ARCHITETTURA DELLE RETI

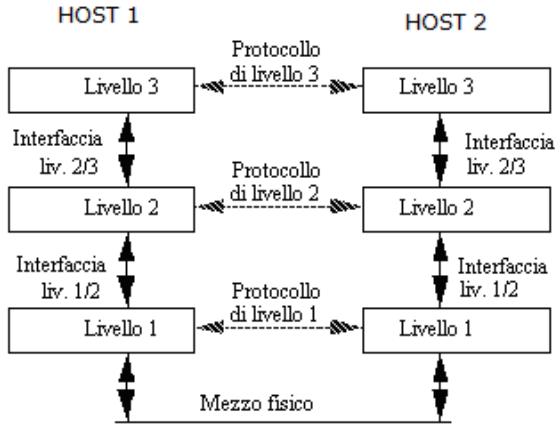
Le applicazioni di rete sono programmi che permettono lo scambio di informazioni tramite la rete. Gli esempi sono tanti, dai browser che consentono la navigazione in Internet ai programmi per il download di file. Questi programmi sono come la punta di un iceberg: rappresentano la parte a contatto con l'utente e si appoggiano su una struttura molto complessa di strati di software, ciascuno con un suo compito. Questa struttura è l'architettura di rete.

Un'***architettura di rete*** specifica i componenti (hardware e software), le funzioni che questi componenti svolgono e come interagiscono tra loro.

MODELLO A STRATI DEL SOFTWARE DI RETE

Sappiamo che per semplificare un problema complesso conviene scomporlo in parti, ciascuna delle quali è suscettibile di una trattazione più circoscritta, come avviene con la suddivisione di un programma in funzioni secondo la metodologia top-down.

Per ridurre la complessità del progetto, il software di gestione delle reti viene organizzato a ***livelli*** (o ***strati*** o ***layers***), ciascuno costruito sopra il precedente a formare una ***pila*** (***stack***).
Il numero e le funzioni dei livelli caratterizzano un'architettura di rete e possono essere diversi fra un tipo di rete ed un'altra, ma i principi generali sono simili per tutte:



- un livello offre **servizi** al livello superiore, nascondendo i dettagli implementativi di tali servizi;
- il livello **n** su un host porta avanti una conversazione col livello **n** su un altro host, eseguendo alcuni compiti in trasmissione ed altri, complementari, in ricezione.

Le regole e le convenzioni che governano la conversazione sono indicate come ***protocollo di livello n***. I processi software o i dispositivi hardware che effettuano tale conversazione si chiamano ***entità*** e le entità di pari livello si dicono peer entity.

Il blocco dati che le peer entity si scambiano si chiama ***PDU*** (***Protocol Data Unit***).

Servizio e protocollo possono essere così definiti:

Servizio	insieme di operazioni che un livello offre al livello superiore.
Protocollo	insieme di regole che definiscono il contenuto (formato e significato) e il modo dello scambio di informazioni tra le entità di pari livello.

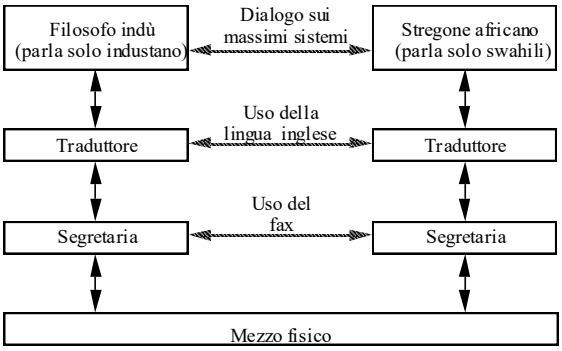
Le peer entity pensano concettualmente ad una comunicazione orizzontale fra loro (comunicazione logica), basata sul protocollo del proprio livello; in realtà il dialogo fra due peer entity di livello n avviene materialmente tramite i servizi offerti dal livello n-1 (comunicazione fisica verticale): come illustrato dalla figura di sopra, non c'è un trasferimento diretto di dati al livello n. Ogni livello di HOST1 passa i dati al livello sottostante, fino al livello più basso (livello 1).
Al di sotto del livello 1 c'è il mezzo fisico, attraverso il quale i dati vengono effettivamente trasferiti da HOST1 ad HOST2.
Quando arrivano a HOST2, i dati vengono passati da ogni livello, a partire dal livello 1, a quello superiore, fino a raggiungere il livello più alto della pila.

VANTAGGI DEL MODELLO A STRATI

- Riduzione della complessità del progetto
- Indipendenza dei vari strati: ogni strato ha un compito specifico, diverso dagli altri e la sua struttura non è vincolata da quella degli altri
- Facilità nel modificare uno strato, senza toccare gli altri, a patto di non modificare l'interfaccia

DUE ESEMPI

Per comprendere i meccanismi basilari di funzionamento del software di rete si può pensare ad un filosofo indù che vuole conversare con uno stregone africano, come illustrato dalla figura a fianco. In questo esempio il filosofo e lo stregone sono due entità del livello più alto; i due traduttori (anche loro peer entity) sono entità che forniscono il servizio di traduzione alle entità di livello superiore e così via.



Altro esempio: quando parliamo, la nostra mente concepisce e ordina il pensiero che vogliamo esprimere (livello logico). Questo pensiero viene trasformato in una serie di comandi per il nostro apparato di fonazione (gola, corde vocali, lingua, …) che emette una serie di onde sonore (livello fisico). Queste onde sonore si propagano nell'aria (mezzo trasmissivo) ed arrivano all'orecchio del nostro interlocutore (livello fisico) che le trasforma in impulsi nervosi che arrivano al suo cervello che ricostruisce il pensiero iniziale (livello logico).

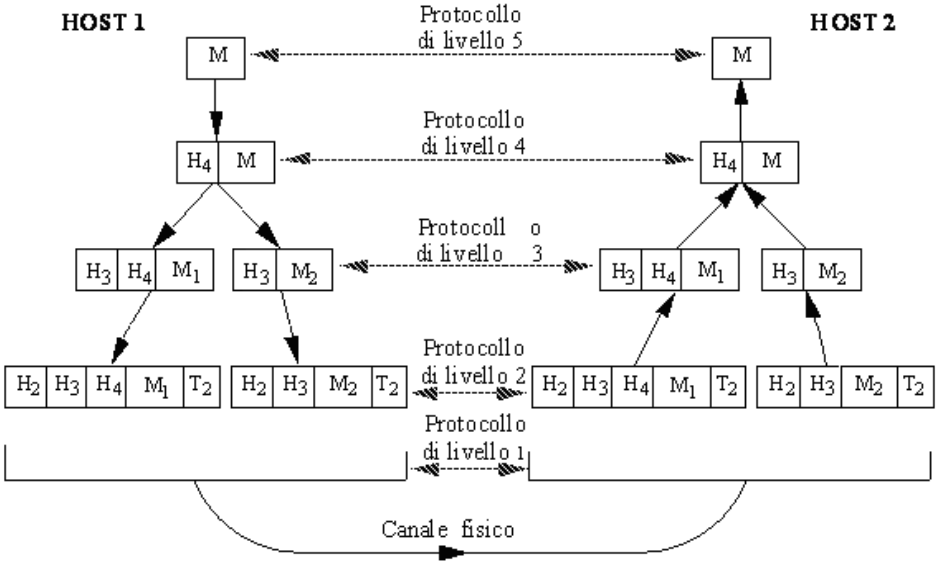
INCAPSULAMENTO DEI DATI ATTRAVERSO I LIVELLI

La comunicazione fra due host avviene con una modalità che, almeno in linea di principio, è uguale in tutte le architetture di rete e che prevede che:

- i dati da trasmettere siano suddivisi in blocchi;
- ogni livello possa aggiungere informazioni di controllo in testa (***header***) o in coda (***trailer***) (incapsulamento o imbustamento);
- i blocchi possano essere ulteriormente suddivisi (frammentazione).

Vediamo un esempio, ipotizzando una strutturazione in 5 livelli:

1. il programma applicativo (livello 5) deve mandare un messaggio M alla sua peer entity;
2. il livello 5 consegna M al livello 4 per la trasmissione;
3. il livello 4 aggiunge un header in testa al messaggio (si dice che il messaggio è messo nella busta di livello 4);
4. il livello 4 consegna il risultato al livello 3;
5. il livello 3 può trovarsi nella necessità di frammentare i dati da trasmettere in unità più piccole, a ciascuna delle quali aggiunge il suo header;
6. il livello 3 passa i pacchetti al livello 2;
7. il livello 2 aggiunge il proprio header (e magari un trailer) e lo passa al livello 1;
8. il livello 1 trasforma i dati in segnali adatti ad essere spediti sul canale fisico;
9. nell'host di destinazione i pacchetti fanno il percorso inverso, con ogni livello che elimina (elaborandoli) l'header ed il trailer di propria competenza e passa il resto al livello superiore.



La figura rappresenta i passaggi sopra esposti. Non è detto che in tutte le architetture il livello 3 frammenti il messaggio e che solo il livello 2 aggiunga un trailer; sicuramente in tutte le architetture i vari livelli aggiungono le loro informazioni di controllo, sotto forma di header ed eventualmente di trailer e ci sono livelli che hanno il compito di fare in modo che i pacchetti da trasmettere non superino una determinata dimensione e pertanto suddividono (frammentano) un pacchetto qualora sia troppo grande.

Le informazioni di controllo contenute negli header e nei trailer possono essere, ad esempio:

- indirizzo del mittente e del destinatario,
- numero di sequenza del messaggio,
- dimensione del messaggio,
- time stamp, cioè l'indicazione della data e dell'ora in cui il messaggio viene creato,
- priorità,
- campo per il controllo degli errori di trasmissione.

ASPETTI DI PROGETTO DEI LIVELLI

Una buona progettazione di un'architettura di rete deve:

- minimizzare le informazioni da trasferire tra i livelli;
- rendere possibile modificare l'implementazione di un livello con una più attuale che offra gli stessi servizi, senza dover modificare il resto dell'architettura (ad es. sostituire le linee telefoniche con canali satellitari).

Per quanto riguarda il numero di livelli, possiamo dire che:

- più sono i livelli, maggiore è la ridondanza di informazioni di controllo;
- meno sono i livelli, maggiore è la complessità di ciascuno.

Decisioni di progetto vanno prese nei vari livelli. Le principali sono:

1. Metodi di identificazione di mittente e destinatario (*indirizzamento*), in ogni livello.
2. Meccanismi per il controllo degli *errori di trasmissione*; è possibile:
   - rilevarli oppure no;
   - correggerli oppure no;
   - avvertire il mittente oppure no.
3. Meccanismi per il mantenimento o la ricostruzione dell'*ordine originario* dei dati.
4. Meccanismi di sincronizzazione per regolare le *velocità* di sorgente e destinazione.
5. Decisioni sulla *dimensione* (minima o massima) dei blocchi di dati da inviare e su come eventualmente frammentarli.

TIPI DI SERVIZIO: CONNESSO/NONCONNESSO-CONFERMATO/NON CONFERMATO

Il tipo di servizio offerto da un livello a quello superiore può essere:

- connesso (orientato alla connessione o connection-oriented) oppure non connesso (connectionless);
- confermato oppure non confermato.

Servizio connesso

In un servizio connesso due nodi coinvolti in una comunicazione sono collegati "direttamente", anche se in modo virtuale. Un servizio di questo tipo si sviluppa in 3 fasi:

1. instaurazione della connessione (set-up), coinvolgendo tutti i nodi di commutazione (router) che sono sul percorso individuato;
2. trasmissione dei dati, che seguono lo stesso percorso e arrivano nello stesso ordine in cui sono partiti. Non è necessario che i singoli blocchi di dati contengano l'indirizzo del destinatario, che è determinato durante la fase di set-up;
3. rilascio della connessione, attività che coinvolge di nuovo tutti i nodi sul cammino.

Un tipico esempio di servizio di servizio connection-oriented è quello della telefonia: quando si digita il numero telefonico della persona da contattare, la rete instaura un collegamento diretto tra i due apparecchi telefonici e la voce viene trasferita su questo collegamento che rimane attivo per tutta la durata della conversazione.

Servizio non connesso

I pacchetti viaggiano indipendentemente gli uni dagli altri, possono prendere strade diverse ed arrivare in ordine diverso da quello di partenza o non arrivare affatto.

La fase è una sola:

1. invio del pacchetto.

Ogni pacchetto deve contenere l'indirizzo del destinatario.

Un tipico esempio di servizio di servizio connectionless è quello postale: se una persona invia più lettere allo stesso destinatario, ogni lettera viaggia e viene recapitata in modo indipendente dalle altre.

CONFRONTO FRA PROTOCOLLI CONNESSI E NON CONNESSI

	CONNESSO	NON CONNESSO
SET UP INIZIALE	SI	NO
INDIRIZZO DI DESTINAZIONE	DURANTE IL SET-UP	NEI PACCHETTI
ORDINE DEI PACCHETTI	GARANTITO	NON GARANTITO

Servizio confermato (reliable o affidabile)

Un servizio ***confermato*** non perde mai dati, cioè assicura che tutti i dati spediti verranno consegnati al destinatario e senza errori. Ciò in genere richiede che il ricevente invii un ***acknowledgement*** abbreviato in ***ack*** (***conferma*** –per questo si dice confermato**-**) al mittente. Si introduce ovviamente overhead, che in certe situazioni può non essere desiderabile.

Servizio non confermato

Non offre la certezza che i dati spediti arrivino a destinazione o che siano esenti da errori.

Si noti che se un livello non offre nessun servizio affidabile, qualora tale funzionalità sia desiderata dovrà essere fornita da uno dei livelli superiori, cosa che accade spesso.

Unendo i concetti di orientato alla connessione e affidabilità si ottengono quattro tipi di servizi:

- ***connesso e confermato***: garantisce la consegna corretta e in ordine dei dati. Adatto per il trasferimento di file (non ci devono essere errori, non devono mancare pezzi e il file non deve essere "rimescolato");
- ***connesso non confermato***: garantisce che i pacchetti arrivino in ordine, ma non garantisce che siano esenti da errori. Usato nelle trasmissioni in streaming, in cui i dati audio/video vengono riprodotti man mano che si scaricano. Queste sono trasmissioni isocrone, in cui le relazioni temporali fra i bit del flusso devono essere mantenute ed è meglio qualche disturbo ogni tanto, piuttosto che interruzioni momentanee, ma avvertibili, del flusso di dati;
- ***non connesso confermato***: i pacchetti sono consegnati in modo indipendente l'uno dall'altro, ma è prevista la conferma della ricezione di ciascuno. Adatto a quando si invia un breve messaggio e si vuole essere assolutamente sicuri che è arrivato;
- ***non connesso non confermato*** (detto anche ***datagram*** service): non garantisce né la consegna corretta del pacchetto né l'ordine. Utilizzabile quando non importa se qualche messaggio si perde.

INTEROPERABILITA' E STANDARD

Attualmente le reti sono ***sistemi aperti***: ad esse ci si può collegare con qualunque tipo di computer e sistema operativo, purché dotato degli opportuni protocolli. Al contrario, le prime architetture di reti degli anni '60 e '70 del secolo scorso erano sistemi chiusi, di tipo proprietario. Un'architettura proprietaria generalmente è incompatibile con architetture diverse, in quanto si basa su scelte arbitrarie del costruttore, che non rende pubbliche le specifiche per cui nessun altro può produrre software e apparati compatibili.
Proprio i problemi causati dalla mancanza di interoperabilità hanno stimolato lo studio di architetture aperte.

Architetture aperte sono definite da ***standard***, che sono le regole comuni che, se vengono rispettate, permettono la comunicazione. Uno standard fornisce le linee guida a cui tutti i costruttori si adeguano per assicurare che tutti possano connettersi e comunicare.

Uno standard si dice *de iure* (giuridico) quando è emanato da un ente internazionale che si occupa di standardizzazione. Le specifiche di uno standard de iure sono di pubblico dominio e

ogni costruttore può proporne una propria implementazione.

Uno standard *de facto* (di fatto) si è imposto non in quanto emanato da un ente di standardizzazione, ma per la sua larghissima diffusione. Le sue specifiche sono comunque di pubblico dominio, per cui diversi costruttori possono proporre la propria implementazione, garantendo l'interoperabilità.

Tante sono le autorità che si occupano di standard; tra tutte ricordiamo:

- ***ANSI (American National Standards Institution)***: rappresentante USA nell' ISO;
- ***EIA* Electronic Industries Alliance** - Associazione industriale USA del settore elettronico
- ***IEEE (Institute of Electrical and Electronic Engineers)***: organizzazione professionale mondiale degli ingegneri elettrici ed elettronici che ha l'obiettivo di cercare nuove applicazioni in ambito scientifico (informatico, telecomunicazioni, biomedico,…). Le sue pubblicazioni rappresentano buona parte della documentazione ingegneristica mondiale. Ha gruppi di standardizzazione sulle reti, che hanno prodotto diversi standard.
- ***ISO (International Standard Organization)***: il principale ente di standardizzazione internazionale, che si occupa fra l'altro anche di reti;

Attualmente nello studio delle reti sono di fondamentale importanza:

Open System Interconnection* (*OSI*) *Reference Model
Internet Protocol Suite (o ***architettura TCP/IP***)

Il primo è il modello di riferimento per lo studio del software di rete, emanato da un ente di standardizzazione, mentre la seconda è l'architettura di rete diffusa a livello mondiale, impostasi come standard de facto.

Modello di riferimento e architettura di rete sono due cose diverse:

Modello di riferimento	definisce il numero, le relazioni e le funzioni (servizi) dei livelli, ma non definisce i protocolli effettivi.
Architettura di rete	come il modello di riferimento, in più definisce, livello per livello, i protocolli effettivi.

2.2.1) MODELLO DI RIFERIMENTO OSI

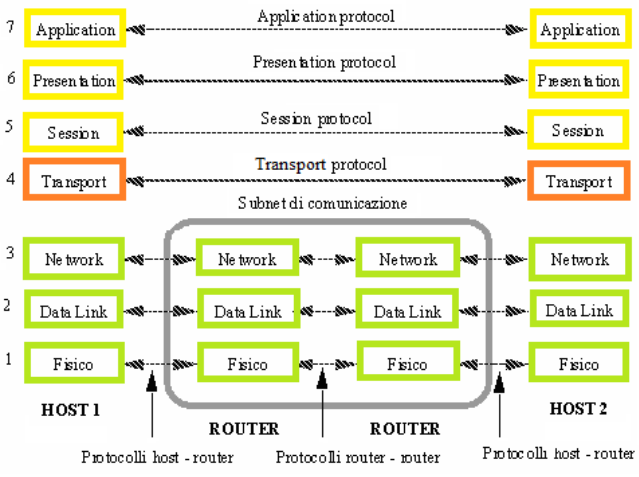
L'OSI Reference Model è lo standard de jure frutto di un lungo lavoro (terminato nel 1978) dell'ISO (International Standards Organization). Lo scopo di OSI era quello di fornire una base architetturale comune ai costruttori di software e di apparati di rete, in modo da superare le limitazioni dei sistemi proprietari e andare verso sistemi aperti. Il documento ISO 7498, dal titolo Basic Reference Model, descrive i principi base di questo standard.
La grande importanza dell'OSI consiste nell'aver introdotto il concetto di stratificazione e di interfacciamento tra i livelli ed è il riferimento teorico su come organizzare le architetture di rete.

OSI non include la definizione di protocolli specifici (che sono stati definiti in seguito, in documenti separati); definisce il numero dei livelli, le loro funzioni e come comunicano tra loro.

L'OSI consta di ***7 livelli***, che sono, dal basso verso l'alto: ***fisico***, ***collegamento dati*** (***data link***), ***rete*** (***network***), ***trasporto***, ***sessione***, ***presentazione*** e ***applicazione***.

I primi tre livelli (livelli inferiori: fisico, data link e rete) si occupano della gestione della sottorete di comunicazione; questi livelli sono presenti anche sui router. Spesso questi livelli, per ragioni di efficienza, sono implementati in firmware.

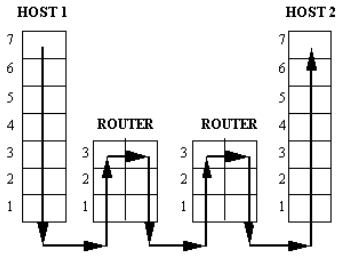
I nodi su cui sono presenti solo i livelli inferiori sono gli elementi di commutazione (router o intermediate system) che realizzano la sottorete di comunicazione.



Il quarto livello isola l'ambiente rete dall'ambiente applicazione.
Se su un nodo c'è il livello 4 (trasporto), allora deve essere presente tutta la pila di livelli (si tratta di un end system, cioè di un host a cui accede un utente).

I tre livelli superiori riguardano le applicazioni di rete.

Schematizziamo i compiti di ogni livello, i nomi dei blocchi di dati che ogni livello tratta, gli apparati e gli indirizzi relativi ad ogni livello:



LIVELLO	DESCRIZIONE	NOME PDU	APPARATI	INDIRIZZO
Application	Offre i servizi alle applicazioni di rete, come la posta elettronica, il trasferimento file, l'web	Messaggio		
Presentation	Fornisce una rappresentazione standard dei dati	Messaggio		
Session	Gestisce le sessioni	Messaggio		
Transport	È responsabile della consegna dell'intero messaggio da mittente a destinatario. Fornisce al livello superiore una connessione end to end, come se mittente e destinatario fossero connessi direttamente	Segmento		Socket
Network	Si occupa della consegna dei pacchetti dal mittente al destinatario, attraverso reti diverse	Pacchetto	Router	IP
Data Link	Provvede alla trasmissione dei dati tra due nodi sulla stessa rete	Frame	Bridge, switch	MAC
Fisico	Si occupa della trasmissione di un flusso di bit attraverso il mezzo fisico. I compiti sono di definire: - le caratteristiche fisiche delle interfacce di rete; - la codifica del segnale - la velocità di trasmissione - la topologia - il verso della trasmissione (simplex, full/ half duplex)	Bit	NIC, hub	

ARCHITETTURA TCP/IP

Prendendo come riferimento il modello ISO/OSI, un gruppo di ricercatori ha definito un modello architetturale semplificato, sempre a strati, cercando di avvicinare la teoria alla pratica, per interconnettere alcune reti già esistenti. Questo ha portato all'architettura nota come ***Internet Protocol Suite*** oppure ***architettura TCP/IP***, dal nome dei suoi due protocolli principali, che ha avuto un grande successo sia per la sua semplicità che per l'economicità dei dispositivi che la implementano.

Essa è un'architettura di rete, in quanto include i protocolli effettivi, che sono specificati per mezzo di documenti detti ***RFC*** (***Request For Comments***), reperibili in Internet.

È lo standard de facto che governa tutte le reti attuali.

Il TCP/IP prevede i livelli:

Physical	Il livello più basso non è specificato nell'architettura; prevede di utilizzare i vari standard disponibili per le varie piattaforme HW. Include le funzioni degli strati Physical e Data Link dell'OSI

Network	E' il livello portante dell'intera architettura. Il suo ruolo è permettere ad un host di immettere pacchetti in una qualunque rete e fare il possibile per farli viaggiare, indipendentemente gli uni dagli altri e magari per strade diverse, fino alla destinazione, che può essere anche in un'altra rete. Fornisce un servizio non connesso non confermato, di tipo datagram. Definisce il formato dei pacchetti ed una serie di protocolli, tra cui il principale è ***IP*** (***Internet Protocol***).
Transport	Consente la conversazione delle peer entity degli host sorgente e destinazione (end-to-end). Sono definiti due protocolli in questo livello: • ***TCP*** (***Transmission Control Protocol***): fornisce un servizio connesso e confermato (tutti i pacchetti arrivano corretti e nell'ordine giusto). Eventualmente frammenta i dati provenienti dal livello superiore in segmenti separati che vengono passati al livello Network. In arrivo, i segmenti vengono riassemblati in un flusso che viene passato al livello superiore. • ***UDP*** (***User Datagram Protocol***): è un protocollo non connesso e non confermato, i pacchetti possono arrivare in ordine diverso o non arrivare affatto. Viene usato quando la velocità è preminente rispetto all'eventuale perdita di dati.
Application	Nell'architettura TCP/IP non ci sono i livelli Session e Presentation, non ritenuti necessari. Sopra il livello Transport c'è direttamente il livello Application, che contiene tutti i protocolli di alto livello usati dalle applicazioni. I primi protocolli furono: • ***Telnet***: terminale virtuale; • ***FTP*** (***File Transfer Protocol***): trasferimento di file; • ***SMTP*** (***Simple Mail Transfer Protocol***) e ***POP*** (***Post Office Protocol***): posta elettronica. Successivamente se ne sono aggiunti altri, fra cui: • ***DNS*** (***Domain Name System***): mapping fra nomi di host e indirizzi IP numerici; • ***HTTP*** (***HyperText Transfer Protocol***): alla base del Word Wide Web.

Nascita di Internet

La storia di Internet inizia con ***Arpanet*** (nata a metà degli anni '60, ai tempi della guerra fredda), un progetto di ricerca finanziato dal DoD (Department of Defense) americano allo scopo di creare una rete estremamente affidabile anche in caso di catastrofi o eventi bellici che ne mettessero fuori uso una parte.

I requisiti di progetto stabiliti fin dall'inizio furono l'estrema affidabilità e tolleranza ai guasti, la possibilità di interconnessione di più reti) portarono alla scelta di una rete a commutazione di pacchetto (packet-switched) e basata su un livello internet non connesso.

Successivamente Arpanet si sviluppò incorporando altre reti e si avvertì l'esigenza di nuovi protocolli per superare l'inadeguatezza di quelli originari nella gestione delle problematiche di internetworking.

Nacque di conseguenza, verso la metà degli anni '70, l'architettura TCP/IP, che il giorno 1/1/1983 divenne lo standard di Arpanet. TCP/IP fu poi mantenuta anche per NSFNET, l'evoluzione di Arpanet (metà degli anni '80).

Il continuo aggiungersi di reti, ad Arpanet prima e ad NSFNET poi, ha creato quella che oggi viene comunemente chiamata Internet, costituita da milioni di host e utilizzata da decine di milioni di utenti.

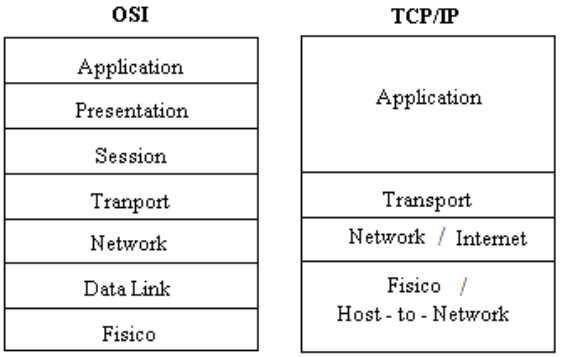
CONFRONTO FRA MODELLO DI RIFERIMENTO OSI E ARCHITETTURA TCP/IP

Basati entrambi sul concetto di pila di protocolli indipendenti, presentano funzionalità simili per i vari livelli, anche se TCP/IP è più snello rispetto ad OSI.

Mentre OSI nasce come modello di riferimento e i protocolli vengono solo successivamente, TCP/IP nasce coi protocolli.

I livelli TCP/IP hanno questa relazione con quelli OSI:

- Fisico e Data link vengono raggruppati in un generico Physical (detto anche Host to Network), che non fa propriamente parte dell'architettura.
- Network (detto anche Internet) di TCP/IP corrisponde a Network di OSI
- Transport di TCP/IP corrisponde a Transport di OSI
- Application di TCP/IP raggruppa le funzioni di Session, Presentation e Application di OSI



I protocolli OSI non sono riusciti ad affermarsi sul mercato per una serie di scelte infelici:

- di tempo: la definizione dei protocolli è arrivata troppo tardi, quando il TCP/IP si era già considerevolmente diffuso. Le aziende non se la sono sentite di investire risorse nello sviluppo di una ulteriore architettura di rete;
- tecnologiche: i sette livelli (e i relativi protocolli) sono stati dettati in realtà dalle architetture proprietarie allora esistenti (SNA dell'IBM e DECnet della Digital), più che da considerazioni di progetto. Per cui il progetto soffre di vari difetti:
  - grande complessità e conseguente difficoltà di implementazione;
  - inutili i livelli Session e Presentation;
  - non ottimali attribuzioni di funzioni ai vari livelli:
    - alcune funzioni appaiono in molti livelli (es. controllo errore e flusso in tutti i livelli);
    - altre funzioni mancano del tutto (ad es. sicurezza);
- di implementazione: le prime realizzazioni erano lente ed inefficienti, mentre contemporaneamente TCP/IP era molto ben implementato (e per di più gratis!). In effetti i protocolli dell'architettura TCP/IP sono stati implementati in modo efficientemente fin dall'inizio, per cui si sono affermati sempre più e quindi hanno goduto di un crescente supporto che li ha resi ancora migliori.

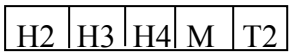
Ad ogni modo, neanche l'architettura TCP/IP è priva di problemi, in particolare alcune scelte iniziali di progetto, in particolare gli indirizzi IP a soli 32 bit.

In conclusione:

- OSI è ottimo come modello, mentre i suoi protocolli hanno avuto poco successo;
- TCP/IP è ottima (per ora) come architettura di rete.

DOMANDE

1. Cosa specifica un'architettura di rete?
2. Quali vantaggi offre la suddivisione di un'architettura di rete in livelli?
3. Cosa si intende per servizio?
4. Cosa si intende per protocollo?
5. Che cosa si intende per peer entity?
6. Nell'esempio del filosofo e dello stregone, quali sono le entità? Quale servizio offre l'entità "Segretaria"? Qual è il protocollo dei traduttori?
7. Come avviene la comunicazione tra due peer entity (comunicazione logica e fisica)?
8. PDU = (sigla) _____

9. Durante il processo di incapsulamento, cosa viene aggiunto ad un messaggio nel suo transitare attraverso la pila dei livelli?
10. Quali campi può contenere un header? E un trailer?
11. Qual è il vantaggio e lo svantaggio nell'avere un n. elevato di livelli?
12. Tutti i livelli mettono in atto meccanismi di controllo degli errori?
13. Quali livelli sono spesso implementati in firmware?
14. Dato il pacchetto 

, cosa rappresentano i diversi campi
15. I pacchetti spediti da un mittente ad uno stesso destinatario, seguono sempre lo stesso percorso?
16. In quali fasi si sviluppa un servizio connesso?
17. In quali fasi si sviluppa un servizio non connesso?
18. Un servizio connesso viene paragonato al sistema
19. Un servizio non connesso viene paragonato al sistema
20. Per un trasferimento di file, quale tipo di servizio è più idoneo?
21. Un servizio connesso non confermato per quali tipi di applicazioni risulta adatto?
22. Cos'è un ACK (acknowledgement)?
23. Secondo il modello OSI, se su un nodo c'è il livello 3, siamo certi di essere in presenza di un nodo finale (host o end system) con tutta la pila dei livelli?
24. Completa la seguente tabella, relativa ai livelli dell'OSI:

N	Nome Livello	Nome PDU (blocco dati)	Apparati	Indirizzi
7				

25. Elenca i compiti del livello Fisico
26. Cos'è il TCP/IP?
27. Quanti e quali sono i livelli del TCP/IP?
28. Corrispondenza tra i livelli dell'OSI e quelli del TCP/IP
29. Quale ente ha standardizzato il modello OSI?
30. Fai alcuni esempi di protocolli di livello Application
31. Qual è il protocollo principale di livello 3 nell'architettura TCP/IP (livello Network)?
32. Quali sono i protocolli principali di livello 4 nell'architettura TCP/IP (livello Transport)?
33. Cosa significa che un sistema è aperto?
34. Cosa sono le RFC?