

Modello ISO-OSI

1 Architetture di protocolli

Per ridurre la complessità e per ragioni di modularità, i protocolli di rete (e le reti stesse) sono organizzati in una serie di *strati* o *livelli*, ciascuno dei quali usa i servizi offerti dal livello inferiore. Si può allora stabilire una gerarchia tra i protocolli, che prende il nome di **architettura di protocolli**.

Ciascuna macchina (nodo) della rete implementa i vari livelli della gerarchia. Logicamente, il livello n di una macchina comunica con il livello n di un'altra macchina (questo è il motivo per cui si dice che un protocollo stabilisce lo scambio di informazioni tra entità paritetiche). In realtà, invece, tutti i dati passano dal livello n del mittente a quelli sottostanti, fino al livello fisico, che li trasmette concretamente sulla rete; quando essi arrivano alla macchina ricevente, “risalgono” infine livelli di quest'ultima, dal livello fisico al livello n . Queste interazioni tra due livelli adiacenti (n e $n - 1$) avvengono mediante un'**interfaccia**.

2 Architetture di rete

Un insieme di protocolli, uno per livello, è detto **pila di protocolli**. Un insieme di livelli, protocolli e interfacce è chiamato **architettura di rete**. Le architetture di rete sono alla base di tutte le reti di calcolatori.

3 Modello ISO-OSI

In letteratura, esistono due modelli per descrivere l'organizzazione dei protocolli: ISO-OSI¹ e TCP/IP.

Il modello ISO-OSI è composto da sette livelli:

¹International Organization for Standardization – Open Systems Interconnection

7	Applicazione
6	Presentazione
5	Sessione
4	Trasporto
3	Rete
2	Data Link
1	Fisico

Questi livelli si ricavano mediante un processo di

1. *raggruppamento*: si raggruppano funzioni simili per logica e/o tecnologia, in modo da definire gruppi omogenei;
2. *stratificazione*: i gruppi così definiti si organizzano gerarchicamente in **strati** o **livelli**, in modo da identificare regole di interazione inter-strato (**interfacce**) quanto più semplici e univoche.

Ogni strato n si interfaccia con lo strato immediatamente superiore $n + 1$ e con quello immediatamente inferiore $n - 1$ (ove presenti).

Ciascuno strato n è costituito da una o più entità, dette **n -entità**, che svolgono le funzioni di tale strato. L'interazione tra due sistemi avviene mediante interlavoro di entità omogenee (cioè dello stesso livello, ovvero **entità pari**) sui due sistemi.

I sette livelli del modello ISO-OSI sono stati definiti in base ai seguenti principi guida:

- tramite l'organizzazione a livelli si definisce un grado di astrazione;
- i livelli devono corrispondere a funzioni definite;
- le funzioni devono considerare l'insieme degli standard internazionali;
- i confini tra i livelli devono minimizzare il flusso delle informazioni tra livello e livello;
- il numero di livelli deve essere ottimale (né troppi né pochi).

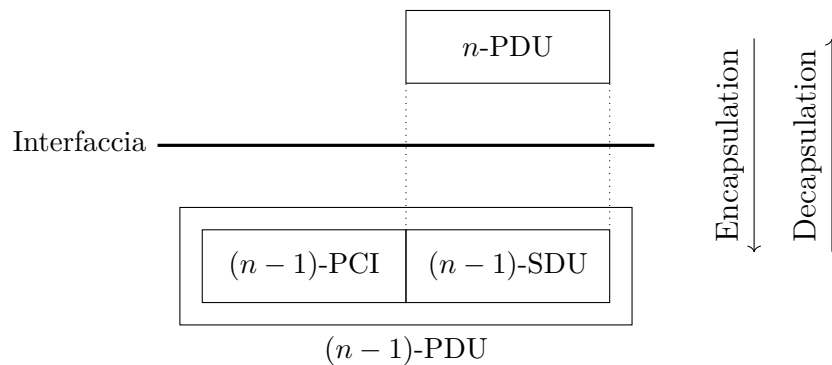
4 Interazione tra livelli adiacenti

Ogni livello usufruisce dei servizi resi dallo strato inferiore mediante opportuni **Service Access Point (SAP)**, che costituiscono appunto l'interfaccia tra i due livelli.

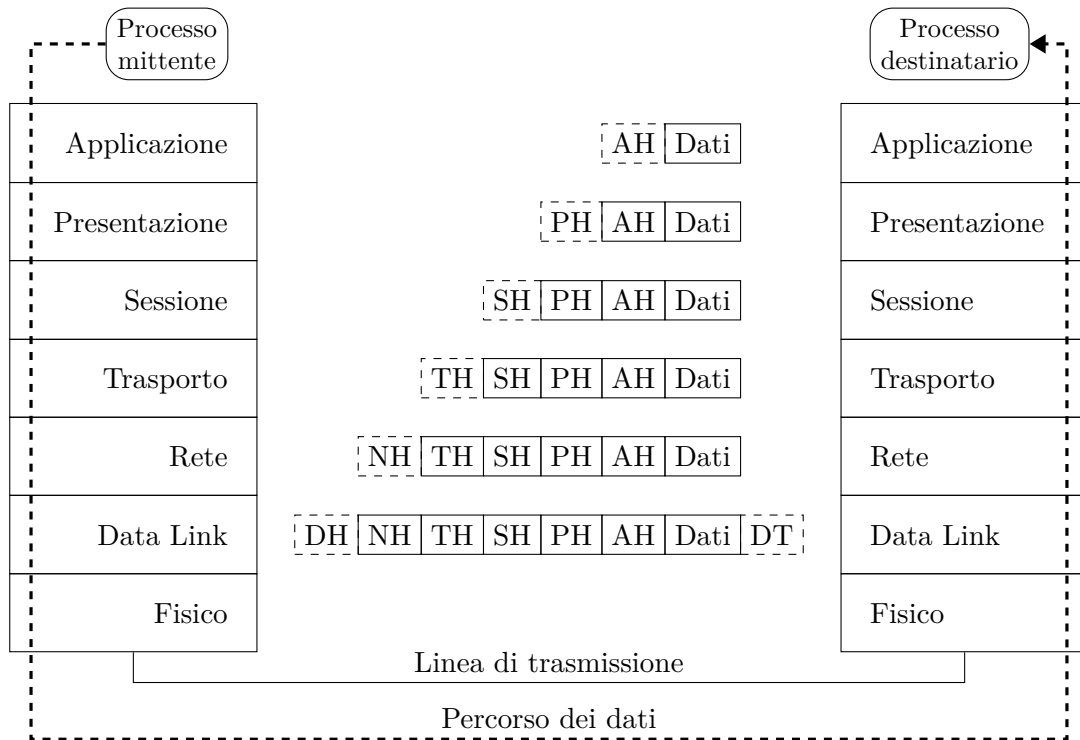
L'unità informativa che un'entità di livello n desidera inviare prende il nome di **Protocol Data Unit** di livello n , **n -PDU**. Essa viene passata al livello inferiore, $n - 1$, che costruisce a sua volta una **$(n - 1)$ -PDU**, composta da:

- **Service Data Unit, $(n - 1)$ -SDU**: l'intera PDU di livello n ;
- **Protocol Control Information, $(n - 1)$ -PCI**: le informazioni di controllo aggiuntive necessarie per lo svolgimento delle funzioni del livello $n - 1$. Queste informazioni sono tipicamente inserite sotto forma di **header** (intestazione) prima della SDU.

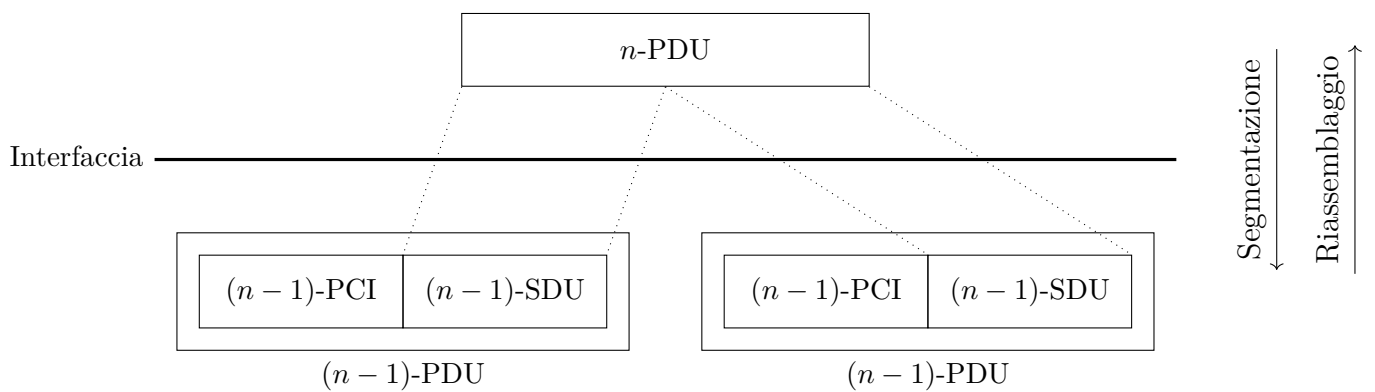
Questo processo prende il nome di **encapsulation**.



Sulla macchina ricevente, man mano che il messaggio ricevuto "risale" i livelli, si verifica il processo inverso, chiamato **decapsulation**: la $(n - 1)$ -PDU viene scomposta in $(n - 1)$ -PCI e $(n - 1)$ -SDU, la PCI viene usata dalle entità del livello $n - 1$ per svolgere le loro funzioni, e la SDU viene passata al livello superiore (nel quale essa prende il nome di n -PDU).



Nel passaggio dal livello n al livello $n - 1$, può anche essere eseguita una **segmentazione** della n -PDU, cioè questa può essere suddivisa in più $(n - 1)$ -SDU di dimensioni minori, ciascuna delle quali viene poi incapsulata in una $(n - 1)$ -PDU separata. In tal caso, le $(n - 1)$ -PCI conterranno le informazioni necessarie per permettere all'entità pari (di livello $n - 1$) sulla macchina ricevente di eseguire il **riassembaggio** dei dati ricevuti separatamente, ricostruendo così l'intera n -PDU.

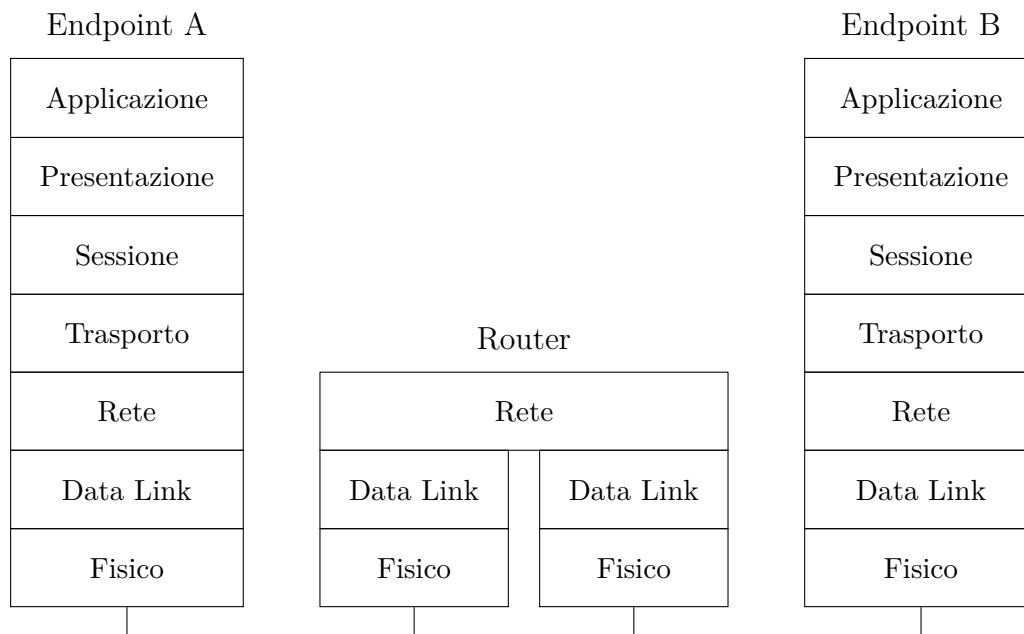


5 Sistemi di relaying

Spesso, due macchine che comunicano attraverso una rete non sono connesse direttamente, ma piuttosto tramite uno o più nodi intermedi. Ciascuno di questi nodi è un dispositivo di *relaying*, le cui caratteristiche e funzioni dipendono dal livello a cui opera:

- i **repeater** (ripetitori) e gli **amplificatori** operano al livello 1;
- gli **switch**, gli **hub** e i **bridge** operano al livello 2;
- i **router** operano al livello 3;
- i **gateway** operano a livelli superiori (tipicamente al livello 4).

Un dispositivo di relaying che opera a un livello n implementa solo i livelli dal fisico (1) all' n , poiché ha bisogno di leggere la n -PCI (e non i dati dei livelli superiori) per svolgere le sue funzioni. Ad esempio, il seguente schema rappresenta due macchine connesse attraverso un router (dispositivo di livello 3):



6 Livello di applicazione

Il livello di **applicazione** (application) fornisce, mediante un processo applicativo, l'interfaccia utente per l'accesso a servizi informativi distribuiti, come ad esempio il World Wide Web, la posta elettronica, il trasferimento di file, la gestione di database distribuiti, ecc.

I servizi forniti da questo livello sono completamente legati alle applicazioni:

- quali dati trasmettere;
- quando trasmettere;
- dove / a chi trasmettere;
- significato dei dati (bit/byte) trasmessi.

7 Livello di presentazione

Il livello di **presentazione** (presentation) consente l'interlavoro tra applicazioni che rappresentano i dati utilizzando formati diversi. Le sue funzioni sono quindi le operazioni di trasformazione, formattazione e modifica della sintassi dei dati, come ad esempio:

- compressione dei dati;
- traduzione dei codici utilizzati per rappresentare i dati;
- transcodifica a scopi di sicurezza (cifatura).

La sintassi con cui avviene lo scambio dei dati sulla rete può essere quella proprietaria di uno dei due sistemi interagenti, oppure una sintassi intermedia di trasferimento. Nel primo caso, solo uno dei due sistemi deve effettuare la traduzione, mentre nel secondo caso la traduzione è necessaria per entrambi.

8 Livello di sessione

Il livello di **sessione** (session) gestisce il dialogo e lo scambio di dati tra entità di presentazione. I principali servizi offerti da questo livello sono i seguenti:

- stabilisce le regole la gestione di una sessione di comunicazione (attivazione, trasmissione e rilascio);
- gestisce la modalità di dialogo (ad esempio, nel caso di una comunicazione half-duplex, tiene traccia di chi ha il turno di comunicazione, oppure permette l'interlavoro tra un terminale half-duplex e un'applicazione che opera in full-duplex, ecc.);
- recupera il dialogo in seguito a un'interruzione del servizio di trasporto;
- gestisce il controllo dei token;
- gestisce la sincronizzazione nel trasferimento dei dati (ad esempio mediante checkpoint).

9 Livello di trasporto

Il livello di **trasporto** (transport) è un livello molto importante, che svolge numerose funzioni. Esso è il primo (cioè il più basso) livello *end-to-end*, ovvero che gestisce la comunicazione tra gli endpoint (il mittente e i suoi uno o più destinatari). Le principali funzionalità che offre sono:

- **Multiplicazione e demultiplicazione:** trasporto dei flussi informativi di diverse applicazioni usando il singolo servizio offerto dal livello di rete (in altre parole, la possibilità eseguire su una stessa macchina più applicazioni che comunicano sulla rete).
- **Indirizzamento** delle unità informative alle diverse applicazioni su una stessa macchina.
- **Segmentazione e riassetaggio** delle unità informative.
- **Controllo di flusso end-to-end:** regolazione della velocità di trasmissione dei dati in funzione dello spazio disponibile nel buffer di ricezione del destinatario.
- **Controllo di congestione end-to-end:** regolazione della velocità di trasmissione dei dati in base allo stato del traffico sulla rete che collega il mittente al destinatario.
- **Controllo degli errori end-to-end.**
- Diffusione di messaggi a più destinazioni (**multicast**).

Questo livello può fornire sia un servizio connection-oriented, che

- garantisce l'assenza di errori e l'ordine di consegna dei messaggi,
- esegue il controllo di flusso e di congestione,

sia un servizio connectionless, senza garanzie.

10 Livello di rete

Il livello di **rete** (network) è un livello *point-to-point*, cioè gestisce la comunicazione tra i nodi (*point*) della rete (a differenza di un livello end-to-end, che, come già detto, si interessa solo degli endpoint della comunicazione). La sua principale funzione è l'**instradamento** delle informazioni dal mittente verso il destinatario. Inoltre, esso:

- nasconde al livello di trasporto la tecnica di commutazione utilizzata;
- implementa l'interfaccia necessaria alla comunicazione tra reti di tipo diverso (*internetworking*);
- esegue il controllo di flusso per prevenire la congestione della rete;

- esegue la moltiplicazione di più connessioni di rete su un unico collegamento dati;
- gestisce l'accounting dei pacchetti sulle reti a pagamento.

Questo livello opera tipicamente secondo un approccio **best effort**, cioè connectionless, senza garanzie. Perciò, i messaggi che esso invia/riceve prendono il nome di **datagrammi**.

11 Livello data-link

Il livello **data-link** realizza il trasferimento di dati privo di errori tra due nodi adiacenti lungo un collegamento trasmissivo. I suoi messaggi sono chiamati **frame**, e le sue principali funzionalità sono:

- il **controllo di accesso** a un collegamento condiviso;
- la rivelazione degli errori;
- il recupero di frame persi;
- il controllo di flusso (per sincronizzare, ad esempio, un mittente veloce con un ricevitore lento).

12 Livello fisico

Il livello **fisico** svolge tutte le funzioni necessarie a inviare e ricevere i bit sul canale fisico di trasmissione, gestendo aspetti elettrici/ottici (codifica dei bit, ecc.), meccanici (ad esempio gli standard dei connettori) e di comunicazione (ad esempio simplex, half-duplex o full-duplex).