

## **Il modello OSI e il modello DARPA.**

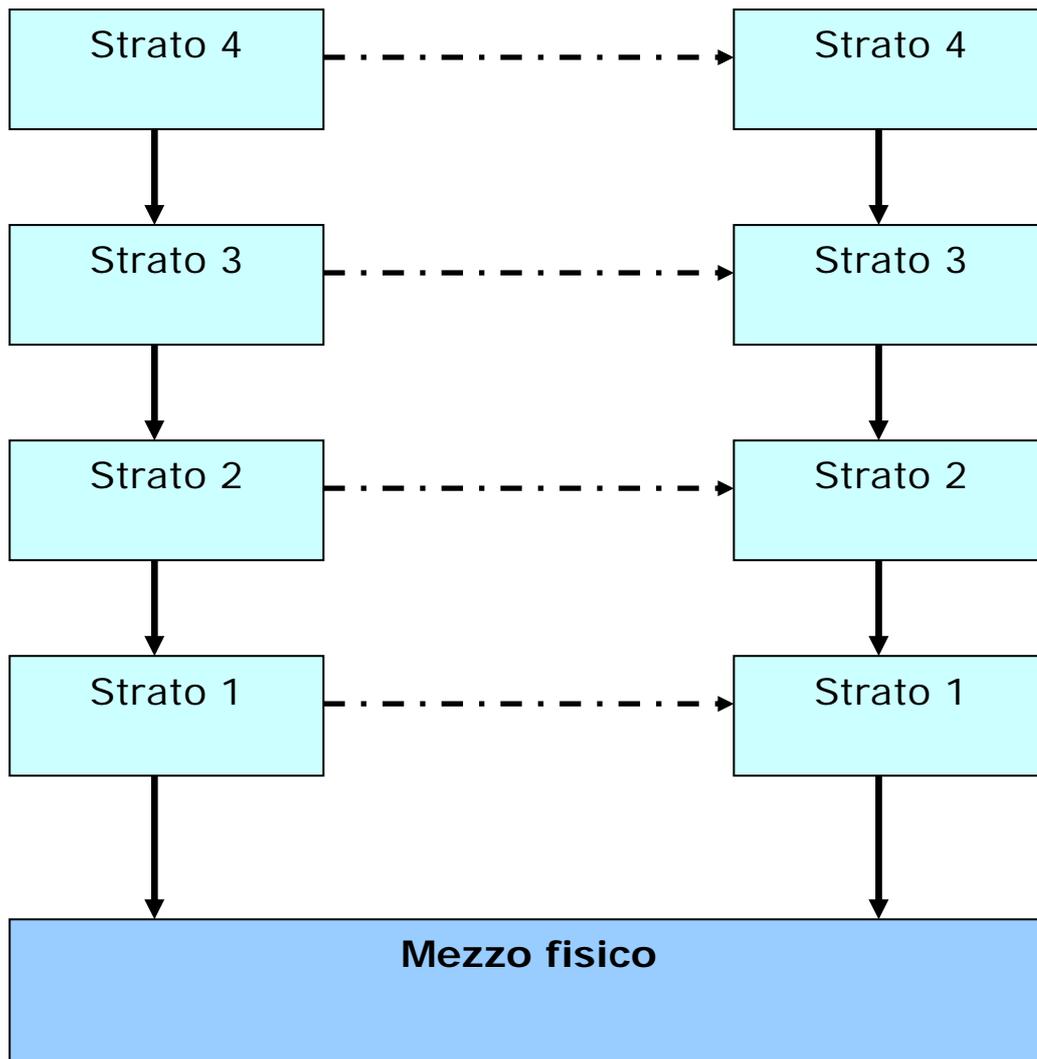
### ***Il modello OSI***

La comunicazione tra due utenti di una rete di computer implica la risoluzione di molteplici problemi e la realizzazione di una gran quantità di servizi, da parte d'opportuni software ed hardware della rete. Si deve garantire, ad esempio, che il singolo bit giunga dalla fonte al destinatario e il suo livello logico sia correttamente interpretato. Si deve suddividere opportunamente il flusso di bit in pacchetti in modo da contrassegnare opportunamente l'inizio e la fine dei vari dati, e distinguere i bit appartenenti ai dati veri e propri dai bit di controllo. Si deve individuare il percorso più efficiente fra due nodi della rete di computer, e così via. A tutto ciò corrisponde una notevole complessità nel progetto di una rete. Viceversa, la realizzazione di hardware e software che si occupino soltanto d'aspetti particolari della comunicazione fra computer in rete, deve essere facilitata da un'opportuna standardizzazione, in modo che il nuovo prodotto sviluppato s'integri in maniera facile con l'hardware e il software già esistente.

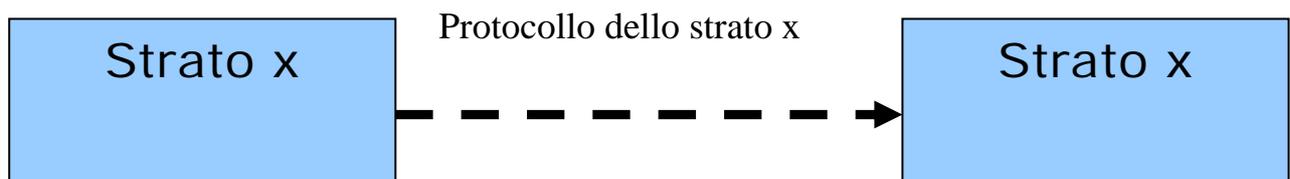
Ciò si ottiene mediante un'organizzazione gerarchica delle reti in più strati o livelli come nella figura seguente. Ogni strato ha lo scopo di fornire servizi allo strato superiore. Esso scherma la struttura degli strati inferiori per cui il software e l'hardware degli strati superiori non hanno alcuna necessità di avere informazioni sulla natura degli strati inferiori. Per fare un esempio, è lo stesso meccanismo per il quale un programma che gira su un computer non ha la necessità di conoscere la precisa struttura del computer stesso, ma, per gestire la tastiera e il video, fa riferimento ai servizi offerti dal sistema operativo.

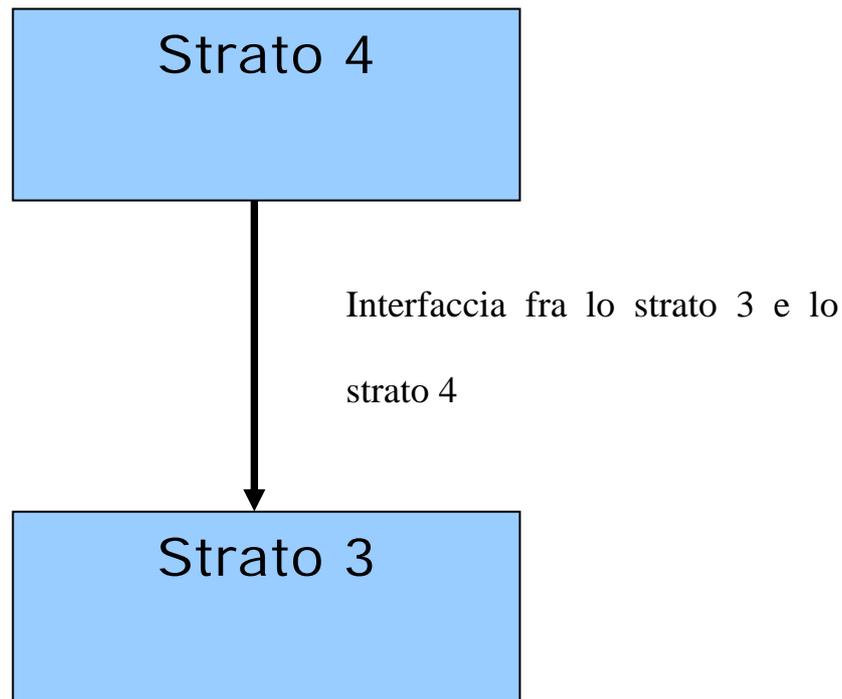
Lo strato  $x$  di una macchina comunica con lo strato omologo di un altro computer seguendo regole e convenzioni che consentono un dialogo corretto e che prendono il nome di protocolli. La comunicazione fra strati o livelli adiacenti avviene, invece, attraverso interfacce che definiscono le operazioni primitive e servizi che uno strato offre allo strato superiore.

Gli enti di standardizzazione definiscono i vari protocolli cui i sistemi devono allinearsi per poter comunicare. La realizzazione delle interfacce è invece lasciata libera ai costruttori, purché esse realizzino funzionalmente le primitive che definiscono il servizio.



In ogni strato è quindi attivo un protocollo. Le specifiche del protocollo di uno strato sono indipendenti da quelle degli altri strati. In altre parole, è possibile sostituire il protocollo di uno strato con un protocollo diverso senza dover cambiare alcunché nei protocolli degli strati adiacenti (naturalmente il servizio offerto allo strato superiore deve rimanere lo stesso).



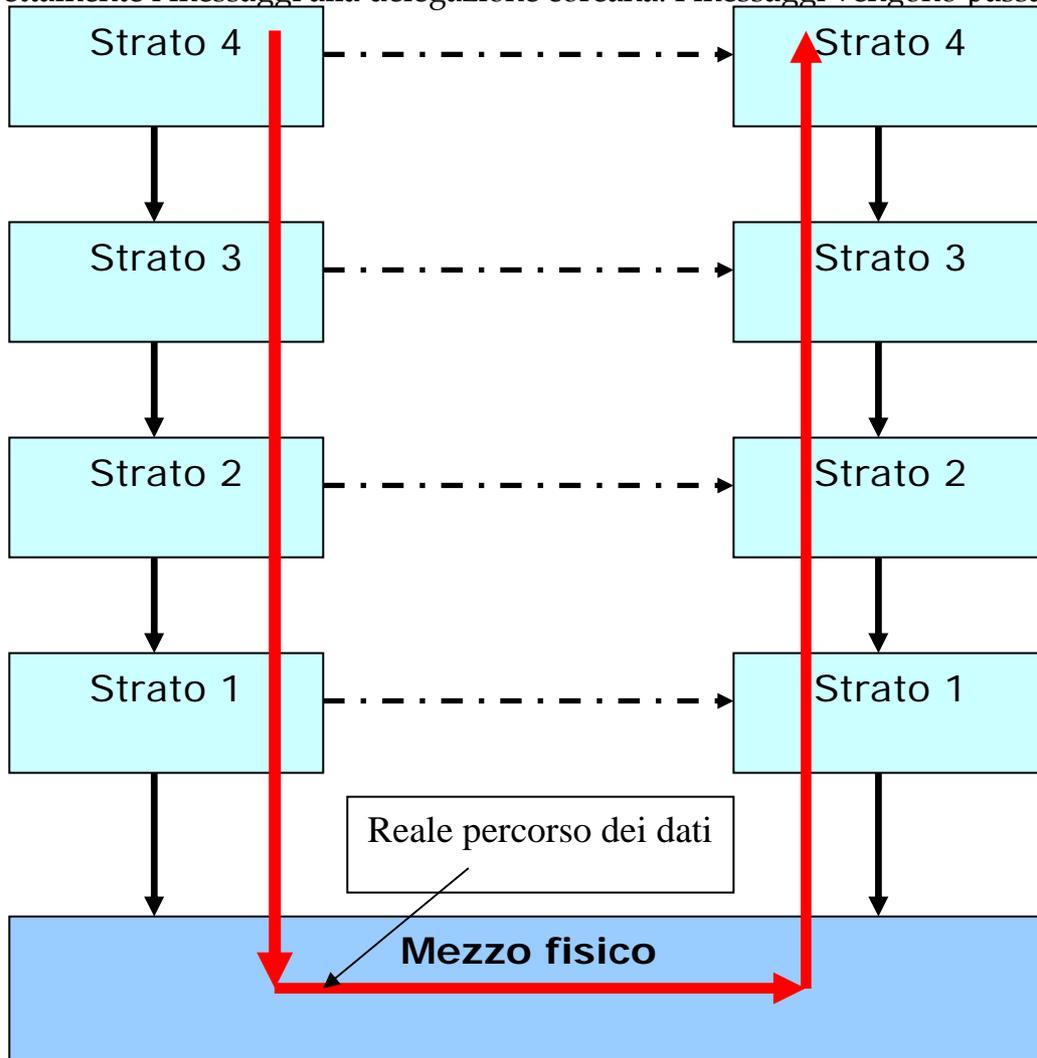


Ogni strato comunica con lo strato omologo dell'altro computer coinvolto nella comunicazione ma tale comunicazione è in realtà virtuale. In effetti, i dati non vengono inviati direttamente all'altro strato ma vanno allo strato inferiore e giù giù fino al mezzo fisico, dal quale passano all'altro computer e risalgono negli strati superiori.

Ogni strato infatti, poiché gestisce funzioni e servizi che svolgono attività simili, può essere modificato senza ripercussioni sugli altri strati; questo consente di suddividere operazioni molto complesse in elementi più semplici, di integrare i prodotti di venditori diversi, e quindi permette l'interoperabilità tra reti con caratteristiche differenti. Lo scambio delle informazioni tra i vari strati è invece delegata ad opportune interfacce di comunicazione.

Facciamo un esempio "non elettronico". Supponiamo che ci sia stata una guerra fra Cina e Corea. Ora ci sono le trattative di pace condotte da due delegazioni. Ogni delegazione risiede in un palazzo diverso. I componenti della delegazione cinese parlano soltanto cinese, così come i componenti della delegazione coreana parlano soltanto coreano. La delegazione cinese utilizza un interprete che parla cinese, inglese e francese. La delegazione coreana utilizza un interprete che conosce coreano, inglese e francese. La delegazione cinese "sta comunicando" con la delegazione coreana, inviandole messaggi

che seguono le regole dei protocolli della diplomazia, ma, in realtà, essa non invia direttamente i messaggi alla delegazione coreana. I messaggi vengono passati



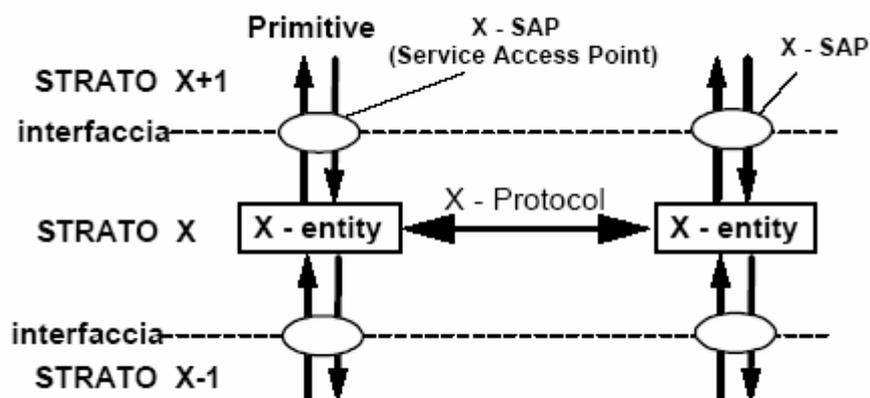
all'interprete attraverso un'interfaccia costituita da una segretaria. Lo strato "interprete" della delegazione cinese comunica con lo strato "interprete" della delegazione coreana usando un protocollo che implica che i messaggi scambiati siano in lingua inglese. Per comunicare, i due interpreti passano i messaggi ad uno strato dattilografo che si occupa di battere i messaggi a macchina. Una volta che il messaggio è pronto passa dallo strato "dattilografo" allo strato "collegamento fisico" che è costituito da un sistema di posta pneumatica. Come si vede dall'esempio, tutto il meccanismo è del tutto trasparente per le due delegazioni. Se, ad esempio, si sostituisce il sistema della posta pneumatica con un sistema di piccioni viaggiatori, le due delegazioni non se n'accorgono, né devono porsi il problema di come avviene l'implementazione della comunicazione.

Nel campo delle reti si utilizza uno standard a strati denominato modello OSI (Open System Interconnection) sviluppato dall'ISO (International Standard Organization).

Le prime specifiche pubblicate nel 1978 dall'ISO (International Standards Organization), fanno riferimento ai sistemi aperti, sistemi cioè che permettono la comunicazione tra componenti hardware e software di fornitori diversi. La versione completa, pubblicata nel 1984, è diventata di fatto uno standard internazionale, nel senso che tutte le case produttrici progettano i loro prodotti in base alle specifiche del modello OSI. Il modello ha avuto successo grazie all'architettura modulare impostata su sette strati completamente indipendenti. Questo modello prevede sette strati

1. Strato fisico
2. Strato collegamento dati o livello di linea o data link layer
3. Strato o livello di rete
4. Strato o livello di trasporto
5. Strato o livello di sessione
6. Strato o livello di presentazione
7. Strato o livello di applicazione

Il generico strato di livello X è costituito da gruppi funzionali chiamati entità (entity), la figura seguente illustra questo fatto.



In ciascuno dei sistemi si ha una entity per strato. La entity è la parte attiva di un sottosistema, in pratica possiamo identificare entity con sottosistema. A volte in un sottosistema si possono avere più entity per strato.

Una entity è caratterizzata dalla sigla del proprio strato, a livello 3 avremo per esempio delle Nentity (N per network o rete).

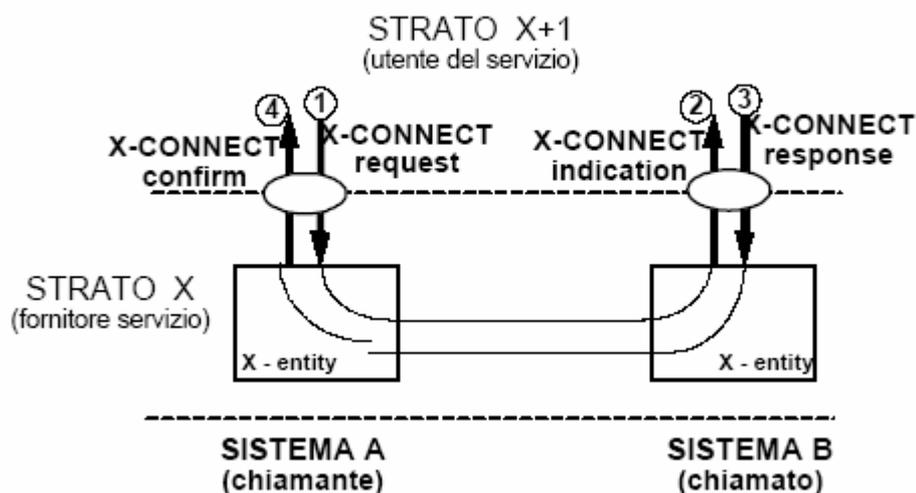
Uno strato (e quindi le sue entity, che dello strato sono le parti attive) fornisce servizio allo strato superiore attraverso un'interfaccia. A questa interfaccia le entity di livello superiore accedono mediante un Service Access Point (SAP). Anche i SAP, come le entity, sono caratterizzati dalla sigla del proprio strato; quindi lo strato 3 fornirà servizio attraverso un N-SAP, lo strato 4 fornirà servizio su un T-SAP e così via.

Un SAP è un indirizzo a cui uno strato invia una richiesta di servizio da parte di un altro strato. Differenti SAP identificano differenti entità di uno strato): Ad esempio a livello di applicazione differenti SAP identificano differenti servizi o applicazioni su un host (posta elettronica, ftp, http, ecc.).

Il dialogo fra due strati avviene, attraverso i SAP, secondo uno scambio di primitive. Ogni entity riceve a sua volta servizio dallo strato inferiore attraverso il SAP di livello inferiore (ad esempio un'entity di livello 4 fornisce servizio su un T-SAP e riceve servizio attraverso un N-SAP).

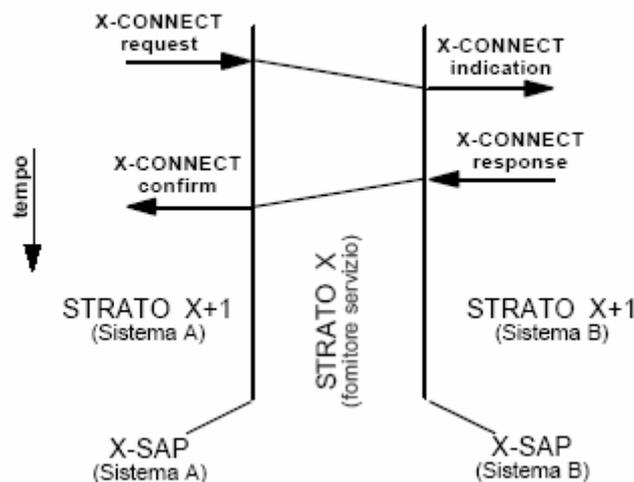
Vediamo ora qualche esempio di interazione fra strati, ossia di come un servizio può essere ottenuto mediante scambio di primitive. Supponiamo che il generico strato X fornisca un servizio di tipo "connection-oriented". Ciò significa che il suo utente (lo strato X+1) prima di poter usare il servizio deve richiedere una connessione.

Come sempre nei servizi "connection-oriented" c'è un utente chiamante e un utente chiamato. Si suppone che il chiamante sia il Sistema A. La Figura 3 illustra la sequenza di formazione della connessione.



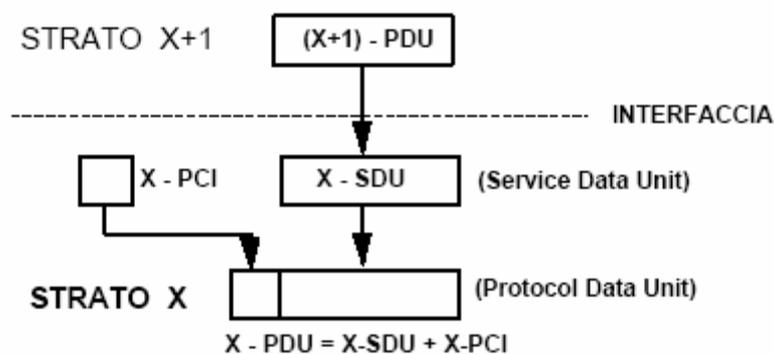
Per richiedere la connessione l'utente usa la primitiva "CONNECT". Di questa primitiva (che essendo usata per accedere al servizio di livello X si chiamerà X-CONNECT, secondo una convenzione che dovrebbe ormai risultare evidente) esistono quattro tipi, dipendenti dalla fase dell'interazione: "Request", "Indication", "Response", "Confirm". Nell'esempio della figura essi sono usati nella sequenza temporale seguente:

1. L'utente chiamante ( $X+1\text{-entity}$  di sistema A) emette, attraverso il proprio SAP, una
2. richiesta di connessione (CONNECT request);
3. Il servizio avvisa il chiamato mandandogli un'indicazione (CONNECT indication);
4. Il chiamato (sistema B) accetta la connessione mediante la CONNECT response;
5. Il servizio notifica al chiamante che la sua richiesta è stata accettata mediante la
6. CONNECT confirm.



Nel precedente diagramma la fascia centrale compresa fra le due rette verticali rappresenta il fornitore di servizio (ossia lo strato di cui si descrivono le funzioni), le zone a sinistra e a destra rappresentano gli utenti del servizio, ossia le *entity* di livello superiore. La primitiva CONNECT è una primitiva di segnalazione, serve a gestire la connessione fra gli utenti ma non provoca il trasferimento di informazione. Per il trasferimento di un messaggio si usa la primitiva "DATA"; di questa primitiva esistono solo i tipi "Request" e "Indication".

La Figura seguente mostra il flusso di dati all'interno di un sistema allorché un'entità dello strato generico X soddisfa una richiesta di servizio. Il servizio richiesto consiste nel trasferire all'altro sistema un brano di informazione (cioè un messaggio)



L'entità riceve attraverso il SAP l'informazione da trasferire, che per essa è una *Service Data Unit* (SDU). Ad esempio una N-entity riceverà una T-SDU. A questa SDU viene aggiunto un insieme di informazioni di controllo che costituiscono la *Protocol Control Information* (PCI) tipica del protocollo usato. Il messaggio così costruito costituisce l'unità di scambio del protocollo usato e viene detta *Protocol Data Unit* (PDU).

L'entità in esame, una volta formata la PDU, la trasmette alla *peer-entity* dell'altro sistema cioè all'entità di pari livello del ricevente. Per effettuare questa trasmissione chiede servizio allo strato inferiore passandole la PDU da trasmettere. Pertanto per lo strato inferiore la PDU diventa una SDU. In generale:

*“la PDU di un certo strato è una SDU per lo strato inferiore”.*

Il servizio fornito allo strato  $X+1$  consiste in una connessione fra due X-SAP. Per richiedere la connessione lo strato  $X+1$  emette una primitiva *CONNECT-Request*. Dopo che è stata formata la connessione è possibile trasferire dati su di essa. Si dice che lo strato X fornisce un servizio *“connection oriented”*.

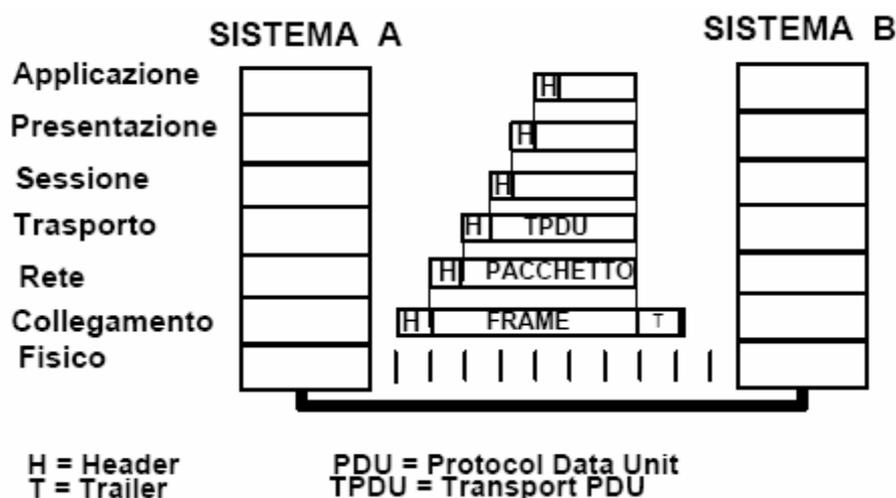
In alternativa lo strato X può accettare SDU di tipo datagramma. In tal caso non esiste connessione, non si usano primitive di tipo *CONNECT* e lo strato X fornisce un servizio *“connectionless”*.

Globalmente, partendo dal livello massimo (strato Applicativo), il trasferimento in uscita (ossia l'emissione) di un messaggio consiste nella:

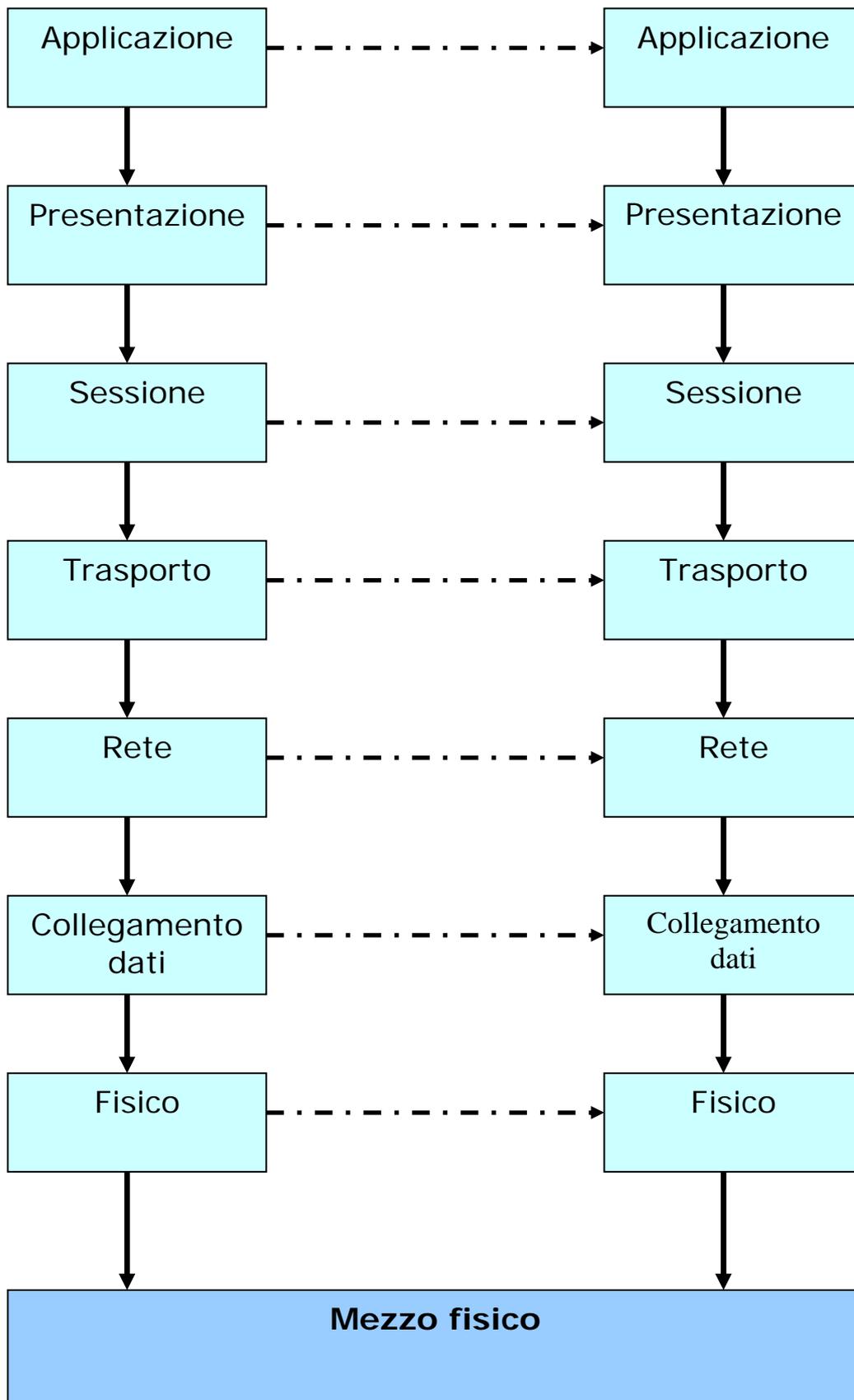
1. formazione della PDU;
2. consegna allo strato inferiore della PDU, che viene ricevuta come una SDU;
3. incapsulamento nel formato specifico del protocollo e formazione di una nuova PDU;
4. ripresa dal punto 2., se lo strato è superiore allo strato fisico.

Arrivati allo strato fisico, la PDU viene trasmessa come sequenza di bit sul mezzo trasmissivo.

Ricevuta dall'altro sistema subisce tutti i passaggi inversi fino a che il messaggio iniziale è ricevuto dall'applicazione remota. La Figura seguente mostra lo schema generale di incapsulamento.



Nell'incapsulamento si fa precedere un *header* ai dati, meno che nello strato 2 in cui ai dati si fa anche seguire un coda (*trailer*). Le PDU di alcuni strati hanno dei nomi specifici, entrati nell'uso comune. In particolare la PDU di livello 2 viene chiamata *frame*, quella di livello 3 *pacchetto*.



## Lo strato fisico

Lo strato fisico si occupa *della gestione del mezzo trasmissivo* (cavo coassiale, cavi STP o UTP, fibre ottiche) su cui avviene lo scambio di informazioni, della trasmissione dei singoli bit lungo la linea di trasmissione. A questo livello ci si occupa di tutte le problematiche che consentono la corretta trasmissione dei bit dal ricevente al destinatario. Si stabilisce, ad esempio, il livello di tensione che rappresenta il valore logico 1 e quello che rappresenta il valore logico 0; la durata del singolo bit, la sincronizzazione di trasmettitore e ricevitore, l'organizzazione delle interfacce fisiche fra computer e linea, ecc.

## Il livello di linea.

Si occupa di stabilire una corretta comunicazione fra due computer di una rete. Organizza i dati in pacchetti o frame individuati da bit o caratteri di inizio e fine, di **ordinare e incapsulare i dati** in una struttura logica detta **trama o frame** e di trasmetterla correttamente al sottostante livello Fisico. Implementa sistemi di rilevamento di errori nella trasmissione dei pacchetti, si occupa di stabilire una corretta connessione fra i due computer, stabilisce delle regole affinché un computer più veloce non sommerga di dati un computer più lento, introduce dei metodi per garantire che i pacchetti trasmessi siano stati ricevuti, ecc. Il livello 2 dunque gestisce la **sincronizzazione tra mittente e destinatario** attraverso l'introduzione di caratteri di sincronismo e *modera lo scambio dei dati* tra terminali di potenza diversa. Tra i protocolli più conosciuti di questo livello vi sono HDLC e BSC.

## Livello di rete.

il livello 3 si occupa *della traduzione del nome logico dell'host destinatario in indirizzo fisico*. Si occupa fondamentalmente di stabilire il miglior percorso possibile fra due computer di una rete. Questo percorso, attraverso altri nodi della rete, può essere definito staticamente e rimanere fisso per tutta la durata della sessione di lavoro, oppure può variare dinamicamente in funzione delle condizioni di utilizzo della rete. Si occupa di gestire la comunicazione controllando il flusso dei dati al fine di evitare la congestione della rete. I dati ricevuti dal livello superiore sono organizzati in **pacchetti o datagrammi**, che contengono il *network header* con l'*indirizzo logico* della sorgente e della destinazione.

Provvede inoltre alla *commutazione del messaggio* o dei singoli pacchetti che compongono il messaggio, determinando se il cammino deve essere lo stesso per tutti i pacchetti, oppure se ogni pacchetto può seguire un cammino diverso.

## **Livello di trasporto**

Il livello di trasporto si occupa sostanzialmente di nascondere il livello di rete all'utente. Occorre tener presente che, nel caso di reti estese geograficamente, il livello di rete è di proprietà di società di telecomunicazioni per cui gli utenti della rete non possono intervenire sulle prestazioni dei primi strati. Il livello di trasporto si deve occupare di garantire un collegamento di qualità fra due utenti rendendolo indipendente da eventuali difetti del livello di rete. Maggiore sarà la qualità offerta dal livello di rete, minori saranno i compiti del livello di trasporto e viceversa. In genere si ha una connessione di trasporto per ogni connessione di rete. Se, però, il traffico fra i due utenti è elevato, il livello di trasporto può creare più connessioni di rete fra le quali suddividere il traffico fra i due utenti. Se, viceversa il livello di traffico fra le connessioni di trasporto è limitato, il livello di trasporto può multiplexare più connessioni di trasporto su un'unica connessione di rete. La funzione principale è quindi quella di garantire che lo *scambio di informazioni sia affidabile e sicuro e che i pacchetti siano ricevuti in sequenza e senza duplicati*. Per questo motivo **controlla il flusso delle informazioni, invia i segmenti nella sequenza corretta ed effettua il controllo e il recupero dei pacchetti attraverso l'acknowledgment e la ritrasmissione**. Il livello di trasporto stabilisce, mantiene e chiude il circuito virtuale; per questo motivo è detto **connection-oriented**.

## **Lo strato di sessione**

Lo strato di sessione offre ulteriori servizi alla comunicazione fra due utenti. Un servizio fondamentale è, ad esempio, la sincronizzazione. Se si stanno trasferendo dei dati fra due computer e si ha un'interruzione della connessione, si dovrebbe ricominciare il trasferimento dei dati dall'inizio. Per evitare questo, il livello di sessione introduce dei punti di controllo nella serie di dati da trasmettere, in modiche, in caso di guasto, la trasmissione riprenda dall'ultimo punto di controllo. Poiché i dati vengono trasmessi su una rete a commutazione di pacchetto, questo livello ha il compito fondamentale di

definire la durata della trasmissione e il **controllo del dialogo** decidendo quale delle due stazioni deve trasmettere. Specifica cioè se la trasmissione deve essere *alternata* tra i due dispositivi in comunicazione, o se entrambi possono inviare informazioni *simultaneamente*. La trasmissione simultanea può provocare una collisione a livello di, quando uno o più messaggi oltrepassano un messaggio inviato precedentemente, causando confusione su uno o su entrambi i sistemi in comunicazione. Nella *trasmissione simultanea*, il livello di sessione chiama altri livelli per gestire la conversazione. Se le collisioni compromettono troppo la comunicazione, allora il *controllo del dialogo* può optare per la comunicazione alternata; in tal caso, attraverso un **data token**, permette ad ogni host di comunicare quando arriva il suo turno. Il livello di sessione realizza inoltre una *sincronizzazione* ad alto livello, introducendo dei *checkpoint nel flusso dei dati*. Ad ogni *checkpoint* entrambi gli host effettuano il backup dei dati, salvano la configurazione della rete e del clock, quindi annotano il punto in cui è arrivata la conversazione; in tal modo, se durante il trasferimento dei dati la sessione si interrompe, la sessione successiva può sincronizzare la comunicazione riprendendo la trasmissione dal punto in cui è stata interrotta.

### **Lo strato di presentazione.**

Questo è il primo strato che si occupa del “significato” dei dati. Con stringhe di bit si possono rappresentare dati della natura più diversa: testi, numeri in virgola fissa, numeri in virgola mobile, ecc, che possono essere rappresentati con codifiche diverse. Lo strato di presentazione si occupa di rendere compatibili le rappresentazioni dei dati negli scambi fra computer che utilizzano codifiche diverse. Grazie a questo livello, *le applicazioni che girano su computer con sistemi diversi possono comunicare tra loro in modo indipendente dalle applicazioni stesse*. Al livello 6 vengono infatti definiti gli standard ASCII ed EBCDIC per gestire i file di testo, gli standard GIF (Graphic Interchange Format), JPEG (Joint Photographic Experts Group) e TIFF (Tagged Image File Format) per rappresentare le immagini, lo standard MPEG (Motion Picture Experts Group) per la codifica e la compressione dei dati su memorie di massa digitali, lo standard MIDI (Musical Instrument Digital Interface) per l'audio digitale, lo standard Quick Time per i file con

audio e video. Altri aspetti di cui si può occupare il livello di sessione sono la compressione dei dati, e la crittografia.

## **Lo strato di applicazione**

Lo strato di applicazione è l'interfaccia verso l'utente finale e gli offre una serie di servizi di alto livello come il trasferimento dei file, la posta elettronica, ecc. Il *livello 7* occupa lo strato più alto della pila e per questo motivo non fornisce servizi agli altri livelli, ma *interagisce in modo diretto con le applicazioni usate dall'utente, fornendo i servizi di rete*. Al fine di soddisfare i servizi richiesti dall'utente, esso verifica innanzitutto se vi sono sufficienti risorse per comunicare tra i sistemi. E' fornito di due interfacce, una diretta e l'altra indiretta. Attraverso **l'interfaccia diretta** con il livello sottostante, *l'Application Layer serve le applicazioni tipiche di una rete*, quali lo scambio di e-mail, il trasferimento dei file, l'accesso ai database, l'accesso ai siti Web, la gestione remota di applicazioni distribuite e l'emulazione di terminali. Ognuna delle applicazioni suddette utilizza un protocollo specifico dell'Application Layer.

*Un'interfaccia indiretta e un redirector vengono invece forniti per le applicazioni standalone (word processor, fogli di calcolo, programmi per la gestione di database ecc.) in un ambiente LAN, per attivare la funzionalità client-server. Il redirector è un protocollo che deve essere attivato sul cliente di rete, che permette di condividere e mappare diversi tipi di risorse. Esso entra per esempio in funzione ogni volta che un utente chiede di salvare un file su un file server o di stamparlo su un print server di rete. In tal caso il redirector rende l'applicazione, con cui è stato creato il file, un client di rete, abilitando così l'archiviazione o la stampa del file richiesto. In generale il redirector permette all'amministratore di rete di impostare sul client locale un nome logico per le risorse remote e di conseguenza abilita un client a utilizzare le risorse di rete come se fossero locali.*

Nella **comunicazione tra i processi** il server mantiene generalmente la connessione con il client solo per il tempo necessario a effettuare la transazione, come il download di una pagina web o l'invio del file al print server o al file server. Terminata la transazione, la connessione viene abbattuta. In questa regola non rientrano tuttavia Telnet, né FTP

## **Il modello di riferimento DoD/DARPA**

All'origine di internet vi è la famiglia dei protocolli TCP/IP e di Internet. Tra le motivazioni che indussero il DoD (Dipartimento della Difesa) ad istituire l'agenzia DARPA (*Defense Advanced Research Project Agency*) e implementare lo *stack* di protocolli TCP/IP vi furono le seguenti:

- Necessità di definire una piattaforma di comunicazione standard per consentire l'interconnessione di computer di vari costruttori con sistemi operativi diversi.
- Condivisione delle risorse (applicazioni, informazioni) residenti su un'ampia varietà di computer (mainframe e minicomputer) presenti all'interno del DoD e dei vari centri di ricerca che cooperavano con esso.
- Favorire la competizione tra i vari fornitori attraverso l'introduzione di standard "de-facto", evitando la proliferazione di protocolli proprietari con conseguenti problemi di interfacciamento tra sistemi di costruttori diversi.

Il protocollo TCP/IP anticipa la standardizzazione OSI emanata da .

## **L'architettura e i livelli del modello DoD/DARPA**

L'architettura di comunicazione DoD/DARPA è stata ideata per consentire di "esprimere" e progettare nel modo più razionale e flessibile applicazioni che interagiscono tra di loro in un ambiente di rete distribuito (identificato spesso anche come Client/Server) e indipendente dalla tipologia e topologia di rete utilizzata per interconnettere i computer o *host*. Essa è basata su una struttura gerarchica all'interno della quale giocano un ruolo fondamentale tre particolari tipi di oggetti:

- Processi: formano la parte centrale di tutto il sistema in quanto rappresentano l'entità direttamente coinvolta nella comunicazione e con la quale avviene l'interazione da parte degli utenti secondo uno schema di tipo client/server. A questo livello di astrazione si può immaginare che ogni processo comunichi con un altro processo in una modalità indicata generalmente con il termine *end-to-end*. I processi operano sugli *host*.

- Host: rappresentano i sistemi sui quali sono “ospitati” (da cui il termine host) i processi. Uno o più processi possono trovare “ospitalità” su un host secondo le potenzialità di quest’ultimo. I processi operano sugli host sfruttando i “servizi di trasporto” messi a disposizione dalla infrastruttura di rete sottostante, allo scopo di interagire con altri processi che possono risiedere o sullo stesso host oppure su host remoti. È importante osservare che, pur essendo il tipo di trasporto una “emanazione” della rete sottostante, esso è pensato per nascondere il dettaglio relativo alla infrastruttura di comunicazione realmente utilizzata. Il suo ruolo fondamentale è di “far credere” ai processi di avere a disposizione una propria rete dedicata, della qualità e affidabilità richiesta. Per questo motivo, a questo livello di astrazione, la comunicazione può essere considerata di tipo *host-to-host* cioè diretta tra host.
- Reti: costituiscono il mezzo attraverso il quale gli host comunicano per scambiarsi le informazioni inerenti i processi che su di essi sono in funzione.

Dalla interazione tra i suddetti oggetti (processi, host, reti e con l’ausilio del concetto di trasporto prima introdotto) è possibile delineare un modello gerarchico attraverso il quale si possono definire le relazioni indicate in figura 1.

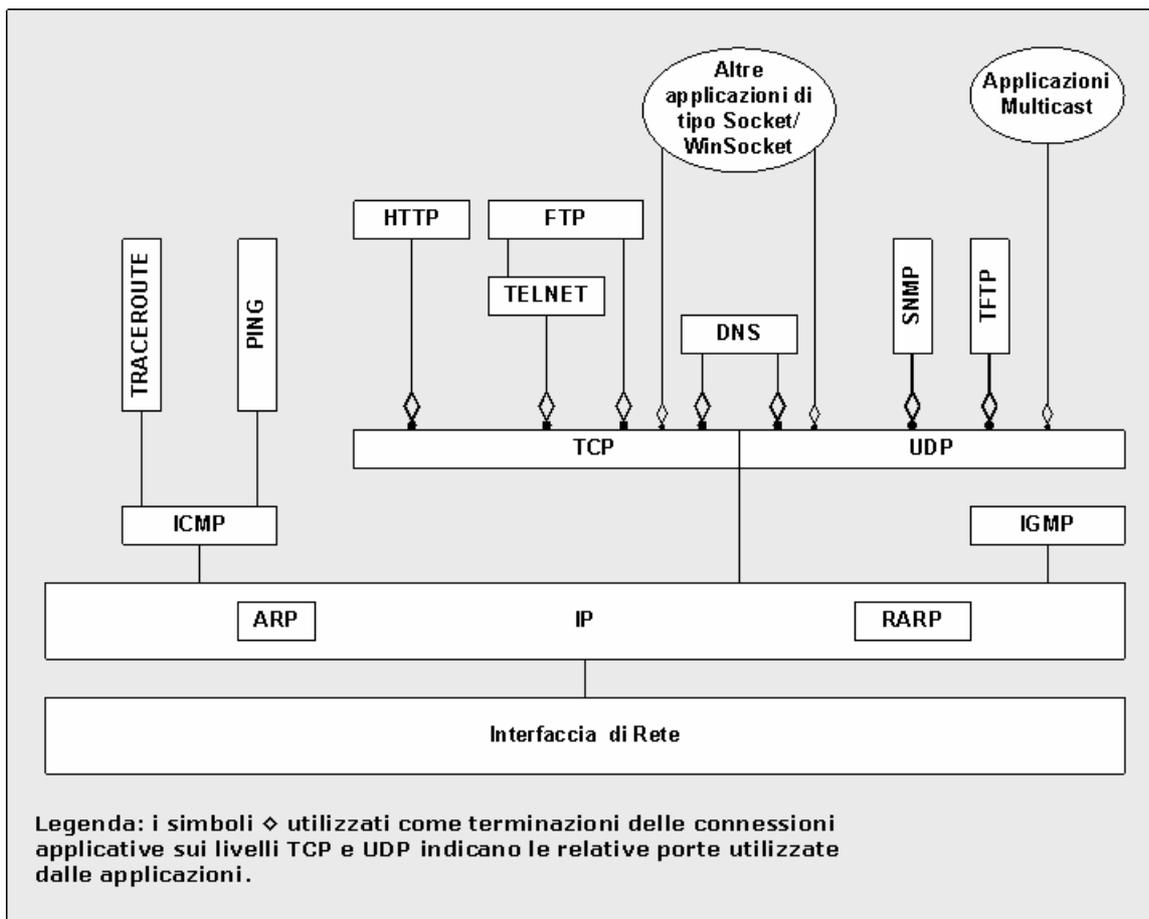
**Tabella 1: Struttura gerarchica del modello DoD/DARPA**

<b>Process/Application Layer</b>	<b>Livello Processo/Applicazione</b>
<b>Host-to-Host o Transport Layer</b>	<b>Host-Host o Livello Trasporto</b>
<b>Internet o IP Layer</b>	<b>Livello Internet o IP</b>
<b>Network Access o Network Interface Layer</b>	<b>Livello Interfaccia di Rete</b>

Esso costituisce il modello DoD/DARPA indicato anche come *stack* o pila TCP/IP, per enfatizzare la relativa struttura gerarchica.

## Protocolli di riferimento

Nella figura è rappresentata la relazione tra alcuni protocolli della famiglia TCP/IP.



Da notare che la standardizzazione DoD/DARPA si riferisce solamente ai tre strati più alti. Il livello Interfaccia di Rete risulta essere indefinito nel senso che può essere utilizzata qualsiasi tipo di rete.

Per quanto riguarda il livello Internet il protocollo principale di riferimento è l'*Internet Protocol* o semplicemente IP. Il compito del protocollo IP è di mettere a disposizione del livello trasporto i servizi di inoltro datagrammi, indirizzamento logico, frammentazione e diagnostica. I servizi forniti dal protocollo IP ai protocolli dei livelli superiori sono di tipo *connection-less* (cioè non orientati alla connessione; infatti l'unità di informazione trattata a questo livello è chiamata appunto datagramma proprio per ricordare che il tipo di

spedizione dei dati non presuppone l'apertura preventiva di una "sessione" da tenere sempre attiva fino al termine della comunicazione, ma ciascuno viene inviato in una unica sessione UDP indipendente). Inoltre, non è compito di questo protocollo implementare meccanismi per aumentare il grado di affidabilità o per il controllo del flusso dei datagrammi inviati.

Esistono all'interno di questo strato altri "protocolli di servizio" non sempre visibili e "percepibili agli effetti esterni" ma altrettanto importanti per le funzionalità di tutto lo stack TCP/IP.

Tra questi quelli più importanti sono i seguenti:

- ARP (*Address Resolution Protocol*) e RARP (*Reverse ARP*): si occupa della risoluzione di un indirizzo IP nel corrispondente indirizzo fisico dell'interfaccia di rete utilizzata. Viceversa, il protocollo RARP consente di effettuare la risoluzione inversa, cioè dall'indirizzo fisico permette di risalire all'indirizzo IP.
- ICMP (*Internet Control Message Protocol*): permette di verificare lo stato di funzionamento di una rete attraverso l'invio di messaggi di "tracciamento" e di segnalare eventuali errori che si possono verificare durante la trasmissione dei datagrammi IP; problemi di routing con mancato raggiungimento di una determinata destinazione .
- IGMP (*Internet Group Message Protocol*): utilizzato per implementare le funzionalità multicast.

A livello di trasporto i protocolli utilizzati sono:

- TCP (*Transmission Control Protocol*): è il protocollo di trasporto che si occupa di offrire alle applicazioni un servizio di consegna dei messaggi affidabile di tipo orientato alla connessione (o *connection-oriented*).

- UDP (*User Datagram Protocol*): è il protocollo di trasporto che offre un servizio minimale di consegna dei messaggi di tipo non orientato alla connessione (o *connection-less*).

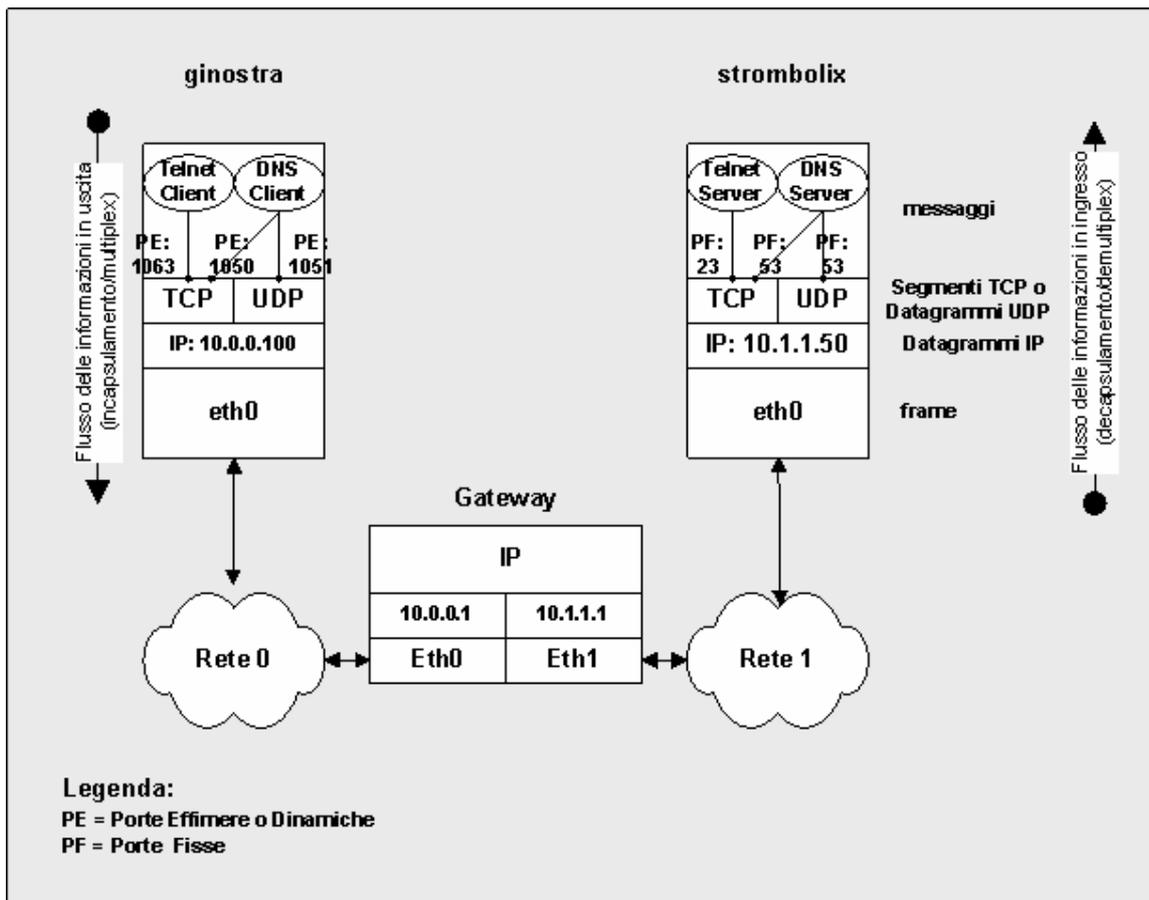
Infine, al livello *Process Layer* si collocano tutte le applicazioni sia native di ogni implementazione dello stack TCP/IP in qualsiasi ambiente operativo (Unix/Linux, Windows, Mac OS X, Novell NetWare, ecc.) che altre applicazioni sviluppate da terze parti.

Come indicato nelle specifiche DoD/DARPA, in alcuni casi particolari, la presenza di tutti i livelli non è richiesta esplicitamente dallo standard. Ad esempio nel caso dei router i livelli Trasporto e Processo non sono necessari. Inoltre è possibile implementare applicazioni che si integrano all'interno del modello DoD/DARPA a qualsiasi livello (Interfaccia di Rete, IP o Trasporto, sia TCP che UDP) a seconda delle funzionalità necessarie.

### **Comunicazione fra host all'interno di una infrastruttura TCP/IP**

Nel caso ancora più generale nel quale gli host sono residenti su reti diverse, queste ultime vengono interconnesse da particolari tipi di dispositivi chiamati *gateway* o *router*. Un gateway può essere costituito sia da un apparato di rete con hardware dedicato a svolgere tale ruolo, sia da un'applicazione software in funzione su un determinato computer (Unix/Linux, Windows NT/XP/2K/2K3, ecc.).

Uno scenario del genere può essere schematizzato come in figura 2.



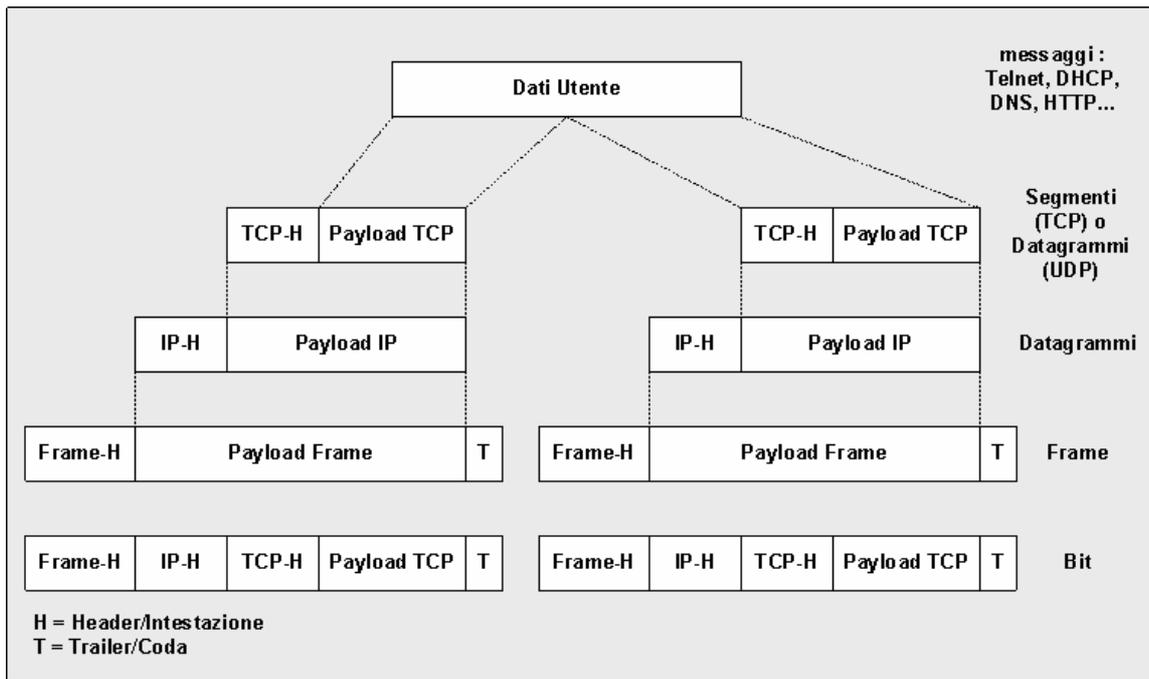
Ogni host caratterizzato dal possedere una sola interfaccia o scheda di accesso alla rete è univocamente identificato da un unico indirizzo IP. Viceversa nel caso dei gateway, dovendo questi interconnettere più reti, essi richiedono una scheda di rete per ciascuna rete da interconnettere. Naturalmente ogni scheda di rete deve possedere un ben determinato indirizzo IP compatibile con la rete sulla quale essa si “interfaccia” o “affaccia”.

Ad ogni livello (con eccezione dei livelli estremi) esiste un protocollo che offre servizi per i protocolli dei livelli adiacenti. In particolare, ogni protocollo di livello n utilizza i servizi messi a disposizione dal livello sottostante n-1, ne arricchisce le funzionalità ed offre al livello superiore n+1 un servizio (o protocollo) a valore aggiunto.

In particolare i dati inseriti da parte dell’utente (o originati autonomamente da un processo interno) vengono passati al processo applicativo con il quale avviene l’interazione uomo-macchina (Browser per navigare in Internet, client DHCP, DNS, FTP, Telnet, SMTP, ecc.). Quest’ultimo li predisporre secondo il formato dei messaggi specifico

del protocollo utilizzato (DHCP, DNS, FTP, Telnet, ecc.) e li consegna ad uno dei due protocolli disponibili a livello trasposto (TCP oppure UDP). A seconda del tipo di applicazione utilizzata, è possibile che quest'ultima utilizzi uno solo di questi ( HTTP, DHCP, FTP/TFTP, Telnet, SMTP, ecc.) oppure entrambi (DNS).

In ogni fase di transizione da un livello ad un altro le informazioni subiscono un processo di incapsulamento che consiste nell'inserire i dati provenienti dal livello superiore in una sorta di "busta virtuale", secondo le specifiche indicate dal protocollo del livello immediatamente inferiore. Questa struttura dati viene indicata come *Protocol Data Unit* (PDU). Ad ogni passaggio di livello vengono aggiunte delle informazioni di controllo specifiche di ciascun livello. Queste informazioni, chiamate *header* o intestazione, sono necessarie per poter compiere a destinazione la fase simmetrica di decapsulamento o "sbustamento" delle informazioni e, soprattutto, per identificare i protocolli e/o servizi per i quali ciascun livello sta operando. In particolare in corrispondenza dell'interfaccia tra il livello applicativo e quello di trasporto si utilizza il numero di *porta* per identificare univocamente l'"applicazione cliente" dello strato trasporto; in corrispondenza dell'interfaccia tra il livello di trasporto ed il livello IP si utilizza il campo *Protocol* del datagramma IP per identificare univocamente il protocollo di livello superiore o *Upper Layer Protocol* (ULP) che può essere TCP,, ICMP, IGMP, ecc., come specificato da IANA alla pagina <http://www.iana.org/assignments/protocol-numbers>. Naturalmente esse costituiscono un *overhead* cioè un sovrappiù rispetto ai dati veri e propri contenuti in una PDU ed indicati nella figura come *payload*.



Il livello di trasporto, esegue la frammentazione del messaggio ricevuto in segmenti di dimensioni più adatte (o frammenti) ad essere inviate in rete. Ad ogni segmento viene assegnato un numero d'ordine (identificato dal campo *Identification* della PDU IP) in modo da consentire all'host destinazione di ricostruire il messaggio originale. Per consentire di identificare l'applicazione (o processo) mittente e destinazione, in ogni segmento TCP o UDP vengono utilizzati due campi da 16 bit identificati come *Source Port* e *Destination Port*. In questo modo, le porte vengono utilizzate come indirizzi delle applicazioni coinvolte nella comunicazione ed attestate alle due estremità di una sessione TCP o UDP.

A questo punto il segmento viene passato al livello IP, il quale procede con l'incapsulamento dei segmenti nei datagrammi IP introducendo un nuovo header nel quale verranno specificati, tra le altre informazioni di controllo, i numeri del protocollo di livello superiore (campo *Protocol* della PDU IP) e gli indirizzi IP mittente e destinatario (o destinatari in caso di comunicazione multicast o broadcast).

### Processo di risoluzione del nome host nel corrispondente indirizzo IP

Nel caso in cui nella instaurazione della comunicazione tra mittente e destinatario si sia utilizzato il nome host, durante la transizione del pacchetto dal livello trasporto al livello

IP è necessario procedere con la risoluzione del nome nel corrispondente indirizzo IP assegnato alla interfaccia di rete utilizzata per la connessione (scheda di rete LAN, modem, scheda Wi-Fi, ecc.). La modalità di risoluzione del nome può essere eseguita in maniera statica o dinamicamente tramite l'utilizzo di un server DNS o WINS (*Windows Internet Name Service*) in una rete basata su sistemi operativi Microsoft che non fa uso del servizio DNS .

Successivamente il datagramma viene consegnato al livello Interfaccia di Rete il quale compone il frame inserendo le relative informazioni di controllo e gli indirizzi fisici (MAC Address).

### **Processo di risoluzione dell'indirizzo logico IP nel corrispondente indirizzo fisico (MAC Address) dell'interfaccia di rete**

Affinché il livello Interfaccia di Rete sia in grado di inviare il frame al destinatario è necessario che nel passaggio dal livello *Internet Protocol* (IP) al livello Interfaccia di Rete venga eseguita la risoluzione dall'indirizzo logico IP assegnato alla scheda di rete nel corrispondente indirizzo fisico o MAC Address. A tal proposito all'interno dello stack TCP/IP viene utilizzato di default il protocollo *Address Resolution Protocol* (ARP).

### **I concetti di porta (Well-Known Port, Ephemeral Port), socket e connessione o associazione end-to-end**

Ad ogni tipo di applicazione/servizio di sistema deve essere associato un numero ben determinato e conosciuto, identificato come *Well-Known Port* o Porta Fissa (PF). Le porte WN o PF devono essere comprese nell'intervallo  $0 \div 1023$ . Viceversa, lato client per i processi utente non di sistema che girano in modalità non privilegiata, ogni implementazione dello stack TCP/IP assegna per convenzione dei numeri di porta temporanei e validi solo per la durata della connessione. Queste porte vengono solitamente indicate come *Dynamic* o *Ephemeral* (i.e.: porta effimera o volatile) o *Local* o *Private Port*, ed il loro valore dovrebbe (sempre secondo IANA) essere compreso tra 49.152

e 65.535 (in quanto si utilizza una variabile intera senza segno a 16 bit (i.e.: due byte/ottetti) per la sua rappresentazione). Il resto delle porte comprese nell'intervallo 1024 ÷ 49.151, vengono indicate come *Registered Port*, in quanto possono essere utilizzate per registrare applicazioni sviluppate da terze parti. Alcune implementazioni di stack TCP/IP fanno un uso arbitrario di queste regole. Ad esempio, in ambiente Microsoft Windows, per le porte dinamiche o effimere vengono utilizzati numeri di porta anche nell'intervallo 1024 ÷ 5000.

Oltre alle informazioni specificate in fase di inserimento del comando precedente, ne esiste un'altra, anch'essa obbligatoria affinché si possa completare la connessione, ma che non viene direttamente indicata dall'utente in quanto è implicitamente definita a livello codice sorgente nel comando utilizzato lato client e dipende, naturalmente, dal tipo di applicazione/servizio al quale ci si vuole connettere (HTTP, Telnet, FTP, TFTP, SMTP, ecc.). Essa si riferisce al tipo di protocollo utilizzato a livello di trasporto. Nel caso Telnet il protocollo di trasporto è TCP.

La conoscenza di quest'altra informazione consente ad un processo di identificare esattamente il "cammino" per giungere al corrispondente processo remoto con il quale interagire.

Per raggiungere questo scopo viene utilizzata da ciascun processo la seguente terna di parametri:

<Indirizzo IP, Protocollo, Porta>

alla quale viene dato il nome di *socket*.

Pertanto, il concetto di porta precedentemente espresso non ha alcun valore se non associato ad un ben determinato protocollo di trasporto (TCP o UDP). Di conseguenza esisteranno 65.535 ( $2^{16} - 1$ ) porte TCP ed altrettante porte UDP.

Mettendo insieme le informazioni contenute nei due socket (lato client e lato server) alle estremità di una connessione, si ottiene la seguente quintupla di parametri:

<Protocollo, IP Mittente, Porta Mittente, IP Destinazione, Porta Destinazione>

la quale permette di identificare univocamente l'associazione o connessione tra i due processi in modalità *end-to-end*.

### Confronto tra il modello DoD/DARPA e il modello ISO/OSI

Nella figura 4 è rappresentata la relazione tra i sette livelli del modello di riferimento ISO/OSI e i quattro livelli del modello TCP/IP DoD/DARPA.



Figura 1: Confronto tra il modello di riferimento ISO/OSI ed il modello DoD/DARPA

Come già detto precedentemente il modello DoD/DARPA ha anticipato l'uscita della raccomandazione ISO/OSI.

Come si può notare dalla figura, il livello Processo dello stack TCP/IP implementa le funzionalità tipicamente espletate dai livelli che vanno dal settimo (Applicazione) fino al quinto (Sessione) del modello ISO/OSI. Lo strato Trasporto del modello DoD/DARPA implementa le funzionalità dell'omologo livello ISO/OSI. Il livello Internet si può equiparare al livello tre OSI. Infine il livello Interfaccia di Rete TCP/IP implementa le funzionalità dei livello uno e due del modello ISO/OSI.