

Architetture di comunicazione a strati

- Occorre distinguere due aspetti distinti nel processo di comunicazione dati attraverso una rete:
 - a. Vi è l'esigenza di trasmettere i dati in maniera *tempestiva, corretta ed efficiente* sotto il profilo dei costi;
 - b. I dati giunti a destinazione devono risultare riconoscibili sotto un profilo *sintattico e semantico*.
- Mentre il primo problema è molto spesso legato alla natura strettamente tecnologica dell'implementazione del sistema di telecomunicazione, il secondo punto trova soluzione nella cosiddetta teoria dell'informazione e della comunicazione (*Information and Communication Technology, ICT*).

➤ Alcuni esempi:

1. Rete telefonica

- ✓ Il numero e la lunghezza dei collegamenti devono essere minimizzati
- ✓ La “banda” (in Mbit/s) percepita dall’utente deve essere massimizzata in relazione al costo

➤ Alcuni esempi:

2. Applicativo che preveda un trasferimento di file tra un PC dotato di un sistema operativo Windows ed una workstation su cui gira il sistema operativo Unix

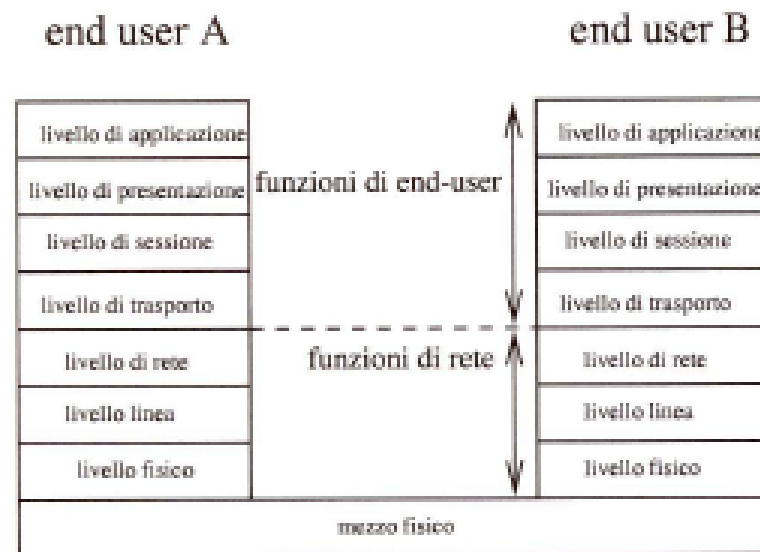
- ✓ Comandi e caratteri di controllo devono essere specificati con precisione nei due sistemi
- ✓ Definizione di regole di comunicazione “ad-hoc” su cui i sistemi interessati a quel particolare scambio di informazioni devono concordare prima che la comunicazione possa avere inizio

Strati bassi e strati alti

- Per affrontare –e risolvere– questi due distinti problemi insiti nella comunicazione dati via rete, sono nate le architetture di comunicazione a strati
- In queste architetture è sempre possibile distinguere tra
 - Strati bassi, *network -oriented*
 - Strati alti, *application - o user-oriented*

Modello di riferimento ISO-OSI

- Un modello di riferimento estremamente importante per le architetture di comunicazioni a strati è il modello ISO-OSI (*International Standards Organization – Open System Interconnection*).
- È un modello, un costrutto astratto schematizzato come segue



Modello di riferimento ISO-OSI

- I tre livelli bassi garantiscono la capacità di networking (per questo vengono anche chiamati *network-oriented*), mentre i quattro livelli alti sono deputati al processing richiesto per presentare i dati all'utente finale in formato appropriato e riconoscibile (per questo vengono anche chiamati *application- o user-oriented*). È questa distinzione logica che consente agli utenti di una rete che adotti tale paradigma progettuale di comunicare apertamente (*openly*).

Modello di riferimento ISO-OSI

- I principi alla base della definizione degli strati (o livelli) sono:
 - Un nuovo livello viene creato ogniqualvolta si rende necessario un diverso grado di astrazione
 - Ciascun livello deve svolgere una ben precisa funzione
 - Tale funzione deve essere individuata con l'obiettivo di pervenire ad una implementazione standardizzata a livello internazionale
 - I confini tra i livelli devono essere scelti così da minimizzare lo scambio di informazioni tra i livelli
 - Il numero di livelli deve risultare tale che
 - Nessun livello sia troppo complesso
 - L'integrazione tra i livelli non sia troppo onerosa
-
- L'architettura sia sufficientemente flessibile

Livello fisico – Physical layer

- Il compito del livello fisico è quello di trasmettere i bit sul canale di comunicazione. A dispetto della semplicità logica vi è una complessità tecnologica non riscontrabile negli altri strati.
- Esempi:
 - ❖ Trasmissione su cavo in rame
 - Quale supporto fisico impiegare (cavo coassiale thick, cavo coassiale thin, doppino binato non schermato, doppino binato schermato, ...)
 - Che tipo di segnale elettrico inviare sulla linea (forma d'onda del segnale elettrico, potenza del segnale elettrico, spettro del segnale, ...)

Livello fisico – Physical layer

- Tecniche per la sincronizzazione del segnale e la rivelazione dei dati impiegate nel ricevitore

❖ Trasmissione su fibra ottica

- Che tipo di fibre adottare (monomodali, multimodali, ...)
- Che tipo di sorgenti luminose utilizzare (laser, diodi laser, ...)
- Che tipo di fotorivelatori sfruttare (fotodiodi, ...)
- Con che tecnica codificare i dati (on/off, a modulazione di ritardo, ...)

❖ Trasmissione via radio

- Quale banda sfruttare

Livello fisico – Physical layer

- Quali antenne utilizzare
 - Che livello di potenza trasmettere
 - Quali tecniche di cifratura adottare per la sicurezza dei dati
- In sostanza occorre definire le caratteristiche meccaniche, elettriche , funzionali e procedurali dell'interfaccia hardware.

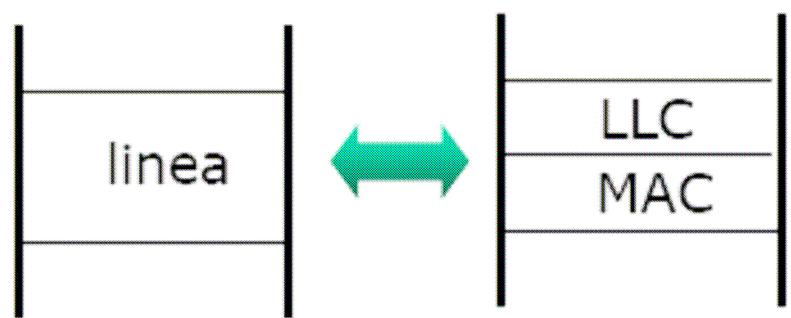
Livello di linea – Data link layer

- Le funzionalità richieste allo strato di linea sono:
 - I. Controllo d'errore. Il ricevitore deve essere in grado di accorgersi che l'informazione rivelata non coincide con l'informazione trasmessa (la causa può essere di varia natura, come interferenza, rumore dei dispositivi elettronici,...) e quindi adottare appropriate contromisure affinché il canale di comunicazione percepito dal livello di rete sia il più possibile libero da errori.
 - II. Controllo di flusso. Il trasmettitore ed il ricevitore devono impiegare delle procedure di *handshaking* per gestire *ordinatamente* il trasferimento di informazione *punto-punto*.

Livello di linea – Data link layer

III. Condivisione del mezzo di propagazione comune. Se il mezzo di propagazione non è esclusivamente dedicato alla comunicazione punto-punto, occorrono delle strategie per regolare l'accesso dei vari terminali alla risorsa comune.

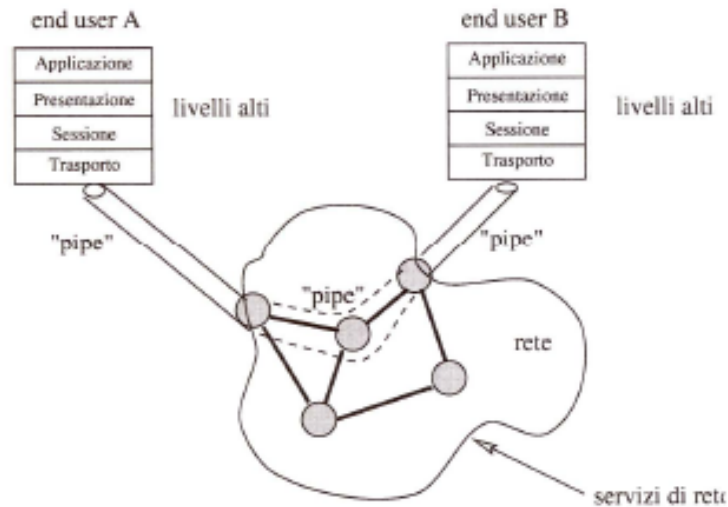
- Il numero e la complessità delle operazioni richieste allo strato di linea sono state tali da richiedere una suddivisione in due sottostrati, il *medium access control* (MAC) sublayer ed il *logical link control* (LLC) sublayer.



Livello di rete – Network layer

- L'obiettivo fondamentale dello strato di rete è quello di trasferire i pacchetti dati dalla sorgente alla destinazione; in sostanza lo strato di rete si occupa di trasferire geograficamente i pacchetti.
- Un problema strettamente legato alla precedente funzionalità è il controllo della congestione. Dato che la quantità di risorse (sia hardware che software) disponibile per trasferire fisicamente l'informazione è limitata, occorre “instradare” i pacchetti in modo da non sovraccaricare una linea di comunicazione lasciandone, ad esempio, inutilizzata un'altra.
- Un altro problema è rappresentato dalla tariffazione; si pensi alla molteplicità delle offerte dalle varie compagnie che offrono servizi di telecomunicazione.

Livello di rete – Network layer



- **Osservazione:** le nozioni di reti operanti a circuito virtuale piuttosto che in modalità datagram trovano la loro collocazione logica proprio in questo livello.

Livello di trasporto – Transport layer

- Il livello di trasporto ha il compito di fornire al livello superiore un servizio efficiente di trasporto dati da sorgente a destinazione. In questo senso la sua funzionalità potrebbe apparire molto simile alle funzionalità offerte dallo strato di linea, ma occorre ricordare che i precedenti tre livelli sono implementati per lo più in hardware e sono controllati esclusivamente dal gestore della rete, mentre i restanti quattro livelli sono eseguiti da un software che risiede nello stesso terminale su cui “gira” l’applicazione.
- In sostanza lo strato del trasporto garantisce che i servizi resi agli strati superiori risultino più affidabili di quelli offerti dallo strato di rete; in altre parole, problematiche come pacchetti persi, dati corrotti, reset inviati dal livello di rete possono essere rilevati e schermati all’applicazione finale.

Livello di trasporto – Transport layer

- Per comprendere pienamente la necessità delle funzionalità offerte dallo strato di trasporto basta osservare che gli utenti non hanno alcun controllo sullo strato di rete.

Livello di sessione – Session layer

- Il livello di sessione fornisce dei servizi a valore aggiunto, come:
 - Controllo del dialogo; ad esempio, nel caso di comunicazione half-duplex tale livello coordina i turni di trasmissione
 - Sincronizzazione delle trame; ad esempio possono essere inseriti dei check-points nel flusso di dati da trasmettere al fine di eseguire dei trasferimenti dati in modalità incrementale

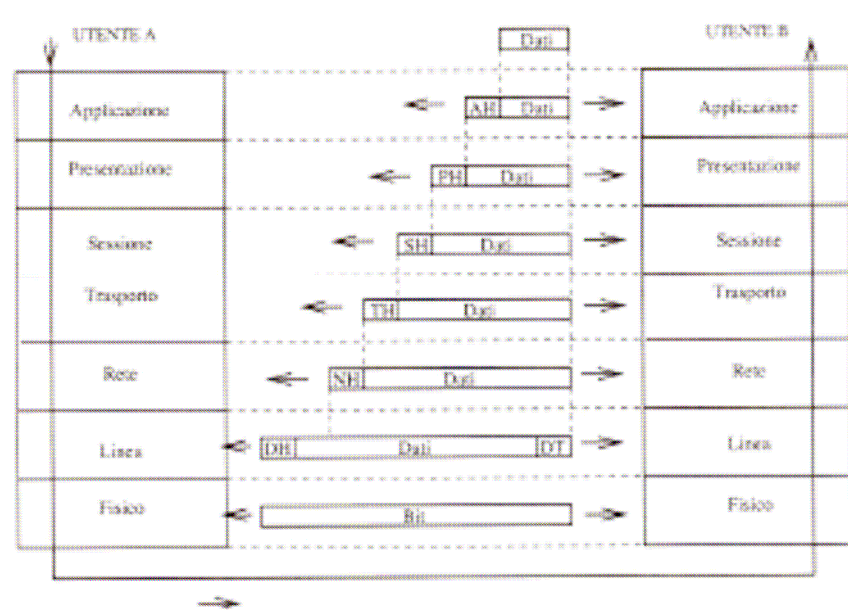
Livello di presentazione – Presentation layer

- Il livello di sessione si occupa della semantica dei dati, infatti
 - Gestisce la conversione dei formati di dati
 - Comprime (decomprime) i dati
 - Crittografa i dati al fine di garantire la privacy e la sicurezza della comunicazione

Livello di applicazione – Application layer

- In questo livello sono comprese le applicazioni così come sono comunemente conosciute, tra le quali troviamo il file transfer protocol (FTP), i servizi di posta elettronica, il web browsing, ...

Comunicazione verticale e comunicazione orizzontale



- Il flusso *fisico* dei dati è chiaramente verticale, tuttavia l'interpretazione dei dati avviene orizzontalmente.

Comunicazione verticale e comunicazione orizzontale

- Prendiamo come esempio il trasferimento di una email. Il testo della lettera viene passato al livello di applicazione che si incarica del trasferimento. Tale livello aggiunge un intestazione (header) ai dati reali per svolgere peculiari funzionalità (ad esempio si vuole marcare una email come urgente); il tutto viene poi passato al livello di presentazione che aggiunge un proprio header al fine, ad esempio di indicare quale algoritmo di crittazione dei dati è stato impiegato per rendere sicuro il contenuto dell'email. Dopodichè il livello di sessione introduce un header per permettere, ad esempio la sincronizzazione dei processi dal trasmettitore al ricevitore (nel senso che quando l'email è stata completamente inviata, al ricevitore arriva un segnale di posta in arrivo).

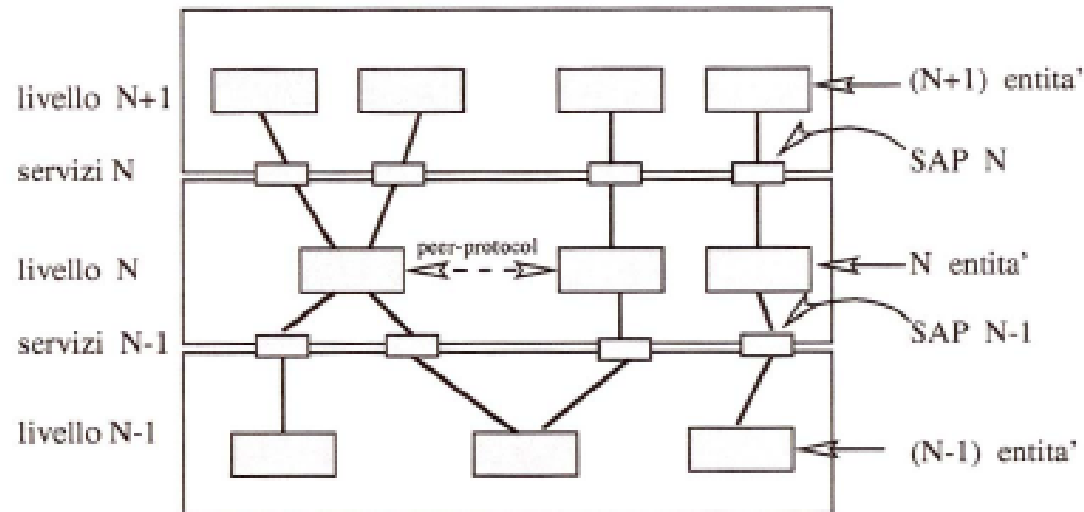
Comunicazione verticale e comunicazione orizzontale

Poi il livello del trasporto si occupa dell'affidabilità della comunicazione, per fare in modo che il contenuto della email giunga integro al ricevitore. Il livello di rete inserisce nel proprio header l'indirizzo a cui deve essere consegnata la lettera. Solitamente, il livello di linea inserisce sia un header, ad esempio per controllare il flusso di informazione, che un trailer per effettuare il controllo d'errore. Infine lo strato fisico ha il compito di trasmettere lo stream di bit "accumulato" dall'intero stack.

- È importante notare come ciascun livello non interpreti in alcun modo le informazioni provenienti dagli altri strati, ma si limiti ad estrarre l'header di competenza al fine di portare a termine le funzionalità richieste.

Concettualmente si tratta di un colloquio tra pari, *peer-to-peer*.

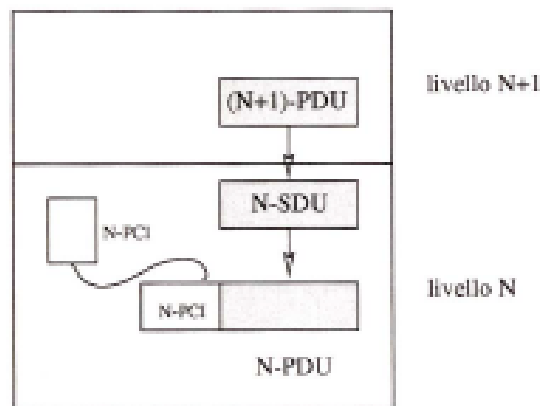
Terminologia OSI



- L'entità può essere sia un frammento di software eseguito da un'unità logica programmabile (ad esempio un microprocessore) che una porzione di chip hardware (ad esempio un controller USB).
 - Entità del medesimo livello che risiedono su macchine diverse sono dette peer-entities.
-

Terminologia OSI

- Le entità del livello N forniscono un servizio alle entità di livello $N+1$
- I servizi sono resi disponibili in corrispondenza ai Service Access Points (SAP)

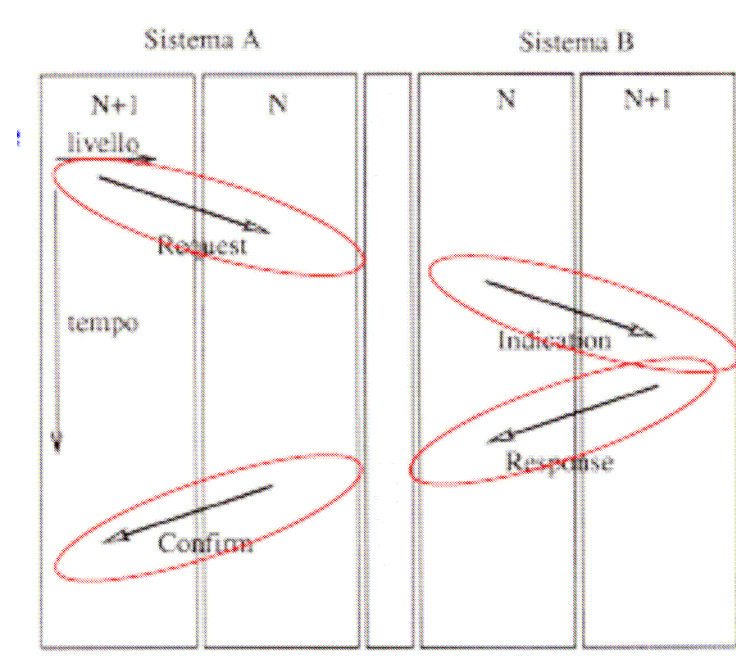


- Nel livello N , alla *protocol data unit* (PDU) proveniente dal livello $N+1$ viene aggiunta la *protocol control information* (PCI) relativa al livello N per formare la *service data unit* (SDU).

Terminologia OSI

- È fondamentale un'astrazione del meccanismo di incapsulazione esposto precedentemente.
- È importante sottolineare il fatto che una PDU possa essere frammentata in più parti per dare luogo a più SDU.

OSI: le primitive di servizio



- Per quanto detto precedentemente, servizio e funzionalità sono sinonimi.
- Un servizio è formalmente specificato da un insieme di primitive (precedentemente chiamate operazioni) che i processi utente possono invocare per accedere al servizio.

OSI: le primitive di servizio

- Le primitive portano il servizio ad eseguire determinate azioni o riferire quelle prese da entità di pari strato.
- Esempiole primitive socket utilizzate per TCP in Berkeley UNIX.

Primitiva	Significato
SOCKET	Crea un nuovo punto finale di comunicazione
BIND	Associa un indirizzo locale ad un socket
LISTEN	Annuncia la capacità di accettare connessioni e dà la dimensione della coda
ACCEPT	Blocca il chiamante fino all'arrivo di un tentativo di connessione
CONNECT	Tenta in modo attivo di stabilire una connessione
SEND	Invia alcuni dati sulla connessione
RECEIVE	Riceve alcuni dati sulla connessione
CLOSE	Rilascia la connessione

OSI: le primitive di servizio

- Praticamente, in questo caso, le primitive sono delle funzioni scritte in linguaggio C che fanno parte del kernel del sistema operativo .
- Esempio microchip per la comunicazione radio secondo lo standard Bluetooth (implementazione del livello fisico)
- Esempio algoritmo di routing eseguito in un router

Tre concetti fondamentali per il modello OSI

- Ciascun livello nel modello OSI garantisce uno o più *servizi* al livello immediatamente superiore; risponde alla domanda “Che cosa ?”
- L'*interfaccia* definisce dove sono resi disponibili tali servizi, quali parametri fornire e quali risultati aspettarsi; risponde alla domanda “Dove ?”
- Per comprendere meglio il concetto di interfaccia si pensi alla programmazione ad oggetti: una funzione esegue una determinata operazione (servizio), ma per far ciò ha bisogno sia di parametri di ingresso che di fornire dei risultati in uscita; la dichiarazione della funzione descrive l'interfaccia da utilizzare per avvalersi dell'oggetto.

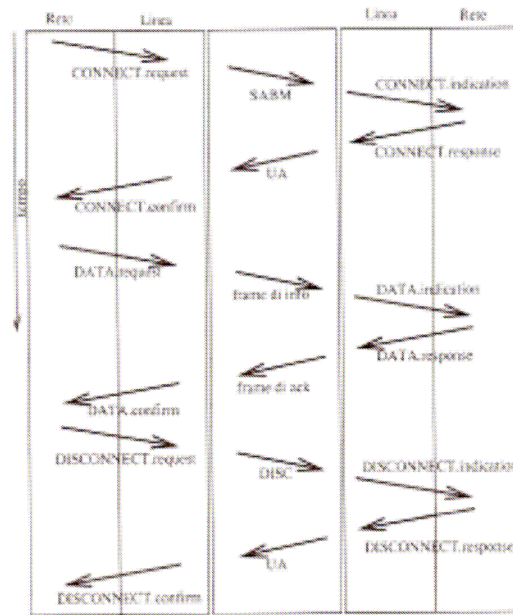
Tre concetti fondamentali per il modello OSI

- Il concetto di interfaccia non è esclusivamente software, infatti basti pensare ad un chip che implementa un controller USB: esso ha bisogno, infatti, come ingressi dei segnali scambiati con la periferica, e fornisce in uscita dei dati che vengono immessi sul data bus dell'host. Praticamente l'interfaccia del controller è un insieme di pin in ingresso e uscita dal chip.
- Un *protocollo* definisce il formato e l'ordine dei messaggi scambiati tra due o più entità, così come definisce le azioni intraprese in corrispondenza alla trasmissione e/o ricezione di un messaggio o all'occorrenza di altri eventi. In altre parole il protocollo definisce le modalità con cui viene fornito il servizio e stabilisce su quali regole si deve fondare il processo di comunicazione tra peer-entities. Risponde alla domanda "Come?"

Servizi connectionless e connection-oriented

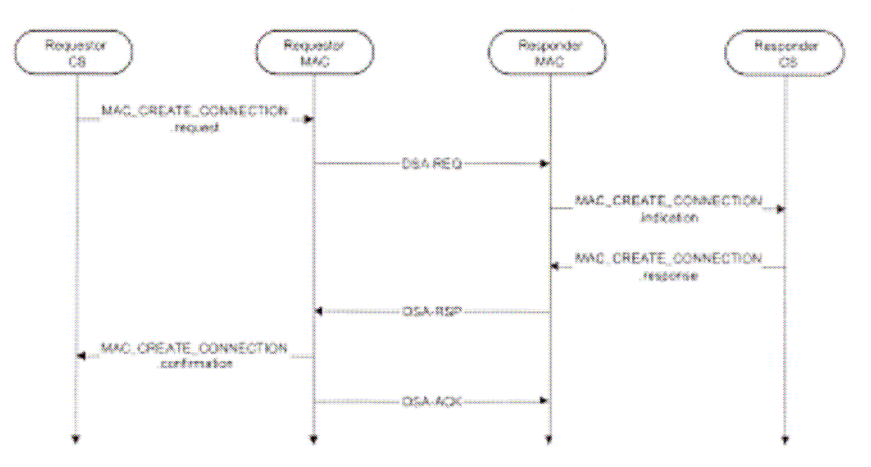
- I servizi resi da un livello, qualsiasi esso sia, possono essere di tipo *connectionless* o *connection-oriented*.
- Nel caso di servizi connection-oriented si hanno sempre tre distinte fasi logiche nell'esecuzione del servizio: apertura della connessione, trasferimento dei dati e chiusura della connessione.

Esempio di impiego del modello OSI



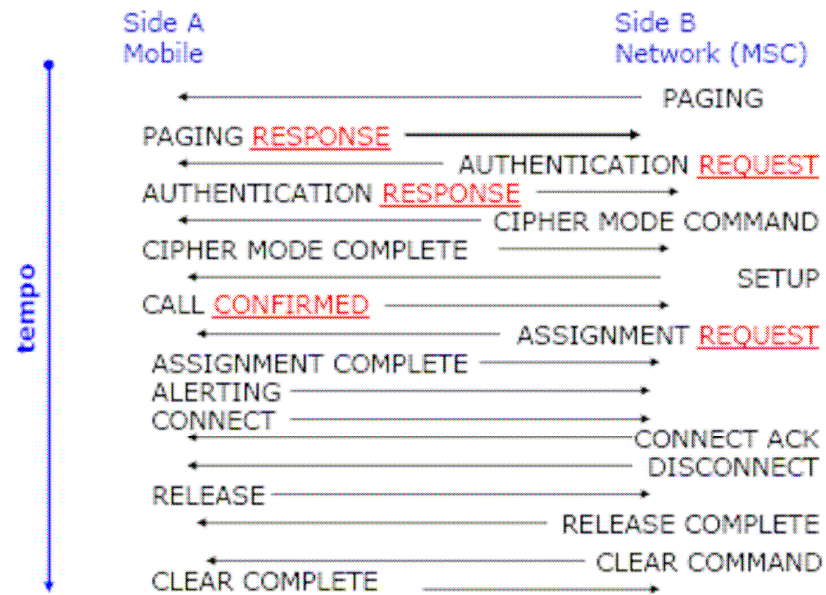
- Scambio di messaggi tra lo strato di linea e lo strato di rete in un link cablato punto-punto che utilizza il protocollo HDLC (che è un protocollo connection-oriented)

Esempio di impiego del modello OSI



- Primitive di servizio per l'apertura della connessione nel protocollo IEEE 802.16 (WiMax); si noti come il sottostrato MAC sia connection-oriented

Esempio di impiego del modello OSI



- Messaggi di segnalazione che vengono scambiati tra un terminale GSM ed il Mobile Switching Center nel caso in cui l'utente radiomobile riceva una chiamata.

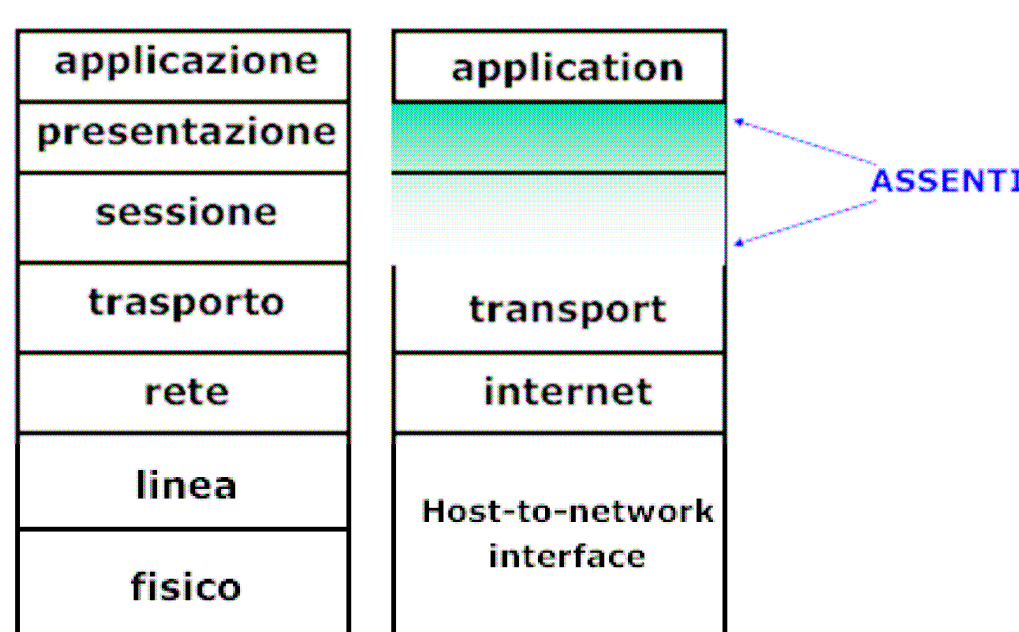
Il modello di riferimento TCP/IP

- Negli anni '50, in piena Guerra Fredda, il Dipartimento per la Difesa (Department of Defense, DoD) USA, temendo un attacco nucleare da parte dell'URSS, si rende conto di avere un sistema di smistamento dei comandi basato interamente sulla rete telefonica; tale sistema era quindi organizzato in modo gerarchico e pochi attacchi mirati potevano facilmente dividere il Paese. Il DoD inizia a finanziare la ricerca per lo sviluppo di una rete in grado di continuare a funzionare fintanto che il terminale sorgente ed il terminale destinazione sono integri. In particolare il DoD sente la necessità collegare reti eterogenee, quindi linee telefoniche con ponti radio e link satellitari; inoltre è necessario poter accedere a servizi di diversa natura, come il trasferimento di file e la trasmissione di segnali vocali.

Il modello di riferimento TCP/IP

- Nel 1968 l'Advanced Research Projects Agency (ARPA) affida la costruzione di una rete a commutazione di pacchetto alla società BBN, mentre affida lo sviluppo del software ai dottorandi delle università, che negli anni sviluppano la suite protocollare TCP/IP.

Il modello di riferimento TCP/IP



- Nel modello TCP/IP lo strato host-to-network non viene preso in considerazione. Si suppone semplicemente che vi siano protocolli hardware a sufficienza per inviare pacchetti IP sulla rete.

Internet layer

- In tale strato si collocano diversi protocolli, ma il più importante è senza ombra di dubbio l'**Internet Protocol (IP)**. Il suo compito è quello di instradare i *datagrammi* sulla rete. Esso garantisce un servizio
 - ✓ Connectionless
 - ✓ Best-effort

Transport layer

- Vi sono due protocolli fondamentali, il *transport control protocol* (TCP) e lo *user datagram protocol* (UDP).
- La caratteristica principale del TCP è l'affidabilità (reliability), nel senso che adotta appropriate contromisure (meccanismo degli acknowledgement) per consegnare al destinatario l'informazione in modo corretto, libero da errori di trasmissione di qualsiasi natura.
- Il TCP è connection-oriented, cioè vi è prima una fase di apertura della connessione in cui vengono riservate le risorse per la comunicazione, poi una fase di trasferimento dei dati, ed infine una fase di chiusura ordinata della comunicazione, in cui vengono rilasciate le risorse precedentemente allocate e si assicura che il trasferimento dati sia andato a buon fine.

Transport layer

- Il TCP implementa il controllo di flusso, evitando che una sorgente veloce “inondi” di informazioni una destinazione lenta, causando di conseguenza la perdita di pacchetti.
- Infine il TCP effettua un controllo della congestione, cioè monitora la situazione dei router che inoltrano i pacchetti spediti ed evita che inviarne dei nuovi se le risorse di rete sono quasi esaurite, sempre al fine di evitare la perdita di informazione.
- Il protocollo UDP è inaffidabile (unreliable), nel senso che non garantisce in alcun modo che i dati trasmessi vengano consegnati correttamente.
- Il protocollo UDP è connectionless, e quindi non prenota le risorse necessarie alla comunicazione in fase di apertura.

Transport layer

- A seconda delle applicazioni viene utilizzato un protocollo piuttosto che l'altro. Ad esempio
 - ✓ Il TCP viene impiegato per i file transfer e le email per sfruttarne l'affidabilità
 - ✓ Essendo il protocollo UDP semplice e snello, si sfrutta la sua velocità nella consegna dei pacchetti nelle applicazioni multimediali, come il video-streaming ed il voice-over-IP (VoIP)
- Il livello di applicazione è identico a quello descritto nel modello di riferimento ISO-OSI

Critica al modello OSI

- I difetti principali del modello OSI sono quattro:
 1. La sua introduzione è stata poco tempestiva, infatti ai tempi della sua comparsa (negli anni '80) lo stack protocollare TCP/IP era già funzionante ed implementato da numerosi costruttori, che non se la sono sentiti di offrire una ulteriore pila di protocolli nei propri prodotti.
 2. Alcune scelte tecnologiche sono scadenti, infatti vi sono troppi strati, di cui alcuni (come lo strato di linea e quello di rete) sono troppo complessi, mentre alcuni altri (come lo strato di presentazione e lo strato di sessione) sono quasi vuoti. Inoltre certi servizi (come il controllo d'errore) appaiono in più strati introducendo una ridondanza ingiustificata nello stack.

Critica al modello OSI

3. Le implementazioni iniziali erano molto complesse e costose, assolutamente sconvenienti se paragonate alle versioni estremamente efficienti, ma soprattutto gratuite, dello stack TCP/IP. La qualità delle implementazioni è poi migliorata nel tempo, ma il gap tecnologico non è ancora stato colmato.
4. Le organizzazioni Statunitensi riuscirono a diffondere meglio il proprio standard.

Critica al modello TCP/IP

- I difetti principali del modello TCP/IP sono:
 1. Non distingue chiaramente i concetti di servizio, interfaccia e protocollo, per questo non è utile ai fini della progettazione di sistemi basati su nuove tecnologie (come lo standard IEEE 802.16 adottato nel Dicembre 2006 in Italia).
 2. Il livello sottostante l'internet network layer non è esattamente un livello, quanto piuttosto un'interfaccia, che comunque non specifica in alcun modo i servizi che devono essere offerti.
 3. Il livello fisico ed il livello di linea non sono distinti, anche se devono assolvere funzionalità profondamente differenti.

Conclusioni

- Il modello OSI risulta estremamente utile per inquadrare e studiare le diverse problematiche che si devono affrontare nella progettazione di una rete di telecomunicazione. Tuttavia diversi fattori (precedentemente citati) hanno impedito la diffusione dei protocolli OSI.
- Il modello TCP/IP è del tutto inutile al livello teorico, per la suite protocollare TCP/IP è estremamente efficiente e funzionale, per questo è alla base della quasi totalità delle reti implementate attualmente.