sistemi operativi incompatibili, sono detti **aperti** quando permettono le comunicazioni in accordo con gli standard specificati nel modello generale **Open System Interconnection (OSI)**. Questi standard sono stati definiti da una speciale commissione dell'**International Standard Organization (ISO)**, ossia l'agenzia dell'ONU responsabile degli standard internazionali, inclusi quelli delle comunicazioni.

Questi standard sono nati come risposta alla diffusa esigenza di interconnettere tra loro sistemi incompatibili. La difficoltà di fondo consiste nel far comunicare tra loro due o più *processi*¹ che usano, internamente, regole e tecniche diverse. Ci si è preoccupati quindi di definire le strutture dei dati trasmessi, le regole e i comandi per la gestione dello scambio dati tra applicazione o tra utenti, i meccanismi di controllo che assicurano uno scambio senza errori.

Il **comitato ISO** ha stabilito le regole e le opzioni per tali interazioni, definendo un **modello di riferimento**. Un *modello di riferimento* è cosa diversa da un'*architettura di rete*:

- un *modello di riferimento* definisce il numero, le relazioni e le caratteristiche funzionali dei livelli, ma non definisce i protocolli effettivi;
- una *architettura di rete* definisce, livello per livello, i protocolli effettivi

Un modello di riferimento, quindi, non include di per sé la definizione di protocolli specifici, che invece vengono definiti successivamente, in documenti separati, come appunto accaduto dopo l'introduzione del **modello ISO/OSI**.

¹ Per *processi* possiamo intendere sia *sistemi informativi* veri e propri sia le *applicazioni* o gli utenti che interagiscono tra loro tramite i *terminali* di sistema.

Tale modello suddivide le necessarie funzioni logiche in sette diversi *strati funzionali*, detti **layer** (*livelli*). La figura seguente, che in seguito sarà ampiamente commentata, illustra tali livelli:

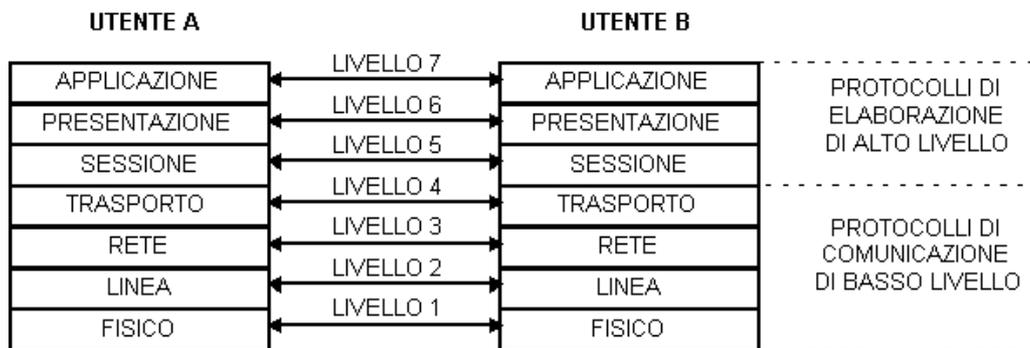


Figura 1 - Schema logico di due utenti (A e B) connessi tramite il modello OSI

L'insieme dei 7 layer garantisce tutte le funzioni necessarie alla rete comunicativa tra sistemi, nonché una gamma molto ampia di funzioni opzionali (come ad esempio la compressione e la cifratura dei dati): in tal modo, si è in pratica suddiviso un compito complesso in un insieme di compiti più semplici.

I principi di progetto che furono seguiti durante lo sviluppo del modello OSI furono sostanzialmente i seguenti:

- 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 ha tre capisaldi:

- **simmetria**: la simmetria assicura che le *funzioni logiche* di due qualsiasi sistemi interagenti siano le stesse, in accordo con gli standard². La simmetria consente di bilanciare i carichi elaborativi sui vari sistemi ed assicura che ogni richiesta, da qualunque parte provenga, venga interpretata correttamente dalla controparte. Si tratta dunque della base per i processi elaborativi di tipo cooperativo e di tipo *client-server*;
- **struttura gerarchica**: i vari sottosistemi (livelli) sono organizzati in una rigida gerarchia operativa; ogni livello riceve i comandi ed i dati dal livello superiore, esegue per esso alcune sue specifiche funzioni e, a sua volta, chiede servizi al livello subordinato. Il livello gerarchicamente più alto è quello dell'applicazione (*application layer*), ossia dell'utente: tutte le funzioni ed i servizi del modello sono sempre attivati su esigenza di chi opera al livello più

² Si tratta perciò di una soluzione opposta alla tecnica usata diversi anni fa, in cui la logica applicativa e comunicativa era tutta concentrata in un solo computer, mentre le stazioni connesse, indipendentemente dalle loro risorse, potevano solo inviare domande e ricevere le relative risposte.

alto (*application layer*); gli altri 6 livelli, in ordine decrescente di gerarchia, sono così denominati: *presentation, session, transport, network, data link control e interfaccia fisica*;

- **modularità**: la modularità garantisce che ogni livello abbia ben definite non solo le proprie *funzioni interne*, ma anche le *interfacce* con cui riceve o trasmette i comandi ed i dati verso i propri livelli adiacenti. La definizione formale di queste relazioni assicura che ogni livello funzionale abbia una propria caratteristica precisa, che lo distingue nettamente dagli altri. Questo implica che, per aggiungere opzioni o per permettergli prestazioni migliori grazie a nuove tecnologie, non sia necessario modificare anche gli altri livelli, il che significa, in altre parole, garantire la possibilità di sviluppo, al crescere delle possibilità tecnologiche e delle esigenze degli utenti.

Soffermiamoci sulla struttura gerarchica. 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.

C'è però un principio generale sempre rispettato: lo scopo di un livello è offrire certi **servizi** ai livelli più alti, nascondendo i dettagli su come tali servizi siano implementati..

Facciamo riferimento alla figura seguente:

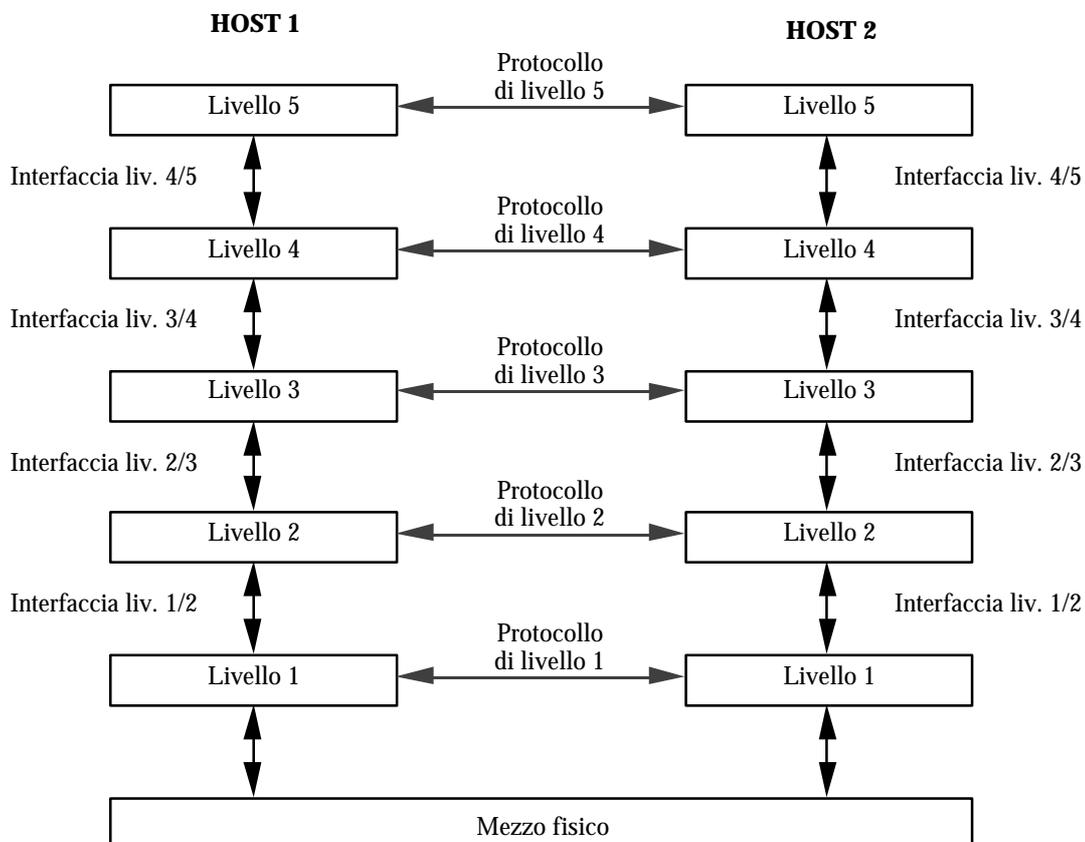


Figura 2 - Dialogo fra peer entity (sono considerati, per semplicità, solo 5 livelli, mentre il modello OSI, come visto, ne prevede 7)

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

E' bene però precisare subito (e lo faremo anche in seguito) che 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, fino al livello più basso, al sotto del quale c'è il mezzo fisico, attraverso cui i dati vengono effettivamente 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.

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

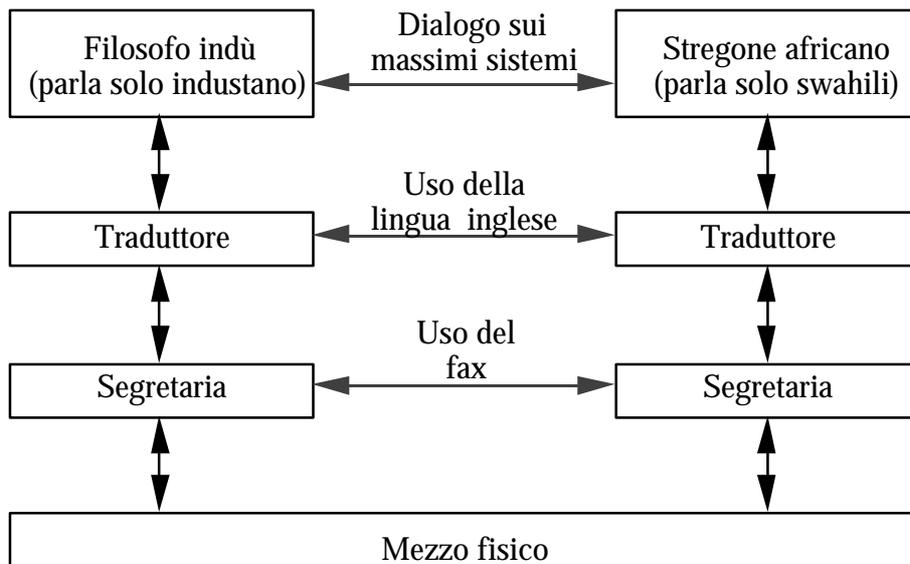


Figura 3 - Dialogo fra grandi menti

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:

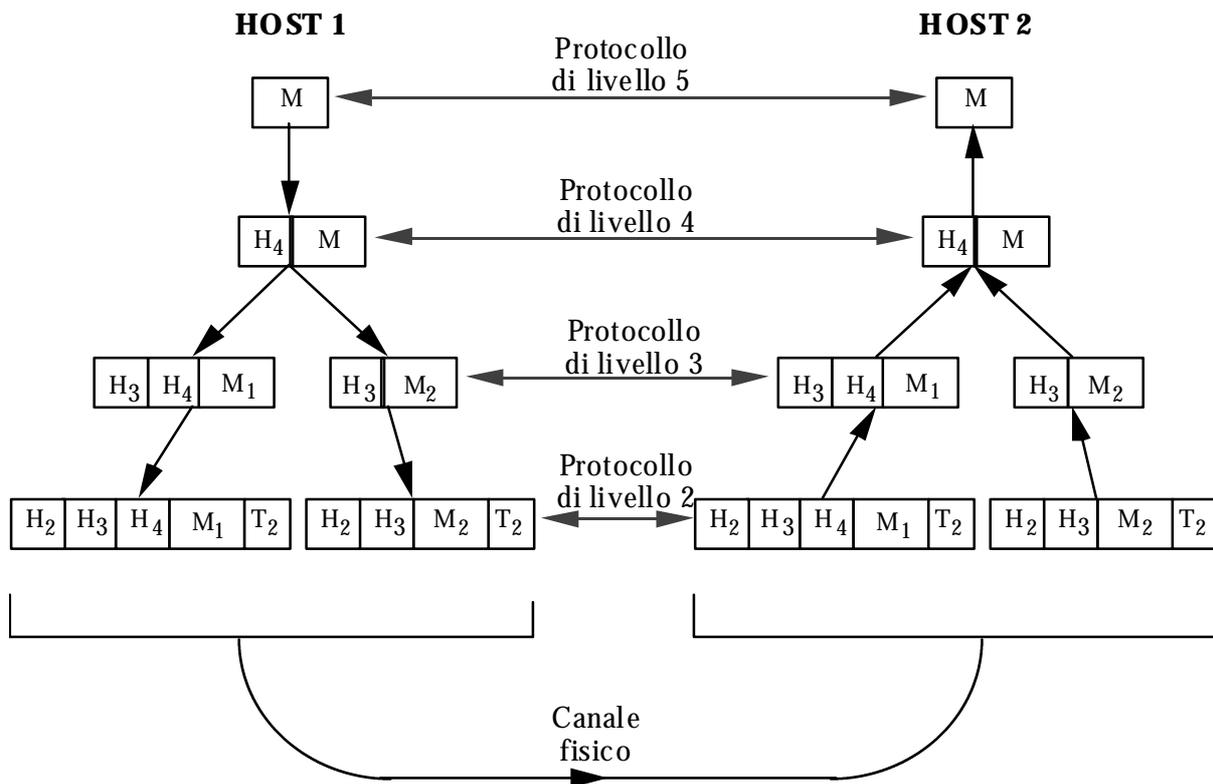


Figura 4 - Flusso dell'informazione fra peer entity

Vediamo cosa accade:

- il programma applicativo (livello 5) dell'*host 1* deve mandare un *messaggio* M alla sua peer entity dell'*host 2*;
- il livello 5 consegna M al livello 4 per la trasmissione;
- il livello 4 aggiunge un suo **header** (*testata*)³; questo header contiene informazioni di controllo, tra le quali il numero di sequenza del messaggio, la dimensione del messaggio e altro.
- il livello 4 consegna il risultato al livello 3;
- il livello 3 può trovarsi nella necessità di frammentare i dati da trasmettere in unità più piccole, (i cosiddetti **pacchetti**) a ciascuna delle quali aggiunge il suo header;
- il livello 3 passa i pacchetti al livello 2;
- il livello 2 aggiunge ad ogni pacchetto il proprio header (e magari un **trailer**) e lo spedisce sul canale fisico;
- 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:

³ Talvolta si dice che il messaggio è inserito nella *busta* di livello 4

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

Principali 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 (Consultative Committee for International Telegraph and Telephone)**: organismo internazionale che emette le specifiche tecniche che devono essere adottate dalle PTT. E' entrato da poco a far parte dell'*ITU (International Telecommunication 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.

I COMPITI DI BASE DEI 7 STRATI FUNZIONALI

Al fine di descrivere i 7 strati funzionali (*layer*) di cui si compone il *modello ISO/OSI*, esaminiamo velocemente i vari comportamenti di una stazione connessa ad una rete (di qualsiasi tipo):

- 1) **livello applicativo**: il processo elaborativo elabora i dati in accordo sia alle richieste dell'utente sia alle norme applicative prestabilite. In determinati momenti, può capitare che certi dati o certe richieste vadano trasmessi ad una controparte remota: il processo elaborativo si occupa allora di preparare sia i dati da trasmettere sia i motivi della trasmissione.
- 2) **livello presentation**: i dati vengono strutturati in modo che il processo remoto possa comprenderli ed elaborarli;
- 3) **livello session**: il sistema esamina se la connessione logica con la controparte è stata già attivata o meno; in caso negativo, prima che venga attivata, è necessario disporre di regole (preventivamente fissate) per il dialogo da instaurare: ad esempio, si deve sapere se una delle due parti può interrompere l'altra oppure se una delle due si dovrà comportare da *slave* nei confronti dell'altra che farà da *master*. Se invece la connessione logica è stata già attivata,

occorre esaminare lo stato della stessa, per stabilire se i dati preparati precedentemente possono essere trasmessi subito oppure è necessario attendere: ad esempio, nel caso di una *connessione master-slave* con tecnica di *poll*, una stazione slave deve aspettare che la stazione master effettui la sua interrogazione (cioè richieda se qualcuno deve trasmettere);

- 4) **livello transport**: prima ancora di effettuare la trasmissione, devono essere definiti una serie di dettagli tecnici, che dovranno essere in accordo con le regole del dialogo; ad esempio, tra questi dettagli tecnici citiamo il numero di sequenza del messaggio, la specifica se esso può essere suddiviso in rete durante la trasmissione, i provvedimenti da prendere se il messaggio arriva errato e altro ancora.
- 5) **livello di network**: appurati i dettagli tecnici della trasmissione, perché questa possa avvenire è necessario scegliere il percorso effettivo dei dati in rete, a meno che la scelta non sia stata fatta precedentemente e per tutti i messaggi della connessione. Ad esempio, possiamo pensare a 2 stazioni che possono essere collegate o tramite una *linea dedicata*, nel qual caso la scelta del percorso non si pone, oppure tramite un percorso di rete (utilizzando perciò dei *nodi intermedi*) che può variare ogni volta, cioè per ogni connessione, o anche nel corso della trasmissione stessa. Opportuni algoritmi di scelta (*instradamento*) determinano il percorso basandosi sulle strade di rete esistenti;
- 6) **data link controllo** (o *livello di protocollo di linea*): l'ultimo passo, prima della trasmissione vera e propria, è quello di strutturare il messaggio secondo il formato previsto dal protocollo utilizzato sulla linea in uscita. Vanno anche definite le funzioni di controllo della trasmissione;
- 7) **livello di interfaccia fisica**: a questo punto, la trasmissione del messaggio può finalmente avvenire, per cui il messaggio viene passato all'*adattatore di linea*, il quale provvede ad inviare, uno alla volta, i singoli bit, in accordo con l'interfaccia fisica della linea utilizzata. La trasmissione avviene tramite una collaborazione dell'adattatore con il dispositivo DCE che collega il sistema alla linea trasmissiva (ad esempio il *modem*).

Lo schema appena descritto è molto generale. Di volta in volta, ad esso si possono aggiungere delle funzioni opzionali, tra le quali citiamo soprattutto la *cifratura dei dati*, i processi di *autenticazione* della controparte remota e simili.

Il modello OSI è intervenuto a formalizzare e generalizzare la sequenza logica descritta poco fa, includendo anche una discreta varietà di funzioni opzionali.

IL MODELLO DI RIFERIMENTO ISO/OSI

Il modello OSI può essere rappresentato così come fatto nella *figura 1* o, ancora meglio, con indicato nella figura seguente:

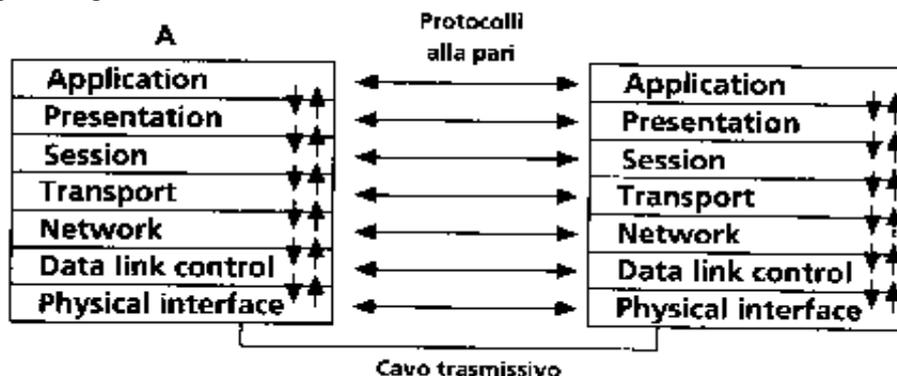


Figura 5 - Schema logico di connessione tra due utenti secondo il modello ISO/OSI.

Quando c'è da effettuare una trasmissione tra i due utenti, ogni livello funzionale (**layer**) chiamato in causa effettua sostanzialmente tre operazioni:

- in primo luogo, esegue le funzioni richieste;
- successivamente, aggiunge a quanto ricevuto una **testata funzionale (header)**, che è specifica del proprio livello ed è destinata ad essere interpretata dal proprio omologo sul sistema remoto. Questa testata serve per ottenere la cooperazione della controparte funzionale remota oppure per inviare una semplice informazione;
- infine, passa al proprio *strato subordinato* sia ciò che ha preparato sia l'opportuno comando funzionale.

Ciascun livello è attivato a partire dal livello funzionale più elevato, che è quello applicativo (*application layer*); ogni livello, esaurite le proprie funzioni, chiama in causa il proprio subordinato, passandogli i dati ed un opportuno comando. Questo vale per tutti i livelli, il che significa che *i dati ed il comando originati dal livello applicativo si arricchiscono, man mano che si scende verso il basso, di testate funzionali*. Particolare è il comportamento degli ultimi due livelli:

- il penultimo livello, cioè quello del *protocollo di linea*, aggiunge la propria testata ma aggiunge anche una *coda* al messaggio, nella quale è incluso il campo per il controllo degli errori in trasmissione (**FCS**, *Frame Check Sequence*);
- l'ultimo livello, quello dell'interfaccia fisica con il mezzo trasmissivo, non aggiunge nulla al messaggio, almeno nella maggior parte dei casi, ma si occupa solo della trasmissione "brutale" dei bit sulla linea di trasmissione utilizzata.

La figura seguente chiarisce come risulta composto un messaggio che ha seguito i passaggi appena descritti:

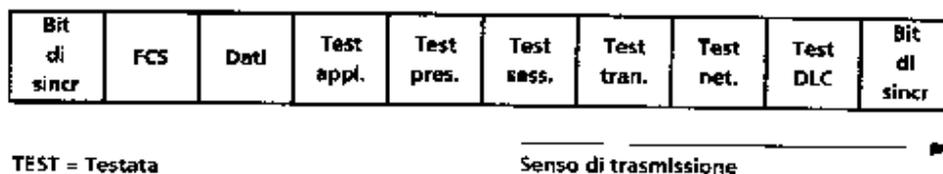


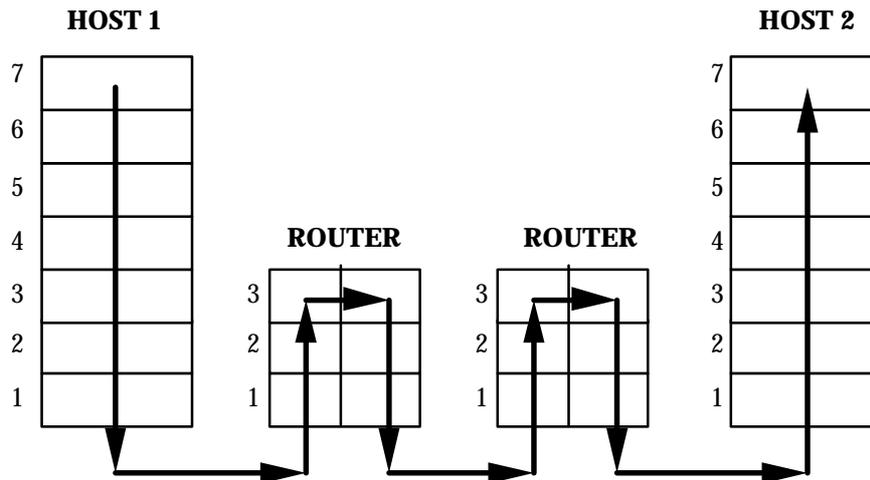
Figura 6 - Struttura del generico messaggio generato dai 7 livelli del modello ISO/OSI

Come evidenziato dalla freccia indicante il senso di trasmissione, *il messaggio viene inviato in linea con i vari campi disposti in senso inverso a quello con cui sono stati generati*. Questo serve in sede di ricezione: il sistema remoto (destinatario), infatti, riceve il messaggio al livello inferiore e lo ricostruisce bit per bit passandolo allo strato superiore, quello del controllo di linea; ogni livello, nel passaggio del messaggio verso l'alto, interpreta solo la testata di propria competenza, esegue le proprie funzioni in accordo con quanto specificato nella testata, memorizza eventuali risultati (da utilizzare, ad esempio, per una risposta al messaggio), elimina la propria testata e passa allo strato superiore ciò che rimane.

In tal modo, al livello applicativo arrivano solo i dati ed il comando incluso nella *testata applicativa*, comando che riguarderà l'azione da eseguire sui dati.

A questo punto, il sistema può preparare la propria risposta o, eventualmente, una propria richiesta, dopo di che il procedimento si ripete, ovviamente a ruoli invertiti.

Uno schema logico esemplificativo di quanto detto prima è il seguente:



Qui sono evidenziati non solo la sequenza logica con cui il messaggio viene costruito nell'host numero 1 e poi "interpretato" dall'host numero 2, ma anche il fatto che il messaggio, per la trasmissione vera e propria, ha generalmente bisogno di attraversare un certo *percorso di rete* prima di giungere al destinatario.

Il percorso di rete sarà composto da un certo numero di *nodi intermedi* (detti anche *elementi di commutazione* o **router**⁴) e sicuramente ciascuno dei due host (sorgente e destinazione) è collegato alla rete tramite un proprio nodo intermedio.

DESCRIZIONE DEI SINGOLI LIVELLI DEL MODELLO ISO/OSI

Il modello OSI può essere discusso da vari punti di vista. Una prima descrizione è quella basata sulla figura 1, che riportiamo nuovamente:

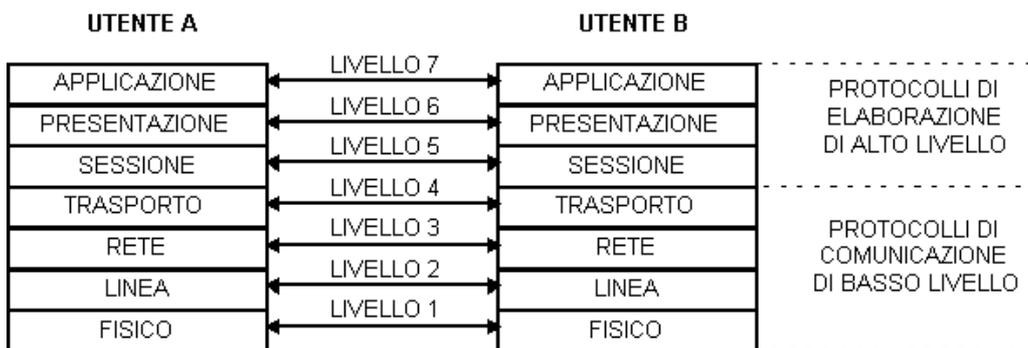


Figura 7 - Schema logico di due utenti (A e B) connessi tramite il modello OSI

In questa figura è evidenziata una possibile suddivisione dei 7 livelli:

⁴ 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.

- i quattro livelli inferiori (interfaccia fisica, protocollo di linea, network e transport) hanno funzioni prevalentemente trasmissive, cioè si occupano essenzialmente delle tecniche correlate alla trasmissione propriamente detta;
- al contrario i tre livelli superiori (session, presentation e application) sono caratterizzati da funzioni di interazione tra le due applicazioni o i due utenti finali, mentre non intervengono nelle tecniche trasmissive.

E' possibile però vedere la cosa da un altro punto di vista, leggermente diverso. Infatti, se consideriamo gli ultimi 3 livelli (livello di rete, livello di protocollo di linea, livello di interfaccia fisica), essi hanno una importante caratteristica: hanno funzioni che riguardano tutti i *componenti di rete* utilizzati durante una connessione, includendo, tra tali componenti di rete, anche eventuali nodi intermedi (o elementi di commutazione o **router**). Al contrario, le funzioni dei quattro livelli superiori (dal livello di transport al livello applicativo) non riguardano eventuali nodi intermedi, ma stabiliscono solo una cooperazione diretta tra i due sistemi interessati alla connessione. Possiamo allora proporre una ulteriore schematizzazione del modello:

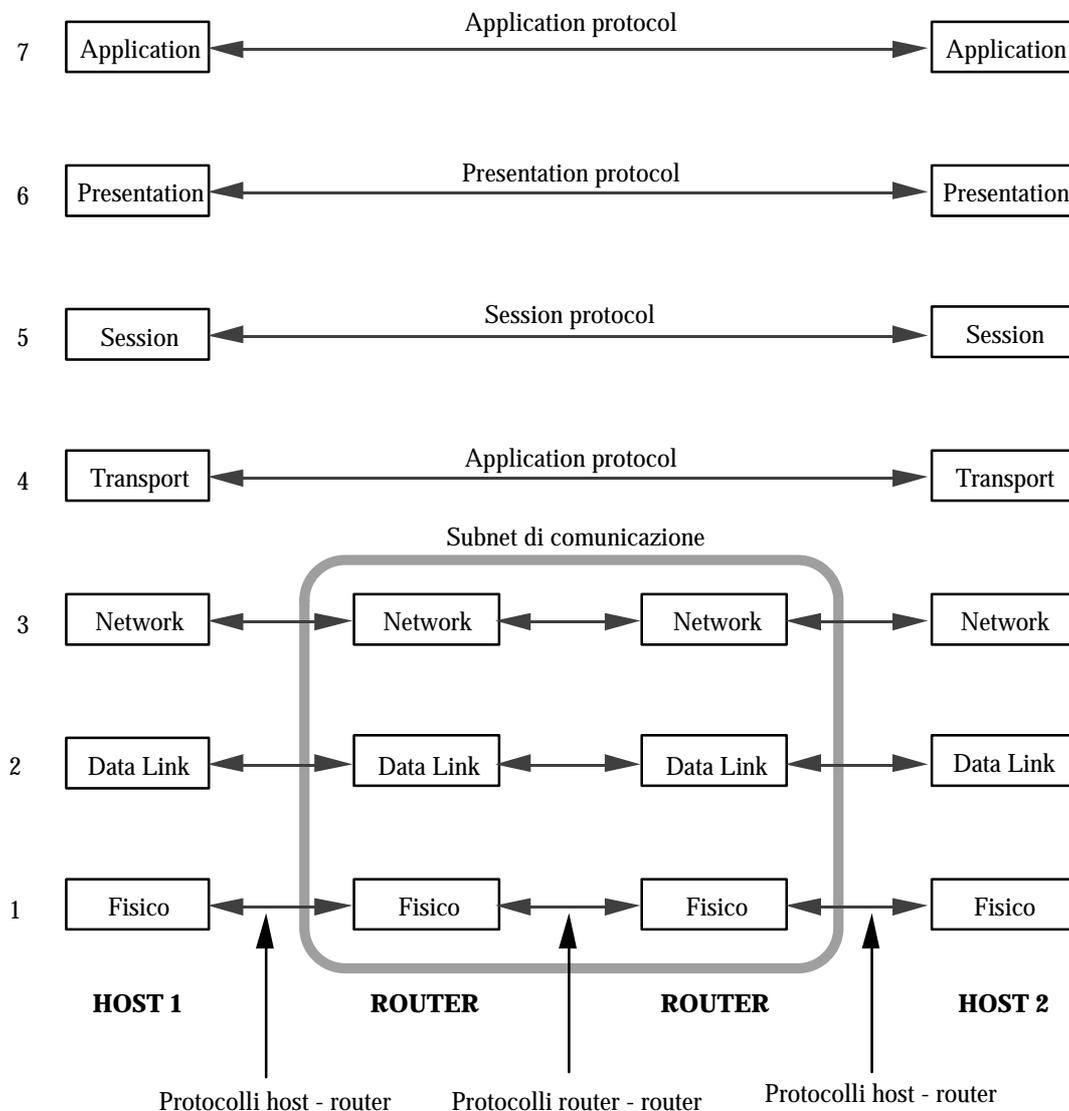


Figura 8 - Schema logico dettagliato della connessione di due utenti secondo il modello di riferimento OSI. Viene qui evidenziata una suddivisione tra i 3 livelli inferiori, comuni alla subnet di comunicazione, e i 4 livelli superiori, che appartengono solo ai due sistemi remoti

Come si nota, solo i tre livelli inferiori hanno a che fare con la cosiddetta **subnet di comunicazione** (*communication subnet* o semplicemente *subnet*): essa comprende tutto ciò che serve a connettere tra loro i due utenti finali (i cosiddetti **end system**), in quanto ha il compito di trasportare messaggi da un end system all'altro, così come il sistema telefonico trasporta parole da chi parla a chi ascolta.

Si dice allora che i tre livelli inferiori hanno funzioni di tipo **box-to-box**, mentre quelli di livello superiore hanno funzioni di tipo **end-to-end**.

E' chiaro che la *relazione box-to-box*, tipica dei 3 livelli inferiori, ha una notevole importanza pratica: essa infatti implica che i protocolli usati a questo livello debbano essere comuni a tutte le *box intermedie* del percorso. Questo non è necessario, invece, per i 4 livelli superiori: infatti, per una *sessione* (o dialogo o connessione) tra i due sistemi, è necessario e sufficiente che le norme a questi 4 livelli siano comuni solo ai due sistemi finali.

Detto in altre parole, eventuali nodi intermedi dovranno essere coerenti, con i due sistemi finali, solo per quanto riguarda gli strati trasmissivi: questo è garantito dall'uso di ciò che in figura è stato indicato come **protocolli host-router**. Non solo, ma è chiaro, come si evince dalla figura, che i livelli superiori non sono chiamati in causa nei nodi intermedi; di conseguenza, i nodi intermedi, qualora avessero capacità elaborative autonome, potrebbero sicuramente usare livelli funzionali di tipo end-to-end del tutto diversi da quelli della connessione per cui operano come nodi di transito. Infine, nella comunicazione tra router e router, non è necessario che la comunicazione avvenga con gli stessi protocolli usati nella comunicazione tra router e host, ma possono essere usati altri protocolli (**protocolli router-router**), purché compatibili con i precedenti.

Da quanto detto, si evince che, tra i 7 livelli, una posizione particolare è assunta dal *livello di transport*, che si trova esattamente a metà delle due classificazioni fatte prima: si è visto, infatti, che tale livello da un lato ha funzioni correlate alla trasmissione ed è quindi assimilabile ai 3 livelli inferiori, ma, dall'altro lato, ha anche funzioni riguardanti il rapporto tra i due sistemi interessati alla connessione, il che lo rende assimilabile anche ai 3 livelli superiori, quelli che gestiscono il dialogo. Possiamo allora affermare quanto segue:

- il compito primario del livello di transport è quello di scegliere le più idonee procedure di controllo della trasmissione, affinché il traffico venga gestito in modo appropriato. Esso deve tener conto sia delle esigenze applicative sia delle caratteristiche di qualità della rete trasmissiva utilizzata; a questo scopo, il livello di transport mette a disposizione dell'utente un certo numero di **classi di trasporto**, cioè diverse modalità (ciascuna con le proprie opzioni) per il controllo sulla trasmissione e l'utente può scegliere quella che preferisce;
- sempre in termini di controllo sulla trasmissione, il livello di transport si occupa anche delle procedure per la gestione degli errori, della sequenza dei messaggi, della lunghezza dei messaggi e simili.

Concentriamoci adesso sui 3 livelli inferiori (che possiamo dire costituiscano un **sottosistema trasmissivo**); essi si occupano sostanzialmente di 3 cose:

- scelta del percorso all'interno della rete (*livello network*);
- scelta del protocollo di linea da usare (*livello di protocollo di linea*), che per esempio può essere il protocollo *HDLC* nel caso di una rete geografica (WAN);
- scelta dell'interfaccia fisica da utilizzare (*livello fisico*).

Queste scelte trovano chiaramente applicazione sulla linea trasmissiva che collega l'*host sorgente* al primo *nodo di rete* lungo il percorso scelto nonché anche sulla linea che collega l'ultimo nodo di

rete con il sistema finale (*host destinatario*); tra i nodi intermedi, possono essere applicate le stesse norme oppure, come si è detto prima, norme diverse (purché compatibili con quelle *host-router*).

La scelta del percorso di rete è un problema molto importante. Il modello OSI contiene un esplicito suggerimento in proposito: viene infatti suggerito, nel caso di trasmissione a lunga distanza, che le funzioni dei 3 livelli inferiori corrispondano alle specifiche necessarie per la connessione ad una *rete pubblica a commutazione di pacchetto*. Tale specifiche (che vanno sotto il nome di **X25**) sono state a loro volta definite come *raccomandazione* (in pratica si tratta ormai di uno standard) dal **comitato ITU-TSS** (*International Telecommunication Union - Telecommunication Standard Support*), che sarebbe il nuovo nome del vecchio **CCITT** (*Consultative Committee for International Telegraph and Telephone*).

Per comprendere meglio la questione, facciamo un esempio, con riferimento alla figura seguente:

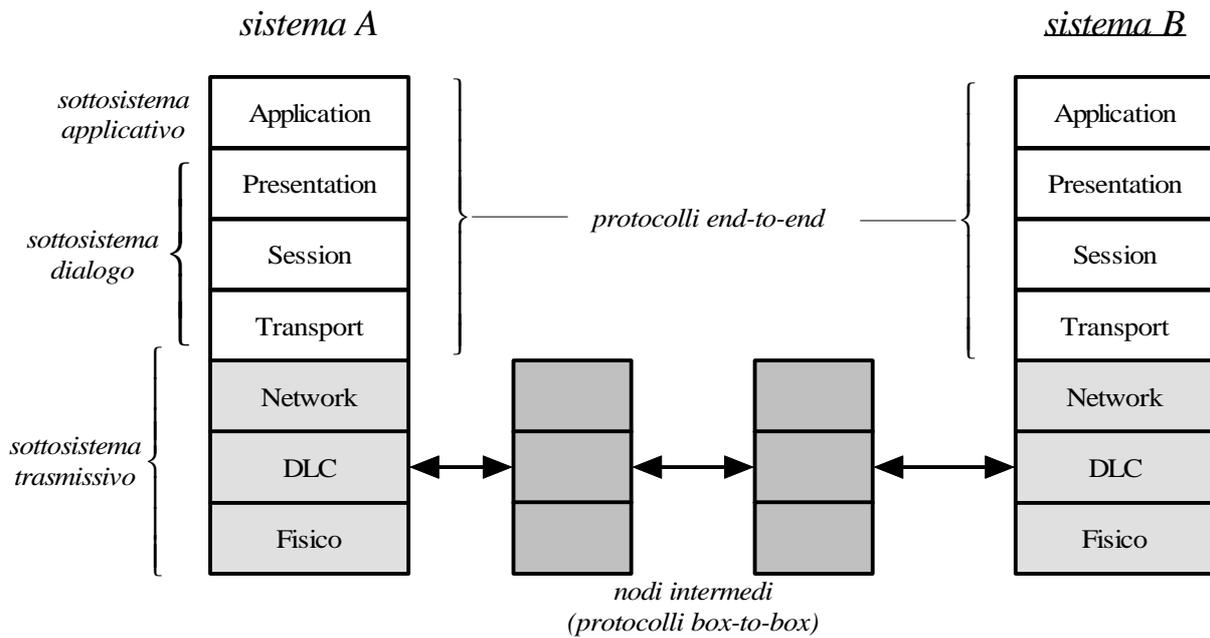


Figura 9 - Protocolli OSI end-to-end (per i sottosistemi applicativo e di dialogo) e box-to-box (per la subnet di comunicazione, cioè il sottosistema trasmissivo)

Supponiamo che i due sistemi (A e B) di figura siano installati a Roma ed a Milano e che sia possibile porli in comunicazione tramite due diverse strade: la prima strada prevede due nodi intermedi, situati a Firenze e Bologna, mentre la seconda strada prevede un unico nodo intermedio, situato a Genova:

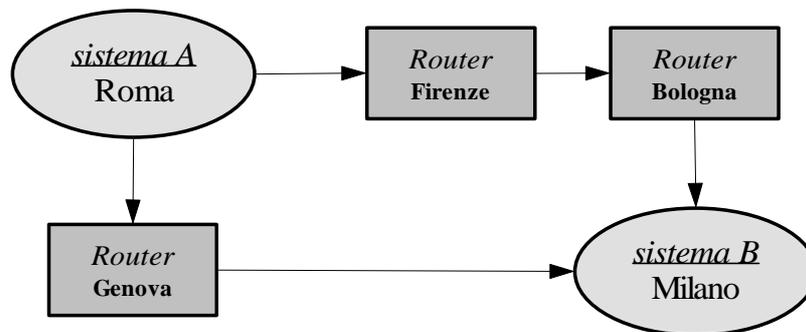


Figura 10 - Connessioni tra due utenti attraverso due distinti percorsi di rete

Quando i due sistemi di Roma e di Milano vogliono entrare in connessione (o *in sessione*), è necessario scegliere una delle due alternative: Roma→Firenze→Bologna→Milano oppure Roma→Genova→Milano. Si usa (a livello *network*) un apposito **algoritmo di instradamento**, che vale per tutta la rete, il quale decide, in base a vari fattori (come la conoscenza della topologia di rete e lo stato delle varie risorse dei percorsi possibili) quale percorso utilizzare.

Il modello ISO/OSI non impone alcun algoritmo specifico di instradamento, ma suggerisce solo di affidarsi alle specifiche, relative al livello *network*, contenute nella **raccomandazione X25**. Quest'ultima, a sua volta, si limita a specificare come ottenere il servizio trasmissivo dalla rete, ma lascia la massima libertà al progettista della rete di commutazione di pacchetto di scegliere l'algoritmo di instradamento. Ai livelli inferiori si applicano le norme del protocollo di linea standard (ad esempio HDLC) e quelle dell'interfaccia fisica scelta per tutte le linee utilizzate nel percorso di rete (ad esempio X21 o X21bis).

In generale, possiamo riepilogare, per il *livello network*, le seguenti incombenze:

- **routing**, cioè scelta del cammino da utilizzare. Può essere statico (cioè fissato ogni tanto e raramente variabile) oppure dinamico (cioè continuamente aggiornato, anche da un pacchetto all'altro come accade per *Internet*);
- **gestione della congestione**: a volte troppi pacchetti arrivano ad un router (ad esempio quando questo dispone di molte linee in ingresso ed un'unica linea di uscita) ed è perciò necessario gestire questo sovraffollamento per limitare al minimo o evitare del tutto la perdita di pacchetti;
- **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: potrebbero infatti presentarsi le esigenze di rimappare indirizzi, frammentare ulteriormente i pacchetti, gestire protocolli diversi.

Al di sotto del livello *network* c'è il livello di protocollo di linea (*data link control*). Lo scopo di questo livello è essenzialmente far sì 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** (sarebbe l'equivalente dei *pacchetti* per il livello *network*)
- invia i frame in sequenza;
- aspetta una conferma positiva (**ACK**) per ogni frame inviato.

Ci sono allora una serie di incombenze da soddisfare:

- aggiunta di delimitatori (**framing**) all'inizio ed alla fine di ciascun frame;
- **gestione di errori** di trasmissione causati da errori di ricezione, perdita di frame, perdita di ACK (con conseguente duplicazione di un frame);
- **regolazione del traffico** (per impedire che il ricevente sia "sommerso" di dati);
- meccanismi per l'invio degli ACK.

Il livello più basso è quello *fisico*, che quindi ha a che fare con la trasmissione di bit "grezzi" su un canale di comunicazione. E' ovvio che tale livello abbia uno strettissimo legame con le caratteristiche meccaniche, elettriche e procedurali delle **interfacce di rete** (componenti che connettono l'elaboratore al mezzo fisico) e con le caratteristiche del mezzo fisico.

Fissate le norme relative ai 3 livelli inferiori, si passa al già descritto *livello di transport*, che per primo interviene nella trasmissione tra i due sistemi finali della connessione. In questo senso, il livello transport è il primo livello realmente **end-to-end**, cioè da host sorgente a host destinatario.

Come già detto, il livello di transport permette all'utente di scegliere alcune modalità di controllo della trasmissione (le citate *classi di servizio*), in relazione all'affidabilità della rete trasmissiva usata.

Continuando verso l'alto, il *livello session* si occupa di vari aspetti, tra cui citiamo:

- apertura della connessione logica (o *sessione*) tra le due applicazioni;
- scelta delle modalità di dialogo (ad esempio, sceglie se usare flussi bidirezionali alternati o simultanei e indica come gestire il traffico urgente oltre a quello non urgente) e quelle di sincronizzazione tra le due applicazioni;
- definizione di possibilità e modalità di ripartenza del dialogo in caso di caduta della connessione.

Il *livello presentation* deve invece strutturare i dati e i comandi ricevuti dal livello applicativo, in modo da renderli comprensibili ed elaborabili da parte del sistema remoto. Il suo compito è dunque la strutturazione formale dei dati, la loro codifica e altre funzioni opzionali quali la cifratura dei dati.

Infine, il *livello applicativo* è responsabile dei servizi applicativi per l'utente finale. Si può affermare che, al contrario degli altri livelli, il livello applicativo si interessa soprattutto al *livello semantico* dei dati scambiati. Esso contribuisce a definire due aspetti:

- una **interfaccia applicativa (ACSE, Application Control Services Element)** comune a tutti i servizi;
- un certo numero di **servizi di dialogo** progettati per le diverse esigenze applicative: per il trasferimento dei file, per il trasferimento delle incombenze tra sistemi, per l'accesso a database remoti, per l'uso di servizi di messaggistica elettronica, per l'interazione di due processi applicativi.

PRIMITIVE OSI

Consideriamo ancora una volta lo schema generale del modello ISO/OSI:

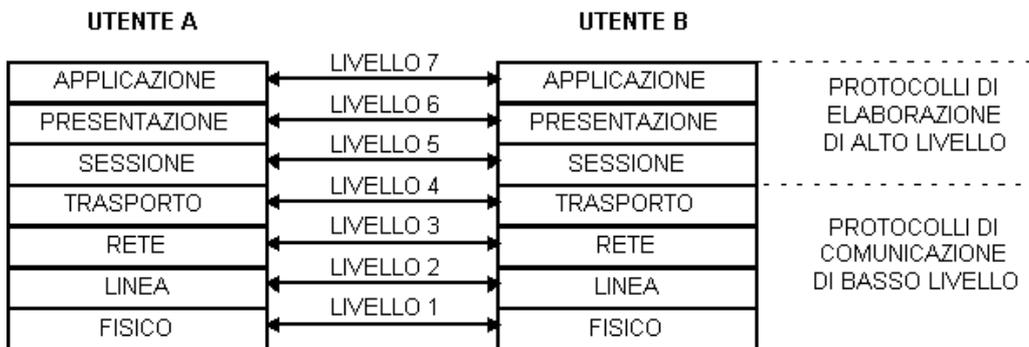


Figura 11 - Schema logico (già proposto nella figura 1) di due utenti (A e B) connessi tramite il modello ISO/OSI

Quando viene richiesto un determinato servizio ad uno strato di livello N, la richiesta viene effettuata tramite uno scambio, tra livello N e livello N+1, di opportune **primitive**⁵ e questo scambio avviene tramite l'**interfaccia** tra i due livelli.

Quando un livello N+1 e quello subordinato N interagiscono tra loro, usano 4 tipi di *primitive*:

- **request**: questa primitiva viene utilizzata dal livello N+1 per richiedere un determinato servizio al livello N, che quindi è il livello *fornitore* del servizio;
- **indicate**: questa viene usata dal livello N per informare il livello N+1 che un determinato servizio è stato invocato dal corrispondente livello N remoto oppure da una entità N locale;
- **response**: questa primitiva viene usata dal livello N remoto per completare la procedura di cui sopra;
- **confirm**: la risposta del livello N remoto arriva fino alla controparte N locale, quella cioè ove si è attivato l'intero ciclo. Tramite la *confirm*, il livello N locale informa lo strato superiore N+1 e il ciclo si completa.

Un servizio viene dunque attivato mediante il ciclo appena descritto, al quale si aggiungono altre due cose:

- in primo luogo, l'indicazione del servizio richiesto: ad esempio, l'indicazione **connect request** viene usata per stabilire una connessione con un altro sistema OSI remoto;
- in secondo luogo, una lista di parametri che specificano le caratteristiche peculiari del servizio richiesto: ad esempio, per una richiesta di connessione, potrebbe essere specificato se si richiede una connessione unidirezionale o bidirezionale, come anche sono possibili indicazioni sulla qualità del servizio richiesto oppure indirizzi di memoria dei dati da trasmettere e così via.

Quando un livello N+1 deve richiedere un servizio al livello N, deve necessariamente conoscere l'**indirizzo** del livello N, in modo da poter indirizzare la richiesta. Tale indirizzo costituisce il cosiddetto **SAP** (*Service Address Point*) del livello N. Quindi, ogni livello è caratterizzato da un proprio *SAP*. Nel caso in cui l'erogazione del servizio coinvolge non solo lo strato N locale, ma anche uno strato N remoto, allora devono essere chiaramente noti i SAP di entrambi i livelli.

In generale, quando due livelli comunicano tra loro, vengono utilizzati 5 tipi di informazioni:

- **SDU** (*Service Data Unit*): sono i dati dell'utente, trasferiti dal livello N+1 al livello N e successivamente al livello N-1;
- **PCI** (*Protocol Control Information*): sono le informazioni destinate al livello remoto corrispondente, per chiedere o inviare informazioni o servizi;
- **PDU** (*Protocol Data Unit*): è semplicemente l'insieme di SDU e PCI;
- **ICI** (*Interface Control Information*): è un insieme *temporaneo* di parametri passato dal livello N+1 al livello N per invocare un servizio;
- **IDU** (*Interface Data Unit*): questo è semplicemente il totale delle informazioni passate dal livello N+1 al livello N per richiedere il servizio; si tratta perciò dell'insieme di SDU, PCI e ICI. Tale insieme va dunque indirizzato al SAP del livello N.

⁵ A tal proposito, conviene ricordare subito che, nel modello OSI, le definizioni delle *primitive* sono puramente astratte: esse infatti indicano i tipi di servizio che possono essere richiesti da ciascun livello, ma non le specifiche di realizzazione. Questo sempre per il fatto che il modello OSI è un modello di riferimento e non un modello concreto.

Vediamo allora qual è il percorso seguito dalle informazioni appena descritte, basandoci sullo schema seguente:

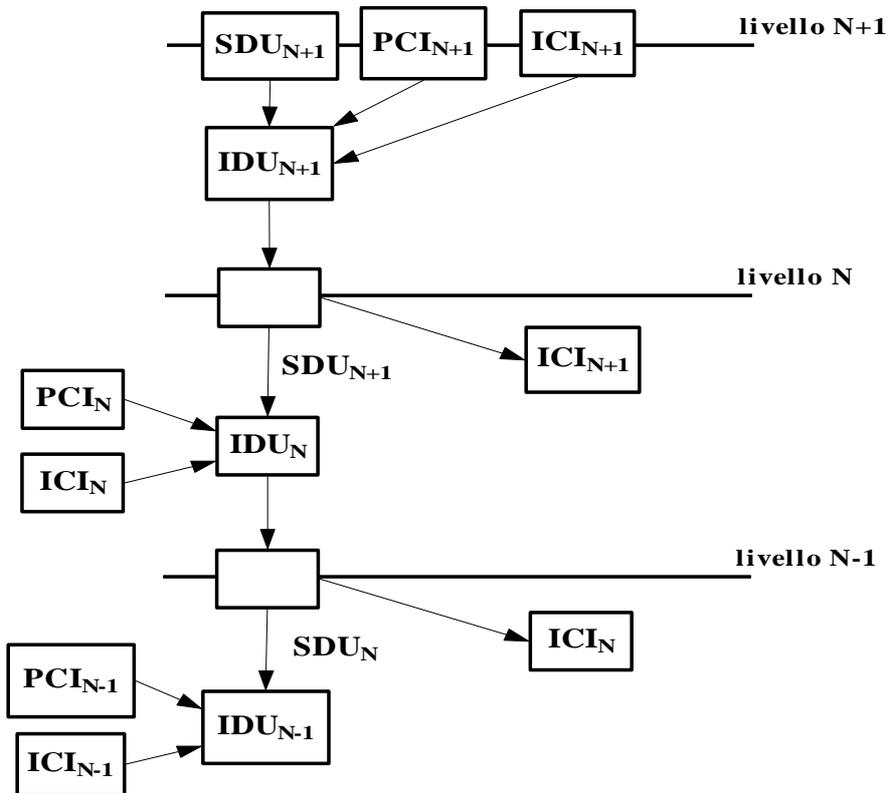


Figura 12 - Scambio di informazioni tra livelli OSI: ciascun livello riceve delle informazioni dal livello superiore, utilizza quelle che servono solo a lui (e non agli altri livelli), dopo di che le cancella e ne aggiunge delle altre da mandare al livello sottostante.

La figura indica quanto segue:

- quando il livello N+1 deve chiedere un servizio al livello sottostante, costruisce i campi SDU, PCI e ICI, che mette insieme nell' IDU da passare al livello sottostante;
- quando il livello sottostante riceve l'IDU dal livello superiore, usa le informazioni contenute nell' ICI e nel PCI al fine di identificare ed erogare il servizio richiesto, dopo di che elimina l'ICI; restano, perciò, a questo punto solo l'SDU e il PCI, che diventano (insieme) un nuovo SDU, da passare al livello sottostante (N-1); ovviamente, il nuovo SDU viene arricchito di un nuovo insieme ICI e da un nuovo insieme PCI, in modo da avere un nuovo IDU da passare al livello sottostante⁶.

⁶ In concreto, l'aggiunta del PCI da parte di ogni livello si concretizza nell'aggiunta di una testata (detta anche *header*) a quanto ricevuto dal livello superiore.

NOTE SUL MODELLO OSI

Facciamo adesso alcune considerazioni aggiuntive e generali a proposito del modello OSI finora descritto.

- Uno schema funzionale analogo a quello del modello OSI (composto quindi da 7 livelli) è caratteristico di quasi tutte le **architetture proprietarie**⁷, incluse anche quelle precedenti allo stesso modello OSI. Probabilmente, *la completezza e la precisione del modello OSI sono addebitabili anche al fatto che la sua definizione è stata facilitata dall'attento esame di strutture architettoniche precedenti*. Rispetto a queste, il modello offre però un maggiore rigore e una maggiore ricchezza di funzioni ed opzioni.
- Spesso, si parla del modello OSI come una architettura standard (“*standard de iure*”) di rete. In effetti, il modello è solo una guida standard alle relazioni funzionali tra due sistemi in una relazione alla pari. Dal punto di vista architettonico, vanno invece fatte alcune osservazioni: ad esempio, il modello OSI definisce le relazioni tra 2 sistemi, ma non quelle tra un numero M (>2) di sistemi; non da cioè una guida per le relazioni tra parecchi sistemi in rete. Tanto per fare un esempio, il modello OSI non definisce ruoli e funzioni di componenti intermedi di rete, come i *cluster controller*, i *gateway*, i *router*, le *unità di bridge* e così via, né suggerisce schemi specifici per la funzione di instradamento, che invece è fondamentale nelle reti reali. Come abbiamo già osservato in precedenza, la funzione di instradamento riguarda il livello 3, cioè il *livello network*: allora, il modello OSI si limita a prevedere, per le reti geografiche (WAN), l'uso di *reti a commutazione di pacchetto* coerenti con il già citato *standard X25*. Ogni rete, tra l'altro, che adotta lo standard X25, ha comunque dei propri algoritmi di instradamento, dato che lo standard X25 non ne indica di specifiche. Questo discorso chiarisce che *il modello OSI, se da un lato lascia una notevole libertà nelle scelte delle reti da adottare, dall'altro resta comunque solo uno standard per relazioni funzionali alla pari tra due sistemi aperti, più che una vera e propria architettura di rete*.
- Grazie al fatto che ogni livello funzionale si serve delle funzioni dei livelli subordinati e che questo vale per ogni livello, *il risultato pratico dello schema è che ogni livello vede l'insieme dei livelli inferiori come una grande black box, alla quale richiedere servizi e dalla quale ricevere informazioni e risultati, ma della quale non è necessario conoscere il funzionamento*. Questo discorso vale per tutti i livelli e quindi, a maggior ragione, per il livello più alto, quello applicativo. E' evidente che, in questo modo, l'utente può chiedere ed ottenere dialoghi con controparti remote conoscendo solo i formalismi necessari a formulare le richieste e disinteressandosi completamente delle complessità tecniche (peraltro sempre crescenti) del sistema comunicativo e trasmissivo sottostante.
- Sempre nel modello OSI, potrebbe essere fonte di confusione il fatto che diverse funzioni sono ripetute a vari livelli. Ad esempio, il controllo di sequenza viene effettuato sia a livello 2 (protocollo di linea) sia a livello 4 (transport). In realtà, i due controlli hanno compiti diversi:
 - * al livello di protocollo di linea, l'obiettivo è controllare che non vengano persi messaggi trasmessi su quella specifica linea che si sta utilizzando, qualunque sia la connessione logica cui appartiene il messaggio. Di solito, una linea non è riservata ad una sola connessione, per cui è normale che su di essa viaggino messaggi appartenenti a diverse connessioni logiche. Allora, il protocollo di linea assegna numeri di sequenza

⁷ Per il significato dell'espressione *architettura proprietaria* rimandiamo all'ultimo paragrafo di questo capitolo.

e controlli per tutti i messaggi, da qualunque sistema arrivino e a qualunque sistema sia destinati;

- * a livello di transport, invece, il controllo di sequenza ha scopi diversi: in primo luogo, esso tiene sotto controllo la sequenza specifica della singola connessione; in secondo luogo, non gli basta il controllo di sequenza effettuato dal protocollo di linea, in quanto un dato messaggio potrebbe essere stato trasmesso bene su una linea del percorso e male sulla successiva⁸. Quindi, il livello di transport effettua un controllo sulle singole connessioni e, in particolare, sulle connessioni end-to-end, ossia da utente finale ad utente finale, sull'intero percorso seguito dai dati e non sulla singola linea.

OSI: STATO ATTUALE

Come si può intuire facilmente, il modello OSI è funzionalmente completo, bene definito e ricco di opzioni, per cui offre possibilità di scelta che lo rendono abbastanza flessibile. Già da molto tempo, i livelli dal primo (interfaccia fisica) al sesto (presentazione) sono stati resi standard; a livello applicativo, invece, sono state standardizzate solo tre funzioni:

- *trasferimento e manipolazione file (FTAM, File Transfer And Manipulation)*;
- *posta elettronica (X400)*;
- *servizi di directory (X500)*.

Di conseguenza, nei sistemi con funzioni comunicative di base coerenti con i livelli 1-6 del modello OSI (parliamo quindi della maggior parte dei sistemi presenti sul mercato), le sole applicazioni OSI offerte riguardano la messaggistica elettronica (X400) ed il file transfer (FTAM), mentre le nuove applicazioni sono ancora da tradurre in prodotti commerciali.

L'aspettativa di mercato nei riguardi del modello OSI è stata notevole, dato che l'esigenza di comunicare con sistemi eterogenei era molto sentita e che il modello in costruzione era teoricamente convincente. Si fecero addirittura delle previsioni in base alle quali, in poco tempo, l'unica architettura di rete presente sul mercato sarebbe stata quella coerente con le specifiche OSI. Alcuni governi (USA e Inghilterra) emisero specifiche per l'adozione del modello OSI negli enti governativi (GOSIP, Government OSI Profile). Il ministero della difesa USA, che aveva emesso le **specifiche TCP/IP** come standard comunicativo interno, dichiarò che avrebbe abbandonato TCP/IP facendo migrare le proprie strutture e applicazioni verso il modello OSI. Questa massiccia accettazione di mercato non c'è stata, come è noto.

Il motivo è presto detto ed è legato alle limitate funzioni a livello applicativo: come detto, le sole funzioni applicative disponibili sul mercato sono quelle per il file transfer e la messaggistica elettronica, il che è troppo poco. La controprova è nel fatto che il TCP/IP, nato come standard interno del ministero della difesa statunitense per la comunicazione tra sistemi eterogenei, ha avuto una accettazione di mercato estremamente vasta, fino a proporsi come standard di fatto per le reti eterogenee.

Dato, quindi, che le funzioni applicative messe finora a disposizione non coprono la totalità delle esigenze comunicative del mercato, gran parte dei produttori ha deciso di affiancare OSI (senza soppiantarlo) con altre architetture di rete. Ci sono, ad esempio, sistemi che hanno opzioni per

⁸ Si pensi all'esempio fatto, nel capitolo 2, a proposito della comunicazione tra un terminale di Torino ed uno di Verona, attraverso un nodo intermedio situato a Milano: un protocollo di linea controlla solo la corretta ricezione dei dati da Torino (mittente) a Milano ed un secondo protocollo di linea controlla solo la corretta ricezione dei dati da Milano a Verona (destinatario); a tali controlli si aggiunge quello del protocollo di transport, che segnala direttamente al mittente la corretta ricezione dei dati da parte del destinatario.

diverse architetture: si usa l'architettura proprietaria per le comunicazioni tra sistemi dello stesso produttore, TCP/IP per le comunicazioni con i sistemi Unix e OSI per le applicazioni X400.

ARCHITETTURA DI RETE

Prima di concludere questo capitolo sull'analisi dettagliata del modello OSI, riportiamo alcune definizioni importanti, alcune delle quali citate nei paragrafi precedenti.

In primo luogo, osserviamo che l'insieme dei livelli e dei relativi protocolli che definiscono una rete di computer è detto **architettura di rete**.

La specifica dell'architettura deve essere abbastanza dettagliata da consentire la realizzazione di software e/o dell'hardware 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.

Esempio classico è *Internet Protocol Suite* (detta anche **architettura TCP/IP**).

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

Autore: **SANDRO PETRIZZELLI**
e-mail: sandry@iol.it
sito personale: <http://users.iol.it/sandry>
succursale: <http://digilander.iol.it/sandry1>