

## 2

### IL MODELLO ISO/OSI

---

Alla fine degli anni '70, l'ISO sentì la necessità di proporre una serie di standard per le reti di calcolatori e avviò il Progetto OSI (Open System Interconnection), uno standard che propose un modello di riferimento per l'interconnessione di sistemi aperti. Il documento principale che illustra tale attività è il *Basic Reference Model* di OSI, standard ISO 7498 [1].

#### 2.1 IL MODELLO DI RIFERIMENTO OSI

Il modello di riferimento OSI ha due scopi:

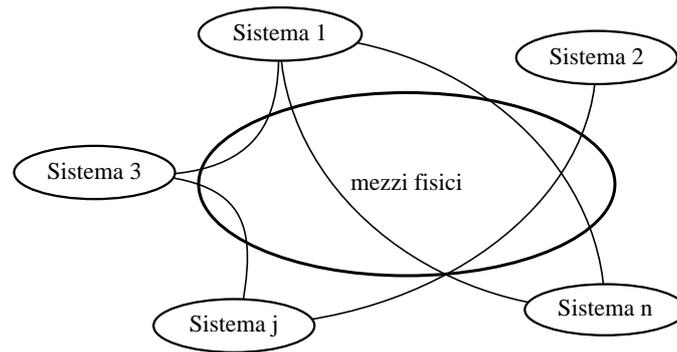
- fornire una base comune su cui sviluppare standard per l'interconnessione di sistemi informatici;
- fornire un modello rispetto a cui confrontare le architetture di rete proprietarie.

Il modello di riferimento OSI non ha come scopo la definizione di servizi o protocolli specifici. A questo sono stati delegati altri enti (es. IEEE, CCITT) o l'ISO stessa, in tempi successivi.

#### 2.2 SISTEMI, APPLICAZIONI E MEZZI TRASMISSIVI

OSI introduce il concetto di *sistema* (*system*) come un insieme di uno o più elaboratori con il relativo software, periferiche, terminali, operatori umani, processi, ecc. che complessivamente è in grado di elaborare dati. Nell'ambito di un sistema un'*applicazione* (*application*) è l'elemento che effettivamente svolge l'elaborazione dei dati.

Lo standard OSI tratta lo scambio di informazioni tra i sistemi e non come i sistemi sono realizzati o funzionano al loro interno. Tale trasferimento di informazioni avviene su *mezzi fisici* (*physical media*) secondo lo schema riportato in figura 2.1.

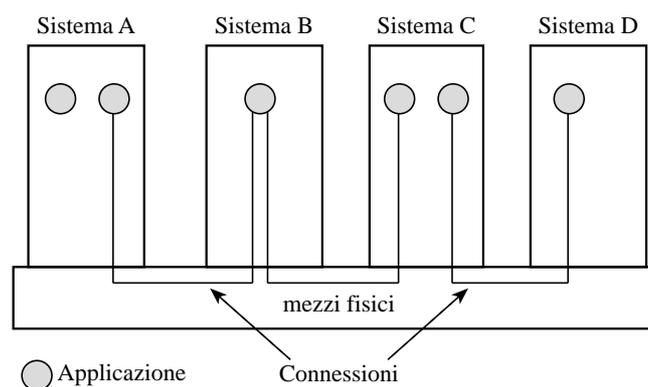


**Fig. 2.1** - Sistemi interconnessi da mezzi fisici.

L'architettura del modello di riferimento OSI è stata progettata pensando a tre componenti principali:

- il processo applicativo che deve scambiare le informazioni;
- la connessione che permette lo scambio delle informazioni;
- i sistemi.

Un esempio è illustrato in figura 2.2.



**Fig. 2.2** - Elementi Base di OSI.

### 2.3 ARCHITETTURA A LIVELLI

Per ridurre la complessità progettuale, OSI introduce *un'architettura a livelli (layered architecture)* i cui componenti principali sono:

- i livelli (*layers*);
- le entità (*entities*);
- i punti di accesso al servizio (SAP: *Service Access Points*);
- le connessioni (*connections*).

In una tale architettura, ciascun sistema è decomposto in un insieme ordinato di livelli, rappresentati per convenienza come una pila verticale. In figura 2.3 sono rappresentati i livelli che compongono il modello di riferimento ISO-OSI.



**Fig. 2.3** - Il modello ISO/OSI.

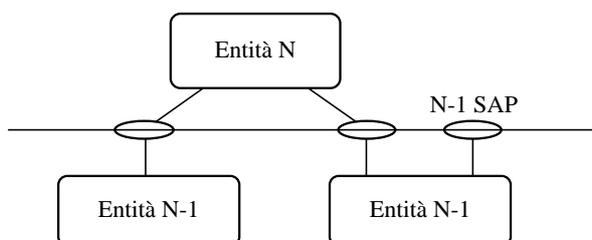
Livelli adiacenti comunicano tramite la loro *interfaccia (interface)*. Ogni livello è poi composto da una o più *entità*. Entità appartenenti allo stesso livello, su sistemi diversi, vengono dette *peer-entities*.

Tale approccio di progettazione a livelli è comune a tutte le moderne architetture di rete; ciò che varia dall'una all'altra è il numero dei livelli, il loro nome e le entità contenute.

Lo scopo di ciascun livello è quello di fornire servizi alle entità del livello superiore, mascherando il modo in cui questi sono implementati. Ad eccezione del livello più alto, un livello N fornisce servizi di livello N alle entità di livello N+1.

Le entità di livello N, eccetto il livello 1, per comunicare usano servizi di livello N-1. Le entità di livello 1 comunicano direttamente tramite i mezzi trasmissivi che le interconnettono.

Le entità usano e forniscono servizi tramite i SAP (Service Access Points), come illustrato in figura 2.4.

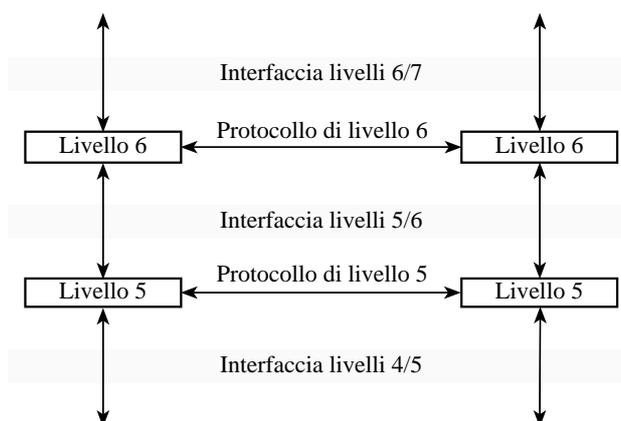


**Fig. 2.4** - Entità e SAP.

Le operazioni specifiche di un livello, cioè la cooperazione tra le entità appartenenti a quel livello, sono realizzate da un insieme di *protocolli* (*protocol*). Affinché due entità di livello N su sistemi diversi possano scambiarsi informazioni, una connessione deve essere stabilita nel livello N-1 usando un protocollo di livello N-1. Tale connessione di livello N-1 è stabilita tra due SAP di livello N-1.

## 2.4 PROTOCOLLI, LIVELLI E INTERFACCE

Riassumendo, livelli N comunicano attraverso un protocollo di livello N: ogni livello deve quindi mostrare un'interfaccia ben definita a quello immediatamente superiore, come viene mostrato in figura 2.5.



**Fig. 2.5** - Livelli, protocolli e interfacce.

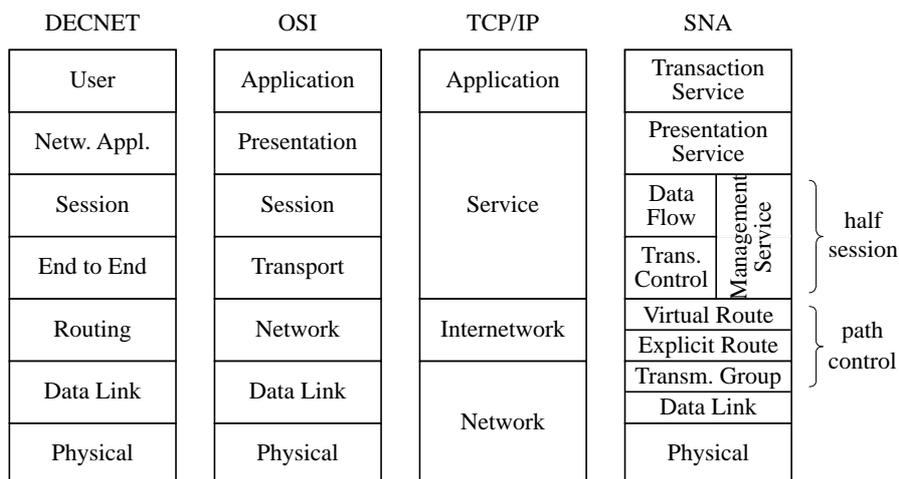
Anche se è definito un protocollo di livello N, nessun dato è trasferito direttamente da un livello N all'altro; infatti ogni livello passa dati e informazioni di controllo a quello sottostante, sino a quando si giunge al livello Fisico, che effettua la trasmissione. L'interfaccia definisce quali operazioni primitive e quali servizi sono forniti da un livello ai livelli superiori.

## 2.5 PRINCIPALI ARCHITETTURE DI RETE

L'insieme dei livelli, dei protocolli e delle interfacce definisce un'architettura di rete. Le architetture di rete più note sono:

- SNA (*System Network Architecture*), architettura della rete IBM;
- DNA (*Digital Network Architecture*), meglio nota come DECnet, la rete della Digital Eq. Corp.;
- *Internet Protocol Suite*, meglio nota con il nome TCP/IP, è la rete degli elaboratori UNIX e rappresenta uno standard "de facto" attualmente impiegato per la rete Internet di estensione mondiale;
- OSI (*Open System Architecture*), che è lo standard "de iure" in via di completamento nell'ambito dell'ISO.

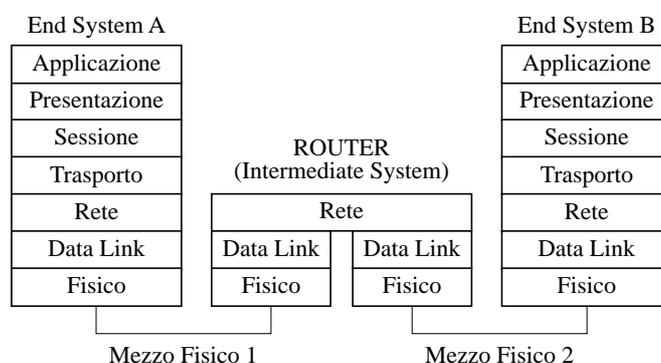
La figura 2.6 riporta un'analisi comparata tra di esse.



**Fig. 2.6** - Principali architetture di rete.

## 2.6 SISTEMI INTERMEDI

Non sempre lo scambio di informazione avviene direttamente tra i due sistemi finali che contengono le applicazioni (*ES: End Systems*). Può anche implicare l'attraversamento di sistemi intermedi (*IS: Intermediate Systems*). In essi esistono delle entità che assumono la funzionalità di *relaying*, cioè di inoltratrici di informazione. Tali entità possono essere collocate a vari livelli del modello OSI e gli IS assumono nomi diversi in funzione del livello a cui avviene il relaying: repeater (livello 1), bridge (livello 2), router (livello 3) e gateway (livello 7). In figura 2.7 è riportato un esempio di utilizzo di router come IS.



**Fig. 2.7** - Esempio di relaying tramite router.

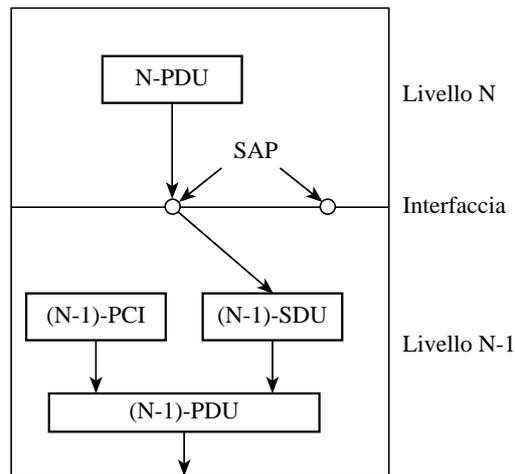
## 2.7 PROTOCOL DATA UNIT

Ogni livello  $N$  aggiunge ai dati ricevuti dal livello superiore alcune informazioni di controllo del protocollo  $N$ , dette comunemente "busta di livello  $N$ ".

Il tutto rappresenta i dati che verranno passati al livello inferiore che opererà in modo analogo. I dati generati da un protocollo di livello  $N$  sono detti  *$N$ -PDU (Protocol Data Unit)*. Essi diventano, una volta attraversata l'interfaccia tra il livello  $N$  e il livello  $N-1$ , una  *$(N-1)$ -SDU (Service Data Unit)*, come evidenziato in figura 2.8. La PDU di livello  $N-1$  viene quindi costruita preponendo alla  $(N-1)$ -SDU una  *$(N-1)$ -PCI (Protocol Control Information)*. Scopo della PCI è quello di contenere le informazioni di controllo del protocollo.

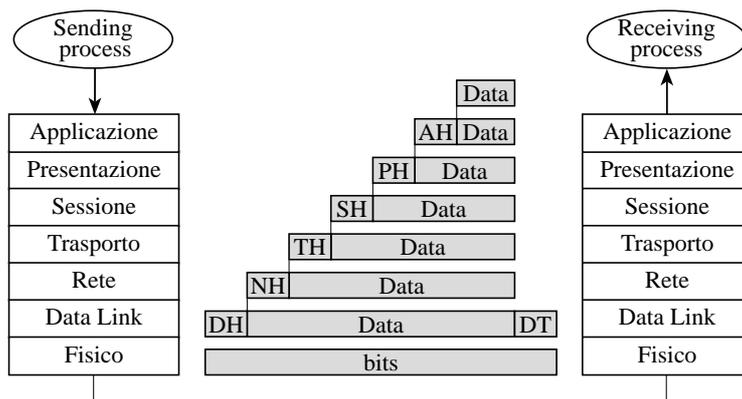
Molto spesso al termine PDU vengono sostituiti quelli meno precisi, ma di uso comune, di pacchetto o trama. Nell'ambito di un pacchetto il PCI rappresenta l'header del pacchetto stesso, già definito busta.

In figura 2.8 è stata fatta l'ipotesi semplificativa che il protocollo non frammenti i dati prima di trasmetterli.



**Fig. 2.8** - Relazione tra livelli.

La trasmissione dei dati avviene quindi attraverso una serie di passaggi da livelli superiori a livelli inferiori in un primo sistema, quindi attraverso mezzi fisici di comunicazione, e poi attraverso un'altra serie di passaggi, questa volta da livelli inferiori a livelli superiori, in un secondo sistema (figura 2.9).



**Fig. 2.9** - Imbustamento multiplo.

Si noti che viene aggiunto un header e in un caso un trailer per ogni livello attraversato.

## 2.8 IL LIVELLO 7: APPLICAZIONE

Il livello 7 è il livello *Applicazione*, cioè dei programmi applicativi (facenti parte del sistema operativo o scritti dagli utenti) attraverso i quali l'utente finale utilizza la rete; esempi di tali applicativi sono: VT (*Terminale Virtuale*), cioè connessione interattiva ad un elaboratore remoto, FTAM (*File Transfer and Access Management*), X.400 (la posta elettronica) e X.500 (*Directory Service*).

## 2.9 IL LIVELLO 6: PRESENTAZIONE

Il livello 6 è il livello *Presentazione*, che gestisce la sintassi dell'informazione da trasferire (ad esempio codifica ASCII o EBCDIC); a questo livello sono previste tre diverse sintassi: astratta (definizione formale dei dati che gli applicativi si scambiano, come in ISO 8824 o in ASN.1), concreta locale (come i dati sono rappresentati localmente) e di trasferimento (come i dati sono codificati durante il trasferimento).

## 2.10 IL LIVELLO 5: SESSIONE

Il livello 5 è il livello *Sessione*, responsabile dell'organizzazione del dialogo tra due programmi applicativi e del conseguente scambio di dati; esso consente di aggiungere a connessioni end-to-end (cioè tra due entità collocate in ES) servizi più avanzati, quali la gestione del dialogo (mono o bidirezionale), la gestione del token (per effettuare mutua esclusione nell'utilizzo di una risorsa condivisa) o la sincronizzazione (inserendo dei checkpoint in modo da ridurre la quantità di dati da ritrasmettere in caso di gravi malfunzionamenti).

## 2.11 IL LIVELLO 4: TRASPORTO

Il livello 4 è il livello *Trasporto*, e fornisce trasferimento trasparente di informazione tra entità del livello sessione. In particolare, si occupa di fornire un trasferimento dati affidabile e di ottimizzare l'uso delle risorse di rete. Compiti del livello 4 saranno quindi tipicamente la frammentazione, la correzione degli errori e la prevenzione della congestione della rete. Il livello 4 è il più basso livello a trascurare la topologia della rete e la presenza di sistemi intermedi (IS) e quindi è il primo livello detto *end-to-end*.

## 2.12 IL LIVELLO 3: NETWORK

Il livello 3 è il livello *Network*, che gestisce l'instradamento dei messaggi; esso determina se e quali sistemi intermedi devono essere attraversati dal messaggio per giungere a destinazione, quindi deve gestire delle tabelle di instradamento e provvedere ad instradamenti alternativi in caso di guasti (*fault tolerance*).

## 2.13 IL LIVELLO 2: DATA LINK

Il livello 2 è il livello *Data Link*, che ha come scopo la trasmissione sufficientemente affidabile di trame (*frame*); accetta come input dei pacchetti di livello 3 (tipicamente poche centinaia di bit) e li trasmette sequenzialmente. Esso verifica la presenza di errori aggiungendo delle FCS (*Frame Control Sequence*) e può gestire meccanismi di correzione di tali errori tramite ritrasmissione.

## 2.14 IL LIVELLO 1: FISICO

Il livello 1 del modello OSI è il livello *Fisico*, che si occupa di trasmettere sequenze binarie sul canale di comunicazione; a questo livello si specificano, ad esempio, le tensioni che rappresentano 0 e 1 e le caratteristiche dei cavi e dei connettori.

## 2.15 NOMI E INDIRIZZI

Il modello di riferimento OSI discrimina tra il *nome (title)* di una entità e la sua collocazione all'interno di un sistema, cioè il suo *indirizzo (address)* dato dalla concatenazione di SAP necessari per raggiungere attraverso i vari livelli tale entità. Questo ha il vantaggio di consentire l'accesso ad una entità anche se questa viene spostata da un sistema ad un altro. Tale distinzione impone l'esistenza di un *directory* (spesso detto anche *name-server*) per tradurre nomi in indirizzi e viceversa.

In particolare l'indirizzo di un applicativo sarà dato come la seguente concatenazione di una serie di SAP:

$$\text{Indirizzo Applicativo} = \text{PSAP} + \text{SSAP} + \text{TSAP} + \text{NSAP}$$

dove PSAP è il Presentation SAP, ecc. Un sistema è identificato in rete da un nome cui corrisponde un NSAP. L'indirizzo di livello 2 (Data Link) è ricavabile tramite opportuni algoritmi a partire da NSAP e viceversa. Molto spesso, in pratica, PSAP e SSAP coincidono e spesso si usa la notazione P/SSAP.

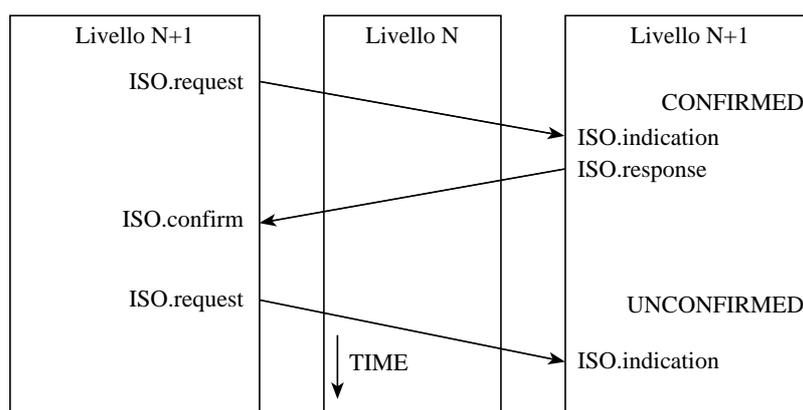
Infine occorre osservare che, nel caso che un applicativo abbia più di una connessione attiva contemporaneamente ad altri applicativi (caso tipico dei server), non è compito del suo indirizzo distinguere tra le varie connessioni, ma tale ruolo viene svolto da un *identificatore di connessione* (*connection endpoint identifier o instance number*). Per esempio un server FTAM può avere più connessioni attive tutte identificate dallo stesso indirizzo, ma con identificatori di connessione diversi.

## 2.16 LE PRIMITIVE OSI

Il modello di riferimento OSI standardizza la modalità di utilizzo dei servizi offerti da un dato livello N ad un livello N+1. Lo standard prevede quattro primitive di servizio: *Request*, *Indication*, *Response* (non è usata a livello 2) e *Confirm*; il diagramma temporale della figura 2.10 ne illustra l'uso.

Le modalità operative previste sono due:

- servizi con conferma (*confirmed*): in tali servizi il livello N+1 ricevente conferma l'avvenuta ricezione;
- servizi senza conferma (*unconfirmed*): in tali servizi il livello N+1 ricevente non conferma l'avvenuta ricezione.



**Fig. 2.10** - Primitive OSI.

## 2.17 PROTOCOLLI CONNESSI E NON

Per tutti i livelli superiori al livello fisico sono definite due modalità operative: una modalità *connessa* (*CONS: Connection Oriented Network Service*) e una modalità non connessa (*CLNS: ConnectionLess Network Service*). Un dato livello può fornire al livello superiore servizi di tipo connesso, non-connesso o entrambi. Questa è una scelta progettuale che varia per ogni livello, da architettura ad architettura. Lo standard originale ISO 7498 prevedeva solo la modalità connessa ma, vista l'importanza della modalità non connessa, è stata aggiunta in seguito come emendamento allo standard stesso (ISO 7498/Addendum 1).

In un servizio non connesso la spedizione di un pacchetto è simile alla spedizione di una lettera ordinaria con il sistema postale. Tutto avviene in una sola fase lasciando cadere la lettera nella buca delle lettere. La lettera deve contenere sulla busta l'indirizzo completo del destinatario. Non vi è alcun riscontro diretto che la lettera giunga a destinazione correttamente.

In un servizio connesso lo scambio di dati tramite pacchetti ricorda le frasi scambiate tra due interlocutori al telefono. Vi sono tre momenti principali:

- creazione della connessione (il comporre il numero telefonico e il "pronto" alla risposta);
- trasferimento dei dati (la conversazione telefonica);
- chiusura della connessione (i saluti finali e il posare il microtelefono).

### 2.17.1 La modalità connessa

Nella modalità connessa lo scambio di dati avviene tramite le tre fasi viste prima. Durante la fase di creazione della connessione (*initial setup*) due peer-entities concordano che trasferiranno delle PDU. Solo durante tale fase devono essere specificati gli indirizzi completi del mittente e del destinatario: successivamente le entità coinvolte specificheranno soltanto l'identificativo della connessione stabilito durante la prima fase. Un servizio connesso fornisce una modalità di trasferimento delle PDU affidabile e sequenziale. Per tutta la durata della connessione le PDU inviate sono ricevute correttamente nello stesso ordine. Se qualcosa non funziona correttamente, la connessione può essere riavviata (*reset*) o terminata (*released*). Per verificare che tutte le PDU inviate giungano a destinazione correttamente un servizio connesso utilizza degli schemi di numerazione dei pacchetti e di verifica dell'avvenuta corretta ricezione (ACK: acknowledgement). Quindi un protocollo connesso è in generale in grado non solo di rilevare la presenza di errori, ma anche di correggerli tramite ritrasmissioni.

### 2.17.2 La modalità non connessa

Con una modalità non connessa la comunicazione ha luogo in una fase singola: il pacchetto è inviato e deve contenere l'indirizzo completo del destinatario. Non essendo i pacchetti organizzati in una connessione, un pacchetto non può fare riferimento ad altri pacchetti trasmessi precedentemente o in seguito. Quindi un protocollo non connesso può solo rilevare la presenza di errori (scartando quindi le PDU errate), ma non correggerli in quanto non si possono realizzare meccanismi di ritrasmissione (in un pacchetto non è possibile fare riferimento ad altri pacchetti).

Un protocollo non connesso è in generale più efficiente di un protocollo connesso, specialmente se bisogna trasferire piccole quantità di dati: in quest'ultimo caso infatti l'overhead della creazione e distruzione della connessione è rilevante.

Un protocollo non connesso (detto anche *datagram*), non potendo garantire l'affidabilità del trasferimento dati, necessita che almeno un protocollo di livello superiore sia di tipo connesso.

Un'analisi comparata tra i due tipi di protocollo è fornita in tabella 2.1.

Caratteristica	Connection-Oriented	Connectionless
Initial setup	Richiesto	Impossibile
Indirizzo di destinazione	Durante il setup	In ogni pacchetto
Ordine dei pacchetti	Garantito	Non garantito
Controllo degli errori	Si	No
Controllo di flusso	Si	No
Negoziazione di opzioni	Si	No
Identificatore di connessione	Si	No

**Tab. 2.1** - Analisi comparata.

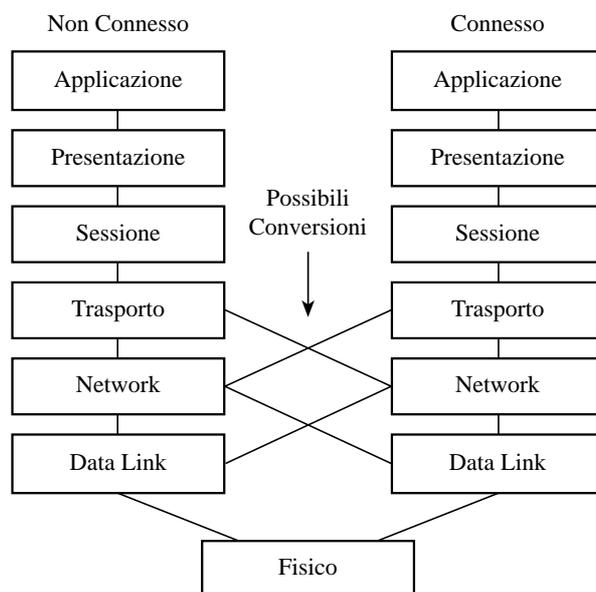
### 2.17.3 Applicazioni connesse e non

Anche le applicazioni possono operare in modo connesso oppure no. Un'applicazione infatti può non essere interessata a sapere se i propri dati sono giunti a destinazione o può implementare suoi schemi proprietari di controllo end-to-end. Si pensi ad esempio ad una applicazione che fornisca l'ora esatta a tutti i calcolatori della rete: essa può addirittura ignorare quali siano i calcolatori a cui fornisce il servizio e quindi operare in modo connectionless. Le applicazioni connesse sono molte, ad

esempio quelle di trasferimento file e di posta elettronica.

Chiaramente sia applicazioni connesse che non connesse devono poter operare su di una rete che ai livelli inferiori ha protocolli di tipo connesso oppure no.

Le possibili combinazioni tra livelli connessi e non nel modello OSI sono riportate in figura 2.11.



**Fig. 2.11** - Possibili conversioni tra modalità connessa e non.

Il livello 1 non può essere considerato né connesso né non connesso; i livelli 2, 3 e 4 possono operare in entrambe le modalità e in tutte le possibili combinazioni. I livelli 5, 6 e 7 devono comportarsi globalmente in modo connesso o non connesso.

A livello 2 l'operatività è normalmente di tipo non connesso quando si opera su reti locali, dove il mezzo trasmissivo è intrinsecamente affidabile, mentre è di tipo connesso su reti geografiche che sono intrinsecamente caratterizzate da un più alto tasso di errore.

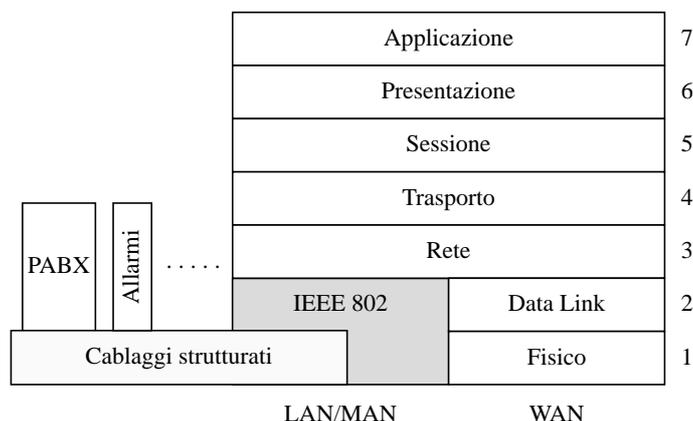
A livello 3, vi è sempre stata controversia tra gli informatici che vogliono un livello 3 non connesso e i telecomunicazionisti che lo vogliono connesso. Occorre anche evidenziare come negli ultimi anni il livello 3 sia sempre più stato considerato di competenza degli informatici e quindi si utilizzino sempre più protocolli non connessi.

Il livello 4 è in pratica sempre connesso: infatti, anche quando opera su un livello 3 connesso, l'affidabilità che tale livello 3 fornisce non è ritenuta soddisfacente.

## 2.18 RELAZIONE TRA OSI E ALTRI STANDARD

L'ISO, dopo aver definito il modello di riferimento OSI, si è posta l'obiettivo di sviluppare altri standard da collocare ai vari livelli. Parte di questi standard sono stati sviluppati direttamente dall'ISO stessa, altri sono stati demandati ad organizzazioni prestigiose quali IEEE, ANSI, CCITT, ecc.

In particolare, agli standard di livello 1 e 2 per le reti geografiche (WAN) ha collaborato il CCITT, mentre per la standardizzazione delle reti locali e metropolitane (LAN/MAN) è stato creato dall'IEEE un apposito progetto, detto progetto IEEE 802 e descritto nel capitolo 5. La relazione tra detti progetti è illustrata in figura 2.12.



**Fig. 2.12** - Relazioni tra standard.

Il progetto IEEE 802 per il livello fisico proponeva mezzi trasmissivi "proprieta-ri", non compatibili con quelli di altri cablaggi per il trasferimento di altri tipi di informazione (es.: allarmi, video, telefonia). Negli ultimi anni c'è stata una forte spinta ad unificare tali cablaggi e sono nati gli standard EIA/TIA 568 e ISO/IEC 11801 che hanno lo scopo di proporre un cablaggio standard unificato che possa essere la base su cui trasmettere sia dati tramite LAN, sia voce tramite PABX (centralini telefonici privati), sia altri tipi di informazione necessari a realizzare un "edificio intelligente". Nel capitolo 3 verranno illustrati i mezzi trasmissivi oggi utilizzati, mentre nel capitolo 4 verranno affrontate le problematiche del cablaggio strutturato degli edifici.

## 2.19 I PRINCIPALI PROTOCOLLI OSI

La figura 2.13 riporta le sigle dei principali protocolli adottati ai vari livelli del modello di riferimento ISO/OSI.

7	MHS CCITT X.400 ISO 10021	Directory CCITT X.500 ISO 9594	VT ISO 9040 (Service) ISO 9041 (Protocol)	FTAM ISO 8571
6	CCITT X.226 - ISO 8822 (Service) ISO 8823 (Protocol-Connection mode) ISO 9576 (Protocol-Connectionless mode)		ASN.1 CCITT X.208 - ISO 8824 (Language) CCITT X.209 - ISO 8825 (Encoding)	
5	CCITT X.225 - ISO 8327 (Service) ISO 8327 (Protocol-Connection mode) ISO 9548 (Protocol-Connectionless mode)			
4	CCITT X.224 - ISO 8072 (Service) ISO 8073 (Protocol-Connection mode) ISO 8802 (Protocol-Connectionless mode)			
3	Internetwork Protocol ISO 8473 (Connectionless)		Packet Level Protocol CCITT X.25 (Connection mode) ISO 8208	
2	Logical Link Control (LLC) ISO 8802.2		CCITT X.25 LAPB ISO 7776	CCITT LAPD Q.921
1	CSMA/CD ISO 8802.3	Token Bus ISO 8802.4	Token Ring ISO 8802.5	FDDI ISO 9314
	Reti Locali e Metropolitane			Reti Geografiche

**Fig. 2.13** - Principali protocolli OSI.

## BIBLIOGRAFIA

- [1] ISO 7498, "Data Processing, Open System Interconnection, Basic Reference Model".
- [2] Fred Halsal, "Data Communications, Computer Networks and OSI", Addison-Wesley, 1988.
- [3] H. Zimmerman, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnections," IEEE Trans. on Communications, Vol. COM-28, No. 4, April 1980, pp. 425-432.