

Architettura di rete

Le reti di tlc sono sistemi molto complessi. La domanda è come posso progettare sistemi così complessi in maniera razionale ed efficace?

Questa complessità deriva dal fatto che una rete ha molte funzioni e si arriva a diversi livelli di astrazione.

Per risolvere questo problema si suddivide il problema in sottoproblemi. Quindi c'è un'esigenza di creare dei modelli funzionali, questo ha anche il vantaggio di consentire un'evoluzione progressiva, di dare flessibilità senza necessità ogni volta di riprogettare tutto quanto.

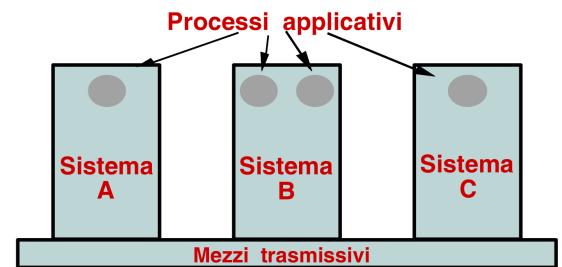
Elementi base dell'architettura

Primo elemento fisico sono i campi elettromagnetici che vengono distribuiti attraverso dei materiali come fibre ottiche, fili di rame, antenne ecc.

Quindi il **campo elettromagnetico** si propaga attraverso dei **mezzi trasmissivi**.

L'obiettivo è utilizzare i **processi applicativi** (come whatsapp, zoom ecc), il cui servizio si basa sull'interazione di diversi sistemi. Un processo applicativo funziona operando su sistemi tra loro remoti e per poter funzionare ha bisogno di comunicare con entità che realizzano il processo applicativo.

I sistemi sono un insieme di hardware e software, come ad esempio lo smartphone, che si connette ai mezzi trasmissivi (es. Wi-Fi) e ci permette di far utilizzare i processi applicativi. Un altro esempio di sistema è un router. Ci sono due categorie di sistemi: quelli che contengono i processi applicativi (end system) e quelli che smistano informazioni (router).



Definizione di strato

Strato più basso: ricevo bit

Salendo di livello consideriamo i bit divisi in pacchetti e controlliamo se questi contengono errori.

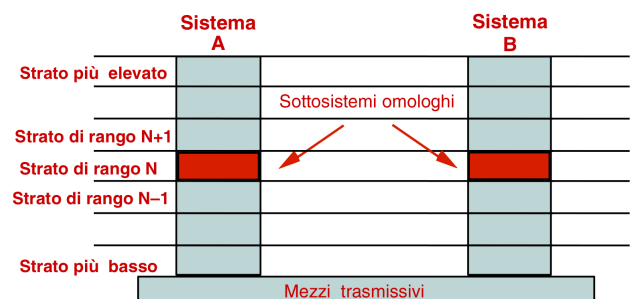
Salendo ancora di più troviamo la rete, quindi dobbiamo trovare un percorso di rete.

Salendo di livello astratto si arriva fino ad arrivare al processo applicativo.

L'insieme di sistemi hardware e software è un insieme di strati (layer).

Si chiama **strato** l'insieme di entità che svolgono la stessa funzione nel medesimo livello.

Per esempio trovare il percorso di rete si chiama *instradamento* e coloro che se ne occupano si trovano sullo stesso strato anche se si trovano in sistemi diversi.



Servizio di strato

Ogni strato di rango N fornisce un servizio allo strato di rango N+1. Per questo scopo utilizza il servizio fornito dallo strato di rango N-1. Arricchisce questo servizio con lo svolgimento di un particolare sottoinsieme delle proprie funzioni.

È comodo suddividere il soggetto in strati in quanto consentono di organizzare il lavoro di definizione e progettazione delle funzioni dei processi di comunicazione.

Modularità: è possibile modificare la realizzazione di una funzione di strato senza dover modificare anche gli altri strati.

Prezzo da pagare: soluzione sub-ottima (ma da un lato meglio perchè si può migliorare mano a mano)

Esempio: 25/02/2021

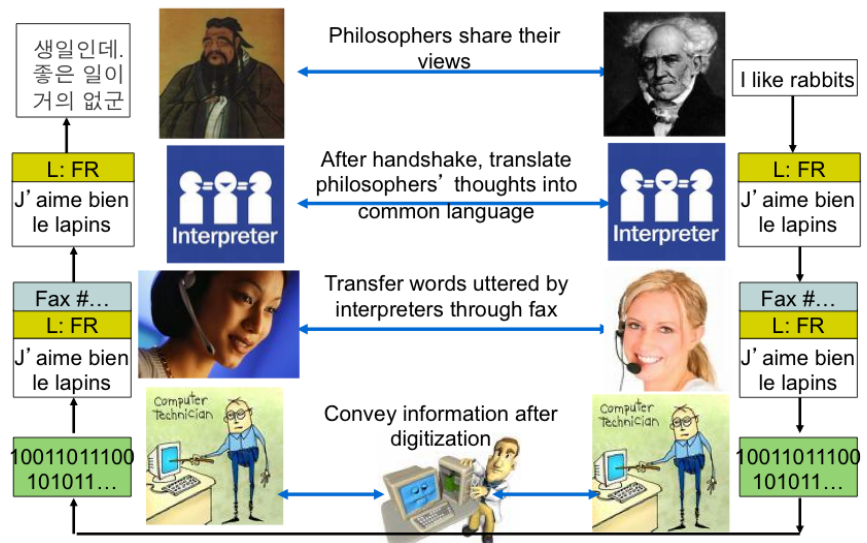
Inizialmente i due filosofi parlano una lingua diversa, lo strato successivo interpreta cosa loro dicono e lo traducono in una lingua comune.

Tuttavia ancora i due filosofi non possono comunicare tra loro e capirsi, allora nello strato successivo, quello

“dell’assistenza” ossia il corrispondente

dell’instradamento. Nell’ultimo strato le informazioni vengono

trasformati in bit e cominciano a saltare di router in router fino ad arrivare alla destinazione finale. (In realtà i bit vengono trasformati in segnali e sono i segnali a saltare). I bit passano poi verso l’alto nella colonna del filosofo arabo.

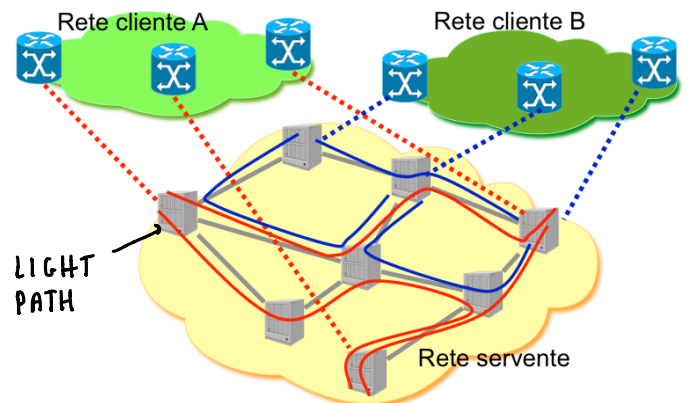


Telecomunicazioni - a.a. 2020/2021 - Prof. Andrea Baiocchi

La comunicazione è sempre alla pari ma avviene in verticale in modo inverso. L’unico tratto in cui c’è uno spostamento fisico è quello nel tratto orizzontale.

Un altro modo di vedere diversi livelli di astrazione .

Le due reti verdi rappresentano reti di internet service provider, questi dispongono di routers ai quali si attaccano i clienti. Tuttavia i routers devono essere connessi tra di loro: o crea una rete fisica propria con fibra ottica (molto costosa) o affitta capacità da una *transport network* (in giallo). Quest’ultima opzione consiste nell’affittare da qualcuno che ha già creato una rete fisica. Il collegamento tra i router avviene attraverso un percorso nella transport network, ad esempio la rete del cliente A ha il percorso rosso. Il percorso nella transport network non è affare della rete A, usufruisce solamente del servizio che ha affittato.



Stessa cosa B, chiede alla rete di trasporto di connettergli i routers.

La rete di trasporto con questi percorsi sta facendo un'operazione di moltiplicazione, in questo modo ottiene un'economia di scala. La rete di trasporto crea una grande scala che ammortizza i costi.

Quindi ad esempio in Italia ci sono all'incirca 10 reti telefoniche ma nessuno possiede una rete propria, ce n'è solo una che affitta capacità.

Momentaneamente c'è una rete fisica e poi più livelli di reti virtuali che si poggiano su di essa.

Internet protocol stack

Applicazione: elaborazione delle informazioni

Trasporto: trasferimento da estremo a estremo (host-to-host) dei messaggi applicativi

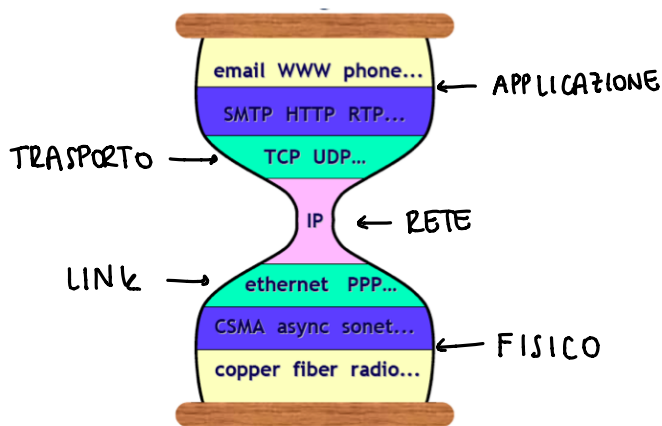
Rete: trasferimento dei pacchetti attraverso la rete (moltiplicazione, commutazione), smistare i pacchetti nelle loro diverse destinazioni. Ossia fare INSTRADAMENTO.

—> IP, ICMP, protocolli di instradamento. È fondamentale avere degli indirizzi.

Link(collegamento): trasferimento dei dati all'interno di una sottorete

Fisico : dai bit/byte ai segnali e viceversa, physical layer e data link layer. È tutto quello che riguarda le sottoreti

Hourglass model (H. Schulzrinne)



Queste funzioni vengono implementate tramite i protocolli, ci deve essere un protocollo comune che è quello della rete.

Protocolli

Un protocollo è un insieme di regole che governano come due o più individui possono comunicare. Definisce l'interazione tra entità.

Esempi: Internet Protocol (IP), Transmission Control Protocol (TCP), HTTP, SMTP

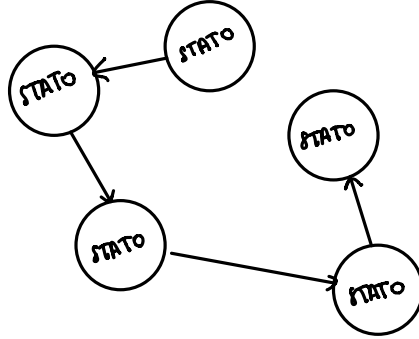
Elementi che costituiscono un protocollo: **sintassi** (specifica come costruire blocchi di dati interpretabili in quel protocollo), **semantica** (cosa fare con quello che ho ricevuto?) e la **temporizzazione** (gli scambi avvengono nel tempo e quest'ultimo è cruciale, l'interazione non deve andare in stallo).

Protocolli come macchine di stato

Un protocollo può essere descritto come macchina a stati finiti. Ossia immaginiamo di avere una variabile che dipende dallo stato in un dato momento, da quello stato posso decidermi se spostarmi in un altro stato o rimanere fermo, quindi facendo ciò creo degli eventi. Si determina un grafo dove i nodi sono gli stati che scatenano le transizioni e gli archi sono l'azione in sé.

Quindi la macchina a stati finiti può essere pensata come un grafo.

Uno stato è un insieme di variabili il cui valore è sufficiente per decidere la prossima transazione data in input



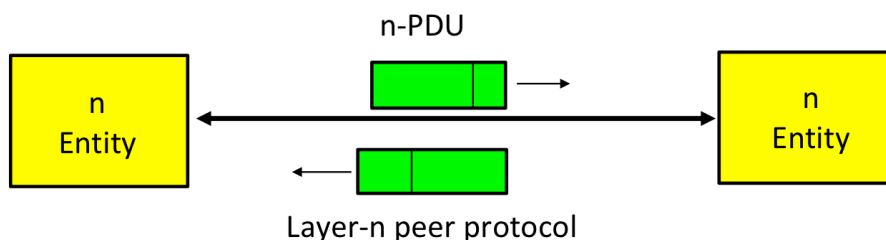
Collegiamo il concetto di stato a quello di protocollo. All'interno di ogni stato le entità interagiscono attraverso i protocolli.

Quando ci si riferisce ad una architettura a strati generali utilizziamo la terminologia: n-strati, n-entità, n-protocollo (stato è la stessa cosa di layer)

L'n-protocollo consiste in una macchina a stati e si svolge tra le n-entità mediante blocchi di dati che prendono il nome di protocol data unit (PDU)

Concetto astratto di protocollo

- Le entità che eseguono le funzioni di uno strato all'interno di sistemi comunicanti sono dette **peer entities**
- La cooperazione tra entità dello stesso strato è regolata dal protocollo di strato n (layer-n protocol)
- Le entità di strato n (layer-n peer processes) si scambiano unità di dati denominate *Protocol Data Unit (PDU)*



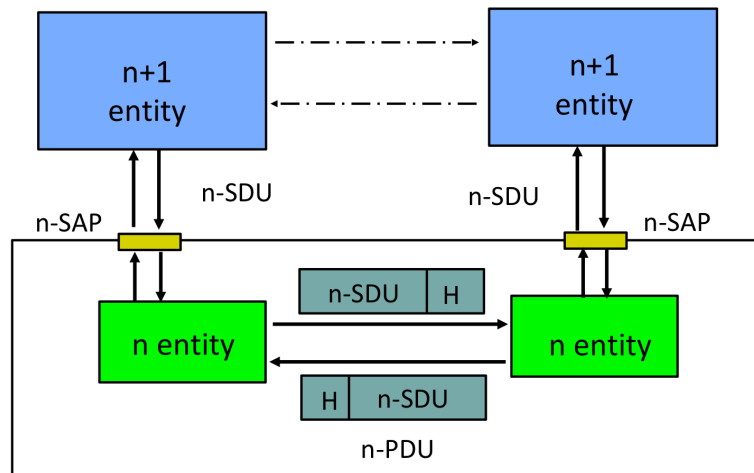
Concetto astratto di servizio

- La comunicazione tra due entità è virtuale ed indiretta
- Uno strato trasferisce le proprie informazioni invocando il servizio fornito allo strato inferiore ($n+1 \rightarrow n$)
- I servizi di uno strato sono disponibili all'interfaccia con lo strato superiore (*service access points- SAP*)
- Ogni strato passa dati e informazioni di controllo allo strato inferiore fino a che è raggiunto lo strato fisico che si occupa dell'effettivo trasferimento
- I dati che lo strato n riceve dallo strato $n+1$ sono denominati *n-Service Data Unit (n-SDU)*
- Le SDU sono incapsulate nelle PDU nelle quali sono anche aggiunte le informazioni di controllo per l'esecuzione delle funzioni di strato (PCI, Protocol Control Information)

Interfacce

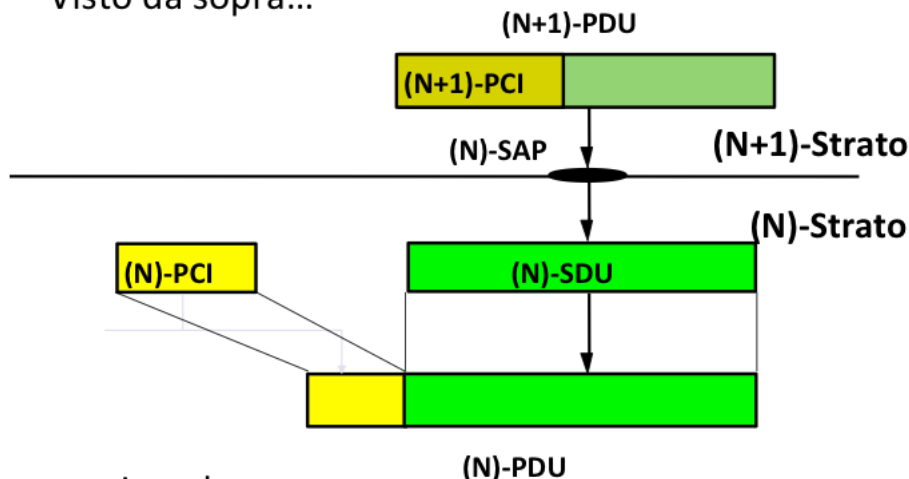
Possiamo identificare due interazioni:

- **Protocolli** —> interfacce orizzontali, tra pari. Entità di uno stesso stato interagiscono attraverso protocolli, inviando e ricevendo n -PDU (tra entità remote in sistemi diversi (in router diversi))
- **SAP** —> interfacce verticali, tra entità di strato cliente ed entità di strato servente (tra entità di strati diversi sempre localmente (nello stesso router))



In questa catena ci si ferma quando si arriva al livello uno.

Visto da sopra...



...e visto da sotto.

Una PDU è sempre costituita da : un'informazione di controllo (in giallo) e il **payload** (in verde).

È come un blocco di bit, la prima parte un'informazione di controllo che spesso sta all'inizio ma non necessariamente, e la seconda parte è il carico. La sintassi del protocollo ci permette di capire dove sta il controllo e dove il carico. Questo compito di dividere in blocchi si chiama *parsing*. Un protocollo deve fare parsing del blocco di dati.

La parte obbligatoria è l'informazione di controllo, nella PDU ci deve essere sempre. Non sempre è presente il payload, il carico pagante c'è solo se c'è qualcosa da dire.

Visto da sopra l' n strato riceve un blocco di bit da $n+1$ layer senza sapere cosa ricevere e si prende l'impegno di trasportare, lo fa tramite la PDU. Ogni volta che uno strato n produce una PDU questa ha un payload e le informazioni che sono i dati che riceve da sopra.

Tuttavia l' n strato non sa quello che ha ricevuto, è come se avesse un blocco piatto.

Il vantaggio di ciò è che se un giorno si volesse cambiare il protocollo $n+1$ non c'è bisogno di cambiare tutto, in quanto gli altri strati trasportano e basta e non sanno che cambia qualcosa.