

Modello OSI e Architettura TCP/IP

Modello OSI

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

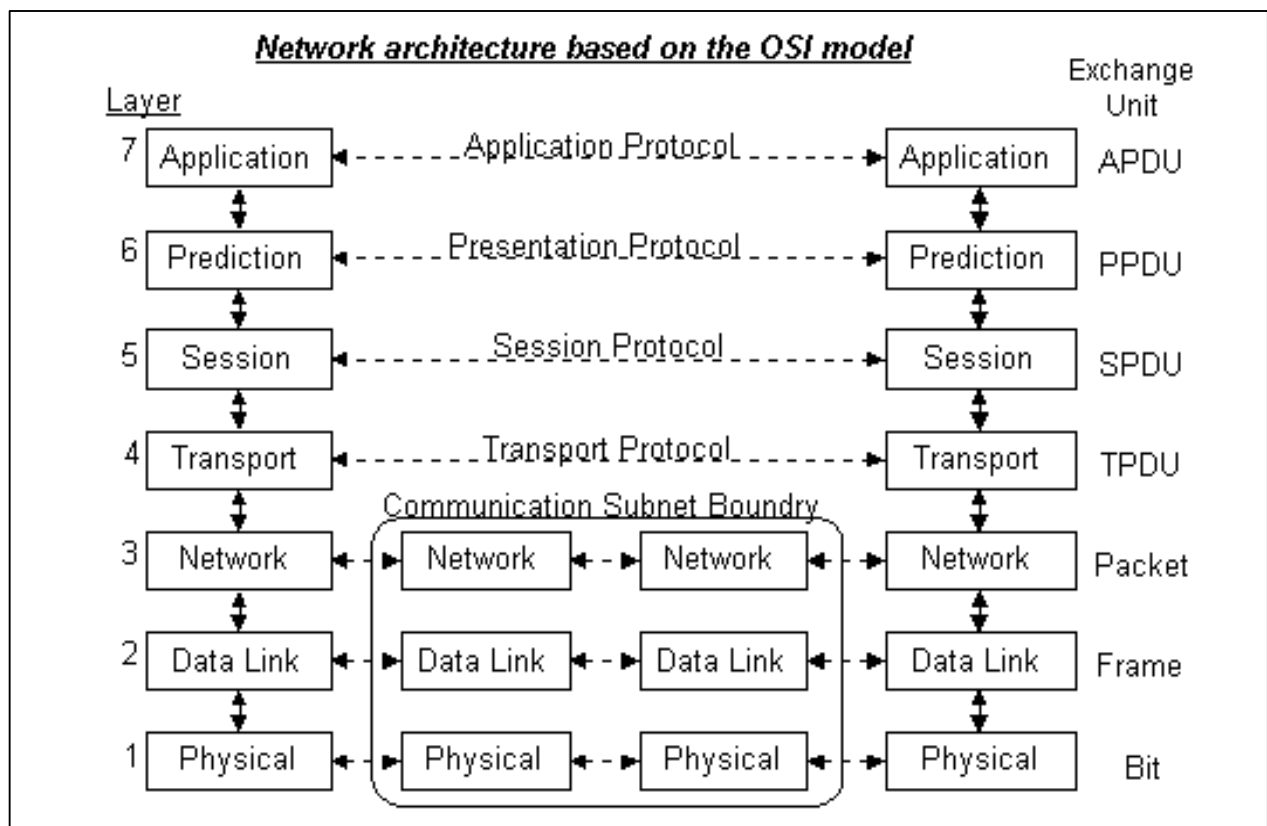
- fornire uno standard per la connessione di sistemi aperti, cioè in grado di colloquiare gli uni con gli altri;
- fornire una base comune per lo sviluppo di standard per l'interconnessione di sistemi;
- fornire un modello rispetto a cui confrontare le varie architetture di rete.

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

Principi di progetto seguiti durante lo sviluppo del modello OSI:

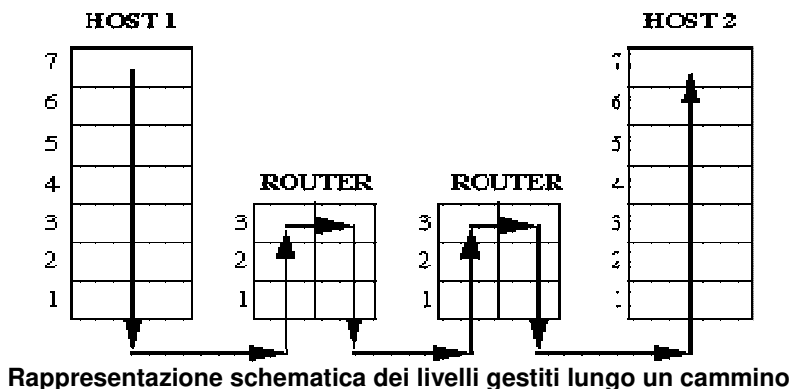
- ogni livello deve avere un diverso livello di astrazione;
- ogni livello deve avere una funzione ben definita;
- la scelta dei livelli deve:
 - minimizzare il passaggio delle informazioni fra livelli;
 - evitare:
 - troppe funzioni in un livello;
 - troppi livelli.

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).



Il modello OSI

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



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 fare in modo 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, cioè 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**, cioè 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**, cioè 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.

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;***
- ***trasferimento file;***
- ***posta elettronica.***

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

➤ Internet Protocol Suite (TCP/IP)

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:

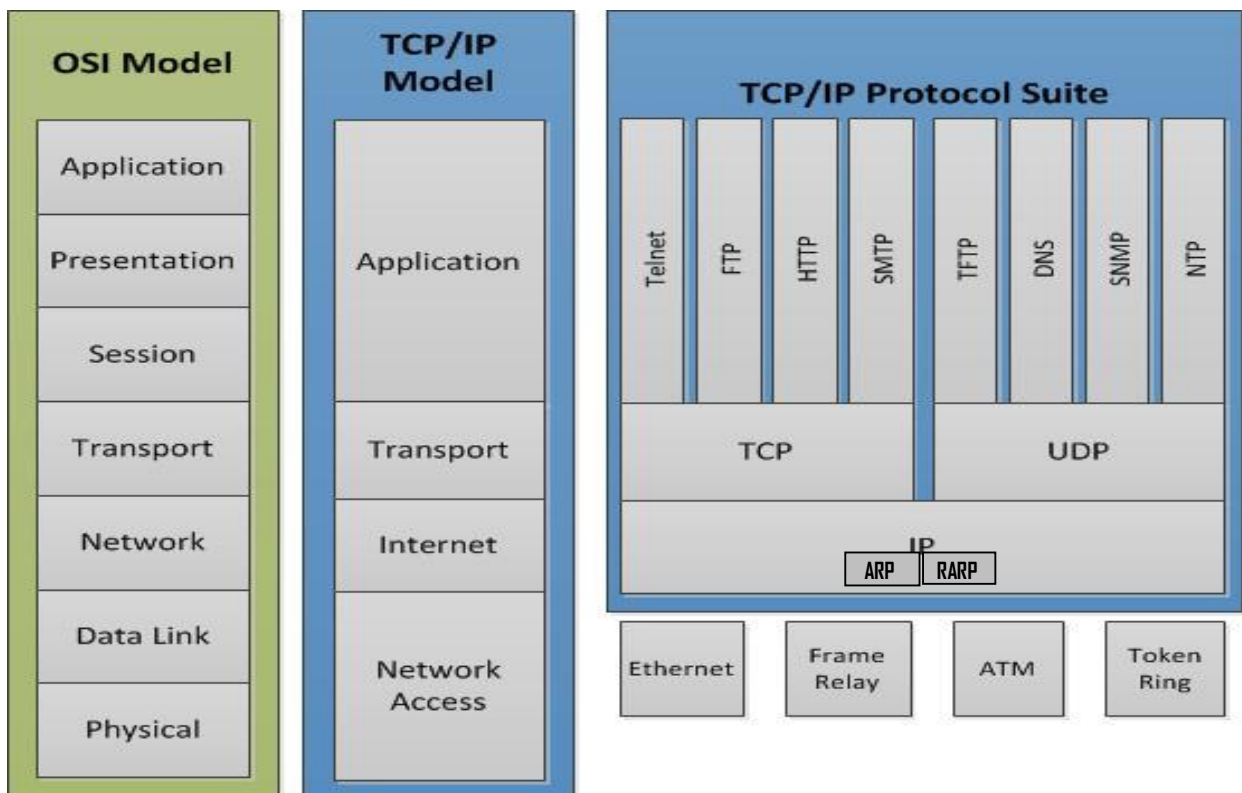


Figura 1-22: Relazione fra i livelli OSI e TCP/IP (alcuni protocolli della suite TCP/IP)

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 Network-Access

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

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

Livello Internet

Tra il livello precedente e questo operano due protocolli: ARP e RARP.

Dobbiamo infatti ricordare che i frame che provengono dal livello fisico e datalink (in questo caso livello Network-Access) utilizzano i cosiddetti MAC Address, ossia gli indirizzi fisici della NIC (Network Interface Card, la scheda di rete), mentre al livello Internet i pacchetti IP devono utilizzare non più gli indirizzi fisici MAC ma quelli logici IP.

Il protocollo RARP (*Reverse Address Resolution Protocol*), che fa il lavoro inverso del protocollo ARP, permette di conoscere l'indirizzo IP di una stazione partendo da una tabella di corrispondenza tra indirizzi MAC (indirizzi fisici) e indirizzi IP della stessa rete locale (LAN).

Il protocollo ARP (*Address Resolution Protocol*) invece permette di conoscere l'indirizzo fisico (MAC) di una scheda di rete corrispondente ad un indirizzo IP. Per far corrispondere gli indirizzi fisici a quelli logici, il protocollo ARP interroga i terminali di rete per conoscere i loro indirizzi fisici, poi crea una tabella di corrispondenza tra gli indirizzi logici e quelli fisici, salvandola in una memoria cache, che poi viene consultata ogni volta che dei pacchetti vengono inviati al livello fisico.

Il livello Internet tiene insieme l'intera architettura. Il suo ruolo è permettere ad un host di inviare 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)**: invio e ricezione 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 World Wide Web.

➤ **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 con i 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 cioè quelli TCP/IP si erano già considerevolmente diffusi. Le aziende non se la sono sentite di investire risorse nello sviluppo di un'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;
 - “non indispensabili” 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 hanno cominciato a pesare (ad es. indirizzi IP a soli 32 bit, 4 gruppi ciascuno di 8 bit, per cui si sta passando a indirizzi IPv6 a 128 bit, 8 gruppi ciascuno di 16 bit) .

In conclusione:

- OSI è **ottimo come modello**, mentre i suoi protocolli hanno avuto poco successo;
- TCP/IP (per ora) è **ottima come architettura di rete**, ma non può essere considerato un modello di riferimento.