# **Definizioni e concetti chiave di RETI DI CALCOLATORI**

RETI – CHAPTER 1

* Una rete può essere vista da 3 punti di vista: **-Programmatore di applicazioni -Gestore di rete -Progettista di rete.**
* Un sistema che è progettato per consentire la propria crescita fino a dimensioni arbitrariamente grandi è detto **scalabile.**
* Possiamo **definire una rete** in maniera ricorsiva come  **un’entità formata da due o più calcolatori connessi tra loro attraverso un mezzo fisico oppure da due o più reti connesse tra loro da un nodo.**
* Per capire come diversi host possono condividere una rete dobbiamo capire il concetto di **multiplexing(multiplazione).**

Il multiplexing si può spiegare in analogia con la strategia del time-sharing usata nei calcolatori dove una singola CPU è usata da più processi, ciascuno dei quali crede di essere l’unico utente del processore.

In modo simile i dati inviati da più utenti all’interno di una rete possono usare la stessa strategia per condividere la stessa connessione fisica che costituisca una rete.

* Per realizzare il multiplexing di più flussi su un unico collegamento fisico esistono diversi modi: uno dei più comuni è il **multiplexing sincrono a divisione di tempo(STDM).** L’idea di base di STDM è quello di dividere i dati in intervalli aventi tutti la stessa durata e assegnare a ciascun flusso, a rotazione, la possibilità di inviare dati sulla connessione fisica.
* Un altro modo per realizzare il multiplexing di più flussi su un unico collegamento, è il **multiplexing a divisione di frequenza(FDM),** che è basato sull’idea di trasmettere ciascun flusso sulla linea di connessione fisica usando diverse frequenze(un po’ come si fa per la televisione).
* **I limiti di FDM e STDM:**

1. Se uno dei flussi(cioè una coppia di host) non ha dati da inviare la sua porzione del mezzo fisico rimane inutilizzata
2. Sono limitati a situazioni in cui il massimo numero di flussi è fisso e noto a priori

* **La modalità di multiplexing che useremo più spesso è chiamato multiplexing statistico.** Prevede la condivisione del mezzo fisico nel tempo, vale a dire prima vengono trasmessi sul mezzo fisico i dati di un flusso, poi quelli di un altro flusso e cosi via come in STDM, però diversamente da STDM i dati vengono trasmessi da ciascun flusso a richiesta, piuttosto che durante un intervallo di tempo prefissato.

Per evitare che una volta che il flusso sia iniziato non finisca più, il multiplexing statico definisce un limite superiore per la dimensione del blocco dati che ciascun flusso può trasmettere **(pacchetto)**.

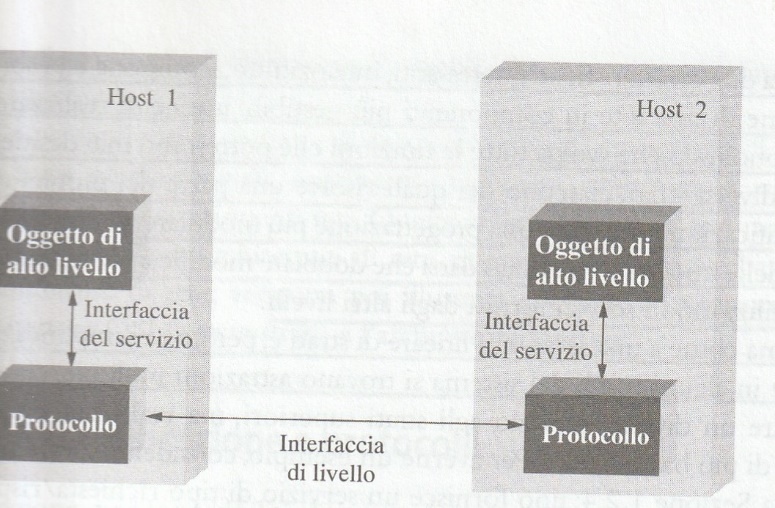
* Considerare una rete come uno strumento dedicato alla semplice consegna di pacchetti sarebbe una visione oltremodo semplicistica. Più corretto sarebbe pensare alla rete come uno strumento che fornisce a più processi applicativi distribuiti su più calcolatori un modo per comunicare.
* Per capire quando una **rete**  si può considerare **affidabile,** cerchiamo prima di capire quali possono essere i **malfunzionamenti.** Esistono 3 classi generiche di malfunzionamenti:

1. **Errori di bit** causati da forze esterne.
2. **Errori a livello di pacchetti**  per esempio se uno switch è pieno, elimina il prossimo pacchetto che gli arriva.
3. **Errori a livello di nodo e di linea di collegamento** per esempio se una linea fisica è interrotta

* **L’astrazione:** Cioè la possibilità di nascondere i dettagli realizzativi dietro un’interfaccia ben definita è lo strumento fondamentale utilizzato dai progettisti per gestire la complessità dei sistemi. Le astrazioni conducono in modo naturale alle **stratificazioni.** Si parte dai servizi offerti a più basso livello(hardware), per poi aggiungere una serie di livelli o strati, ciascuno dei quali fornisce un servizio di livello più elevato(cioè più astratto): **i servizi forniti ai livelli soprastanti sono realizzati utilizzando i servizi dei livelli sottostanti.**
* **La stratificazione ha due caratteristiche interessanti: innanzitutto, scompone il problema relativo alla costruzione di una rete in componenti più gestibili:** invece di realizzare un elemento software monolitico che svolga tutte le funzioni che potremo mai desiderare, si possono realizzare diversi strati, ciascuno dei quali risolve una parte del problema. **In secondo luogo, la stratificazione consente una progettazione più modulare:** se decidete che volete aggiungere qualche nuovo servizio, può darsi che dobbiate modificare le funzionalità di un solo livello, riutilizzando i servizi forniti dagli altri livelli.
* Gli oggetti astratti che compongono che compongono gli strati di un sistema di rete vengono chiamati **protocolli.** Un Protocollo **fornisce un servizio di comunicazione** che gli oggetti di livello superiore usano per scambiarsi messaggi
* **Ciascun protocollo definisce due diverse interfacce:**

1. **Interfaccia del servizio(service interface):** è rivolta agli altri oggetti che, sullo stesso calcolatore, vogliono usare i servizi di comunicazione del protocollo stesso.
2. **Interfaccia tra pari (peer interface):** rivolta alla propria controparte su un altro host.

* **Ciò che è detto sopra è sintetizzato in figura.**

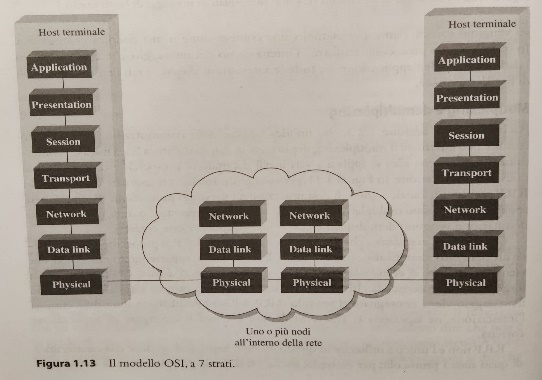
****

* **Un’intestazione (header)** è una piccola struttura di dati che viene usata da controparti di pari livello per comunicare fra loro. Le intestazioni vengono solitamente anteposte al messaggio.
* **IL Modello a 7 strati (pg.26)**

ISO è stata una delle prime organizzazioni a offrire formalmente una modalità comune per connettere calcolatori. La relativa architettura **OSI**(Open System Interconnection) definisce una divisione delle funzionalità di rete in 7 livelli o strati, con uno o più protocolli che realizzano la funzionalità assegnata ad un determinato livello.

Immagine che contiene tavolo

Descrizione generata automaticamente



Lo **strato di rete (network layer)** gestisce l’instradamento tra i nodi in una rete a commutazione di pacchetto; a questo livello il dato atomico scambiato tra i nodi è chiamato  **pacchetto.**

I tre livelli inferiori appena descritti sono solitamente presenti in tutti i nodi di una rete compresi gli switch all’interno della rete e gli host connessi alla periferica della rete stessa. Dallo stato di trasporto in su sono solitamente in esecuzione sugli host terminali e non sugli switch e router intermedi.

**Lo stato di trasporto (transport layer)** realizza ciò che abbiamo chiamato “canale tra processi”, qui il dato atomico scambiato viene chiamato **messaggio.**

**Lo strato di sessione (session layer)** fornisce uno “spazio per i nomi” che viene utilizzato per aggregare flussi di trasporto che possono costituire parti di un’unica applicazione.

**Lo strato di presentazione(presentation layer)** si occupa del formato dei dati scambiati tra le controparti.

**Lo strato di applicazione** consente ai browser web di richiedere pagine ai server web.

Partendo dal basso. Il primo strato è lo **strato fisico,** gestisce la trasmissione di flussi non strutturati di byte attraverso una linea di comunicazione.

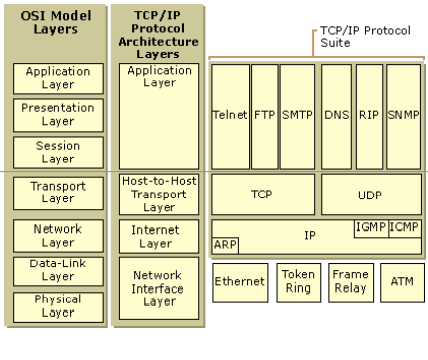
Lo **strato di linea (data link layer)** raccoglie un flusso di bit in un aggregato più grande denominato **frame**.

* **L’architettura Internet aka TCP/IP (pg 27)**

L’architettura Internet a volte viene chiamata anche architettura TCP/IP dal nome dei suoi due protocolli principali.

L’architettura Internet è più semplice della ISO/OSI.

* L’application Layer di TCP/IP copre i livelli Application, Presentation e Session di ISO/OSI
* Network interface layer copre Data Link and Physical

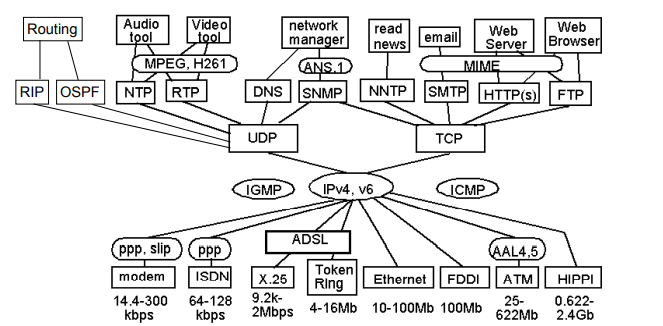


L’architettura Internet ha **3 caratteristiche che è bene sottolineare:**

1. Non implica una stratificazione rigorosa. L'applicazione può bypassare il livelli di trasporto definiti e di utilizzare direttamente IP o altre reti sottostanti.
2. Una forma a clessidra: larga in alto, stretta al centro e largo in basso.

L'IP funge da punto focale per l’architettura.

1. Affinché un nuovo protocollo sia ufficialmente incluso nell’architettura, deve essere fornita sia una definizione di protocollo e almeno una (preferibilmente due) implementazione di riferimento.



* La parte sottile della clessidra rappresenta un insieme minimale di capacità generiche, scelto accuratamente, che consente ad applicazioni di alto livello e a tecnologie di comunicazione a basso livello di coesistere. Tale modello è essenziale perché internet sia in grado di adattarsi rapidamente alle nuove richieste degli utenti e alle modifiche tecnologiche.
* **API (interfaccia per la programmazione di applicazioni)**

Il punto da cui è opportuno iniziare quando si parla di applicazioni di rete è l’interfaccia che viene resa disponibile dalla rete stessa. Poiché la maggior parte dei protocolli di rete è realizzata da software e quasi tutte le piattaforme di elaborazione realizzano i propri protocolli di rete come parte del proprio sistema operativo, quando parliamo di interfaccia resa disponibile dalla rete, stiamo solamente parlando dall’interfaccia resa disponibile dal proprio sistema operativo al proprio sottosistema di rete.

Questa interfaccia di rete viene comunemente chiamata API.

Anche se ciascun SO ha la libertà di definire le proprie API nel corso del tempo alcune di queste si sono molto diffuse, cioè sono state tradotte e implementate anche in SO diversi da quello di origine per esempio come è accaduto all’interfaccia di rete **socket** inizialmente creata per i sistemi Unix.

Ora le socket sono supportate da praticamente tutti i sistemi operativi.

Una socket può essere vista come il punto in cui un’applicazione locale comunica con la rete. L’interfaccia definisce le operazioni per: creare una socket, collegare una socket alla rete, ricevere e mandare messaggi attraverso la socket, chiudere la socket.

* **Prestazioni di una rete**

Le prestazione di una rete si misurano fondamentalmente in due modi: **ampiezza di banda(o throughput) e latenza.**

**L’ampiezza di banda** è il numeri di bit al secondo che possono essere trasmessi nel canale, **la latenza** è il tempo che trascorre tra l’invio del messaggio e la sua ricezione. Spesso si considera la latenza come composta da tre componenti:

**Latenza = Propagazione + Trasmissione + Accodamento**

**Propagazione = Distanza/velocità della luce**

**Trasmissione = Dimensione/Ampiezza di banda**

Dove distanza è la lunghezza della linea di connessione attraverso cui viaggiano i dati, velocità della luce è la velocità della luce in tale linea, dimensione è la dimensione del pacchetto e l’ampiezza di banda è l’ampiezza di banda con cui viene trasmesso il pacchetto.

* **Prodotto ritardo x Ampiezza di banda**

Dal punto di vista intuitivo, se pensiamo a un canale fra una coppia di processi come una conduttura, dove la latenza corrisponde alla lunghezza della conduttura e l’ampiezza di banda corrisponde al suo diametro, allora il prodotto ritardo x ampiezza di banda corrisponde al volume interno della conduttura, cioè il massimo numero di bit che possono trovarsi in transito in un determinato momento.

* **Jitter**

A volte il ritardo non è un problema ma la sua varianza (jitter) si.

Si pensi per esempio allo Streaming audio/video

RETI – CHAPTER 2