Computer Networks
Chapter 1 - Foundations

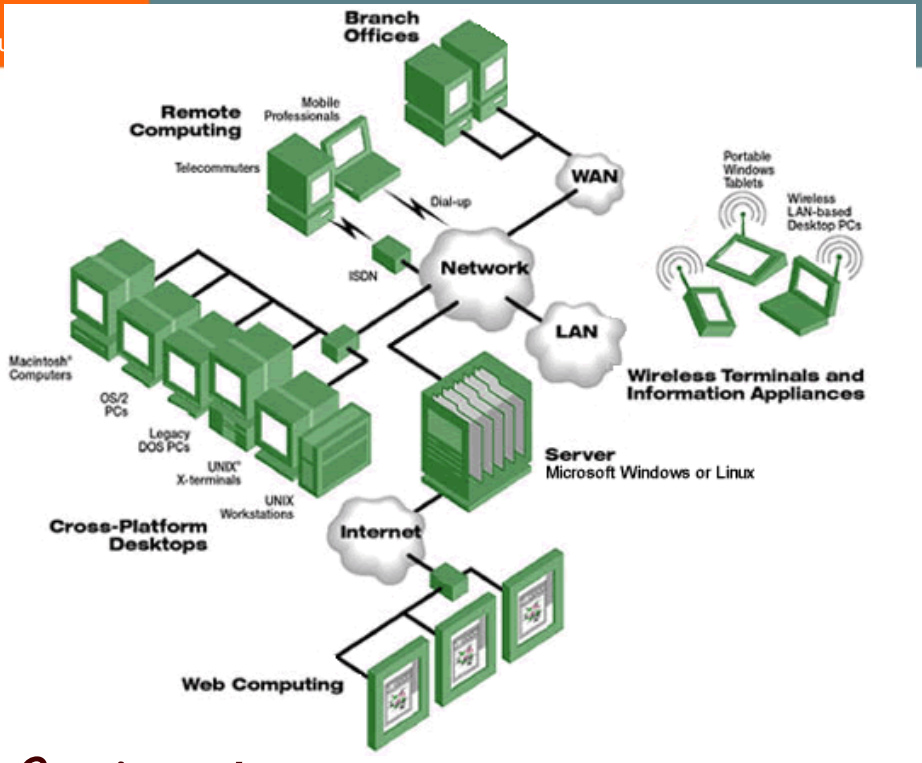
Prof. Marino Miculan

Prologue

What?

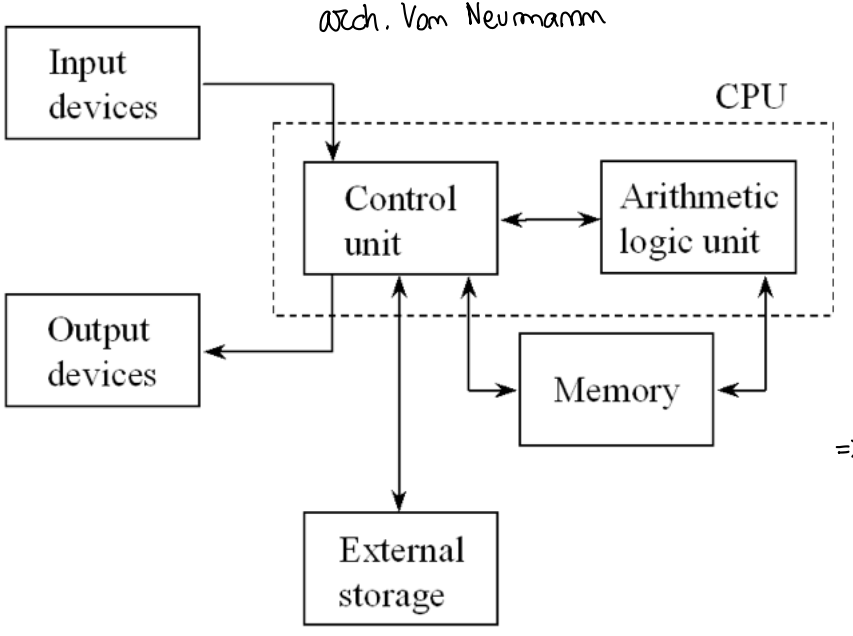
- Suppose we want to build, or choose, or manage, a computer network
  - Who is going to use this network?
  - Why? which are the applications?
  - Which architecture?
  - What hardware/software?

- What is a **(computer) network**?
  - How is a computer network different from other types of networks?
- What is a **computer network architecture**?



Ceci est un reseau.

Many different independent nodes, each with its own computing power



=> non è una RETE perché i singoli nodi non sono indipendenti (nei vecchi sistemi)

Ceci n'est pas un reseau.

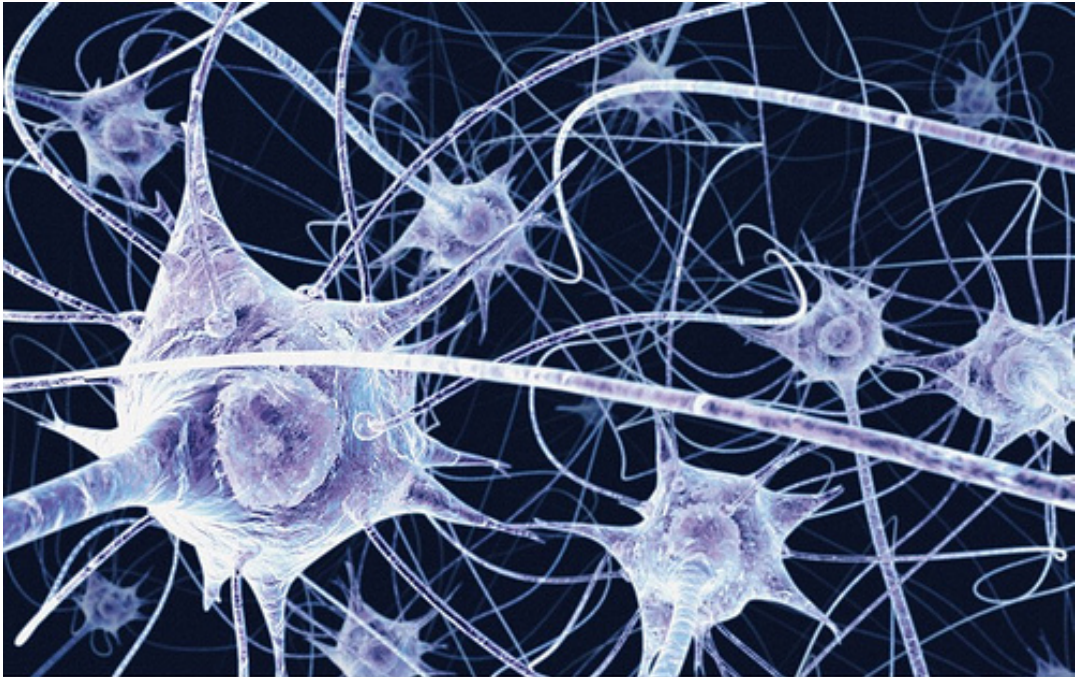
One single "control" node, the others have no computing power





Ceci est un reseau.

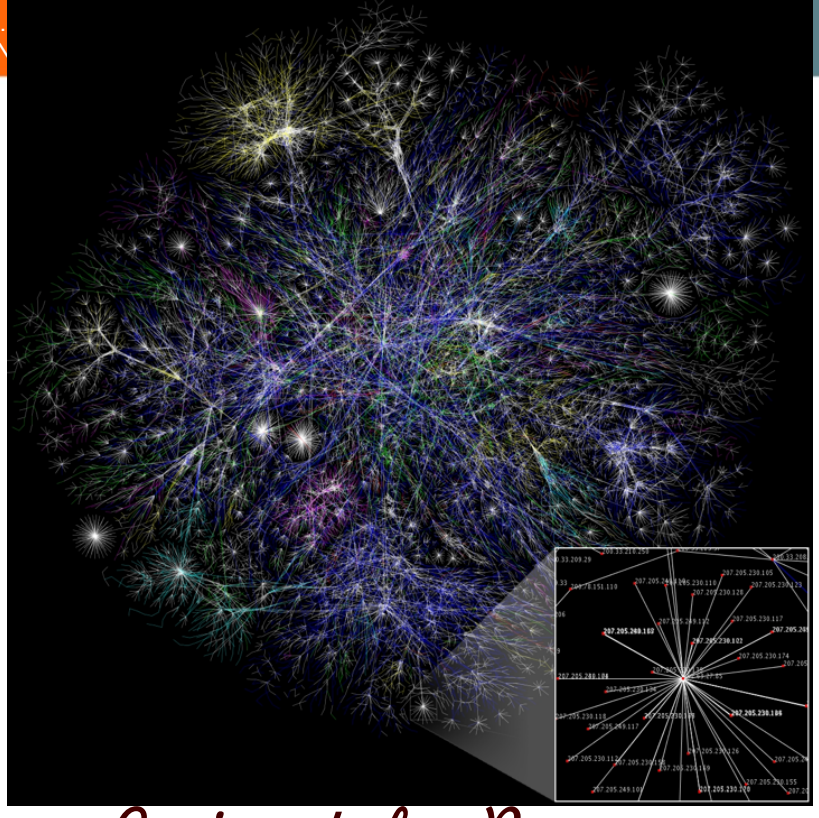
Dynamically established connections, via intermediary specialized nodes and channels



⇒ è una RETE perché ogni neurone ha la sua funzione e collettivamente lavoriamo

Ceci est un reseau.

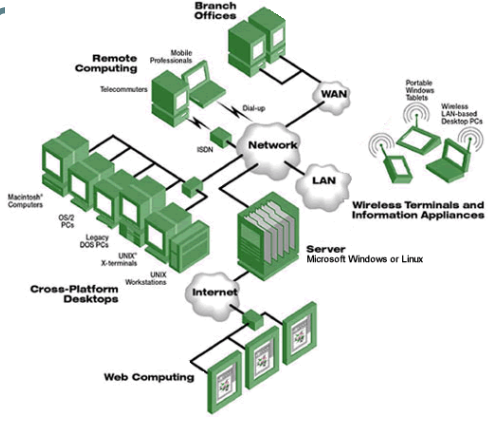
Self-configuring, self-repairing, resilient network.

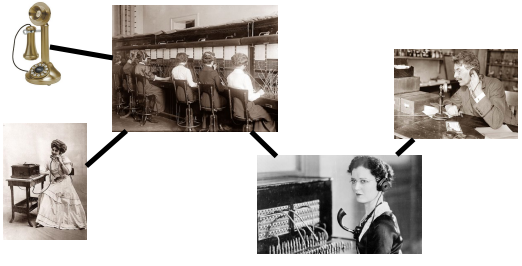


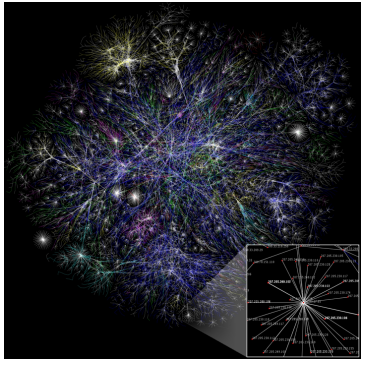
Ceci est le Reseau.

Picture of the Internet (not even complete)

Comm

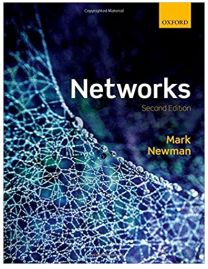

Branch Offices
Remote Computing
Mobile Professionals
Telecommuters
WAN
Dial-up
ISDN
Network
LAN
Portable Windows Tablets
Wireless LAN-based Desktop PCs
Wireless Terminals and Information Appliances
Macintosh® Computers
OS/2 PCs
Legacy DOS PCs
UNIX® X-terminals
UNIX Workstations
Server
Microsoft Windows or Linux
Cross-Platform Desktops
Internet
Web Computing





Common features of networks

- a network is composed by
  - (independent) **nodes** che si scambiano informaz. usando canali di comunicaz.
  - exchanging **informations**
  - using **communication channels**
  - in order to perform a **task** which cannot be done by a single node
- nodes can be computers, people, cells...
- channels can be electrical, optical, chemical, virtual...
- Great book for general networks:
  Mark Newman, "Networks", 2018, OUP



Specific features of *computer* networks

- General purpose
  - not optimized for a specific application (Un esempio di rete per singolo scopo è la rete telefonica)
  - many applications can coexist → una volta con la rete telefonica o chiamavi o navigavi in Internet
- Openness, flexibility
  - new features can be added dynamically (even not devised at the moment of network design / implementation)
  - vendor-independent

RFC insiemi di standard

In this course: general issues and solutions for computer networks

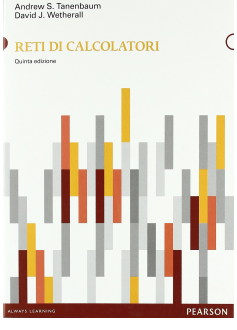
- **requirements** about the services implemented by a network
  - What the applications are expecting from a network?
- **general principles** for implementing, managing and improving these services
  - some theory of signals
  - network architectures
  - protocols
  - some network security
- **main applications and examples**
  - ISO/OSI model, TCP/IPv4 and IPv6 stacks
  - some applications

What we will not see here

- Specific issues of specific applications
- Most application-level protocols and algorithms
- Suggested courses for continuing on these topics:
  - Distributed Systems
  - Semantica e Teoria della Concorrenza
  - Computer Network Security
  - (Tecnologie web per il cloud, IBML second year)

References

- Suggested book:
  - L. Peterson, B. Davie, “Reti di Calcolatori”, 3ª ed., Apogeo, 2012
- Alternatives:
  - A. Tanenbaum, D. Wetherall, “Reti di Calcolatori”, 5ª ed., Pearson, 2011
- But also: Kurose, Halsall, Stallings…
- Slides, examples, old exams, general informations: elearning.uniud.it



Modalità d'esame

- L'esame si compone di una prova scritta e di una eventuale prova orale.
- La prova scritta richiede di svolgere alcuni esercizi inerenti gli argomenti del Corso e rispondere ad alcune domande teoriche.
- La prova orale è obbligatoria solo per discriminare le "situazioni di confine":
  - $17 \leq$ voto scritto ≤ 19 oppure voto scritto ≥ 28.
  - Con voto scritto < 17, non si è ammessi all'orale.
  - ~~Con voto scritto < 10 non ci si può iscrivere all'appello successivo (anche tra sessioni differenti).~~
  - L'orale consiste in due-tre domande e può variare il voto dello scritto al massimo di un paio di punti (in più, ma anche in meno).

Chapter 1
Foundations

Chapter Outline

- In order to understand the design, the implementation and the use of networks, we will see:
  - Example of applications
  - Requirements
  - Network Architecture
  - Implementing Network Software
  - Performance Evaluation

Chapter Goal

- Exploring the requirements that different applications and different communities place on the computer network
  - users
  - application programmers
  - network operators
  - network and protocol designers
- Introducing the idea of *network architecture*
- Introducing some key elements in implementing Network Software
- Define key metrics that will be used to evaluate the performance of computer network

Applications and Requirements of Computer Networks

Applications

- Most people know about the Internet (a computer network) through applications
  - World Wide Web
  - Email
  - Online Social Network
  - Streaming Audio Video
  - File Sharing
  - Instant Messaging
  - …

Example application: WWW

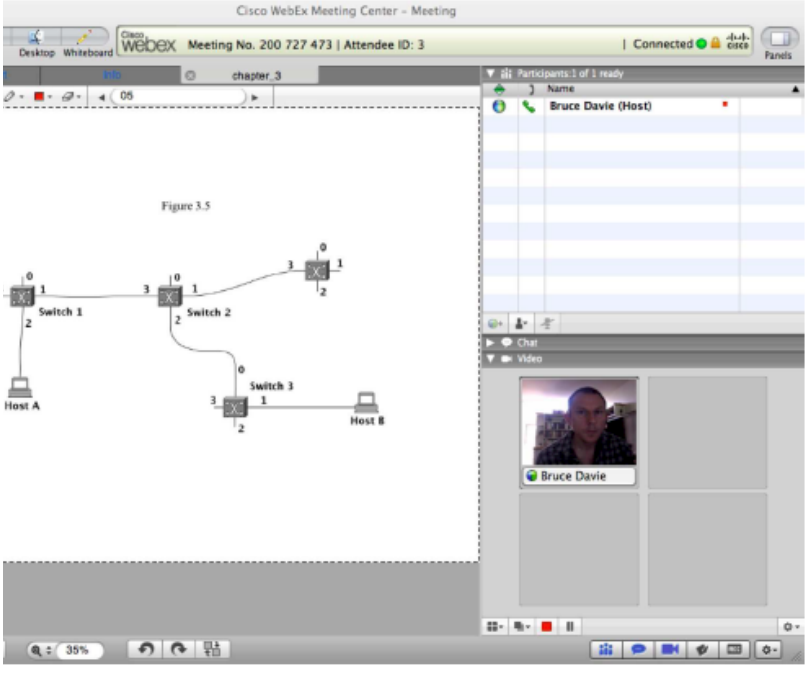
IPERTESTO: insieme di documenti messi in relazione tra loro per mezzo di parole chiave. È una rete con i singoli documenti considerati come nodi della rete

- Main application of Internet for most users
- Hypertext of (possibly dynamically generated) documents, distributed among many servers
- URL: Uniform resource locater
  - http://www.dimi.uniud.it/miculan
- HTTP: Hyper Text Transfer Protocol → protocollo in cui client (browser utente) e server (sito) creano una connessione scambiandosi messaggi
- TCP: Transmission Control Protocol => molti protocolli moderni (applicativi) si appoggiano su TCP.
  Ricorda che un protocollo si appoggia sui protocolli di livello inferiore → in realtà TCP non è implementato fisicamente, è realizzato con il protocollo (sottostante) IP.
- 17 messages for one URL request
  - 6 to find the IP (Internet Protocol) address
  - 3 for connection establishment of TCP
  - 4 for HTTP request and acknowledgement
    - Request: “I got your request and I will send the data”
    - Reply: “Here is the data you requested; I got the data”
  - 4 messages for tearing down TCP connection

Example application: video conferencing

- A multimedia application including video-conferencing.
- Requires audio-video streaming, hypertext, coordination, etc.

Nonostante sia un contesto diverso rispetto a quello di prima, utilizza sempre gli stessi protocolli (TCP/UDP - IP)



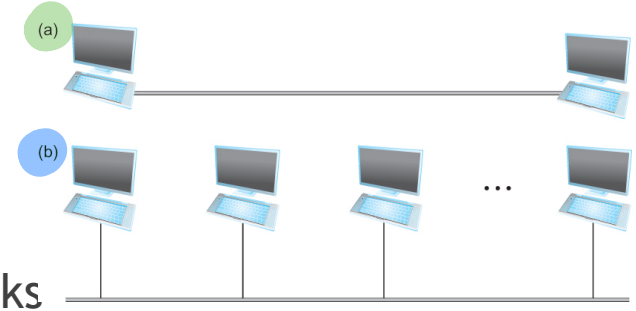
Requirements

- Important to understand general concepts and basic principles, rather than specific cases
- Different points of view:
  - **Application Programmer**
    - List the services that his application needs: delay bounded delivery of data, mobility, throughput…
  - **Network Designer**
    - Design a cost-effective network with sharable resources, efficient, fair for applications and users…
  - **Network Provider**
    - List the characteristics of a system that is easy to manage

i ritardi sono colpa di bottiglia

Connectivity

- Need to understand the following terminologies
  - Designed to scale!
  - Link
  - Nodes
  - Point-to-point
  - Multiple access
  - But directed connected networks would be either very limited or very expensive (or both)



(a) Point-to-point → es. fibra ottica
(b) Multiple access
↳ es. aula, aria

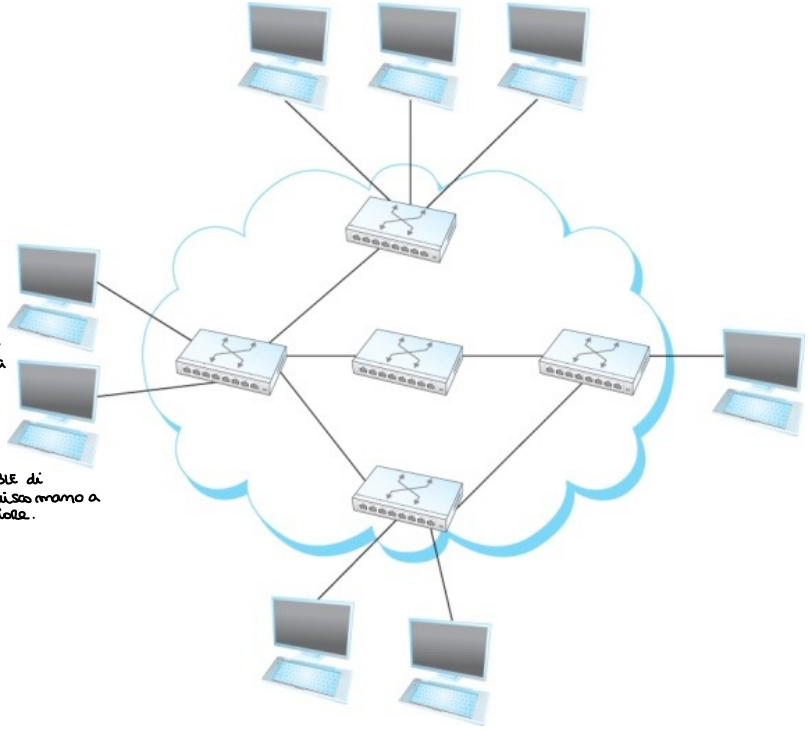
Connectivity

- Terminology (cont'd)
  - Switched Network
    - Specialized nodes allow for connecting different network trunks
    - Circuit Switched
    - Packet Switched
  - Packet, message
  - Store-and-forward

COMMUTAZ. di CIRCUITO
costruisco prima il circuito e poi spedisco
↗ ROUTING STATICO, se salta un nodo salta tutto

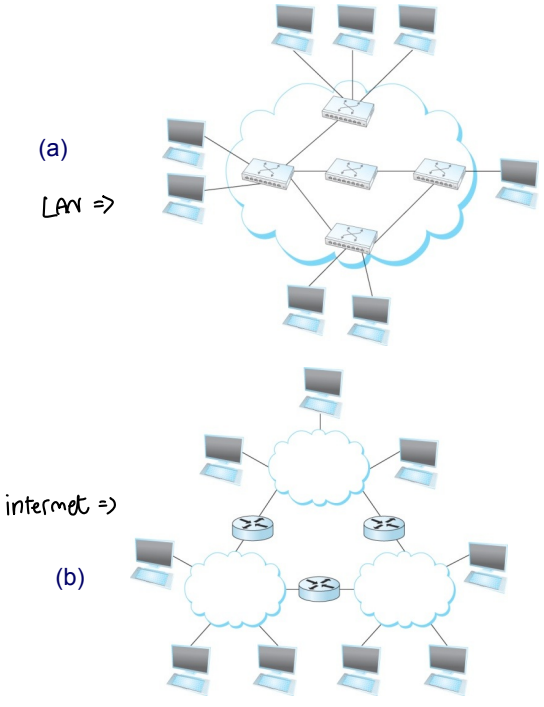
COMMUTAZ. di PACCHETTO
tramite le ROUTING TABLE di ciascun Router e costruisco nodo a nodo il percorso migliore.
ROUTING DINAMICO

↳ ricevo, decido ed eventualmente inoltro



Connectivity

- Terminologies (contd.)
  - Cloud
  - Hosts
  - Switches
  - internetwork — connessione di 2 o più reti, rete di rete
- Internet is an internet — con la i minuscola abbreviazione di internetwork
  - Router/gateway
  - Host-to-host connectivity
  - Address
  - Routing
  - Unicast/broadcast/multicast
    - unicast: UN SOLO DESTINATARIO
    - broadcast: A TUTTI
    - multicast: UN GRUPPO DI DESTINATARI



(a) A switched network
(b) Interconnection of networks

A recursive definition of network

- A (computer) network is an entity formed by
  - two or more nodes connected by a medium
  - or two or more networks connected by a node.

se sono 2 modi è PUNTO PUNTO, sennò è mezzo condiviso

- **A network is a hierarchy of networks** (often very different).

- Leads to the problem of addressing nodes in different networks, in order to allow message routing and delivery

Cost-Effective Resource Sharing

- Important for the network designer and provider
- Resource: links and nodes
- How to share a link?
  - Multiplexing → Uso un canale per condividere più segnali. Ne esistono di 3 tipi: - TIME DIVISION - FREQUENCY DIVISION - STATISTICAL
  - De-multiplexing

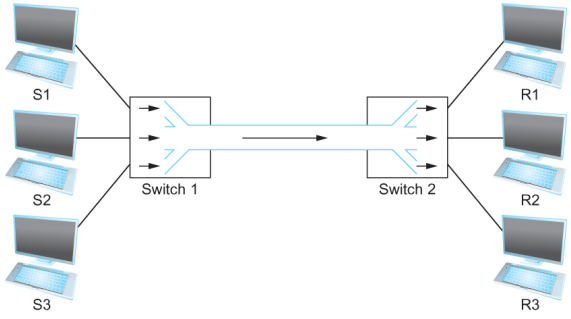
è scomodo ad es. aprire un canale singolo per ogni coppia di processi che si devono scambiare messaggi

prendo più segnali e li trasmetto su un canale

CI SONO 3 TIPI DI MULTIPLEXING:

1. Synchronous Time-division Multiplexing → es. ROUND ROBIN, BLUETOOTH → divido il tempo per ciascun host, quindi il cavo fa "passare" tot secondi di segnale a ciascun host
   - Time slots/data transmitted in predetermined slots → può essere non time effective
2. FDM: Frequency Division Multiplexing → dividiamo in canali es. la radio divido in base alla frequenza

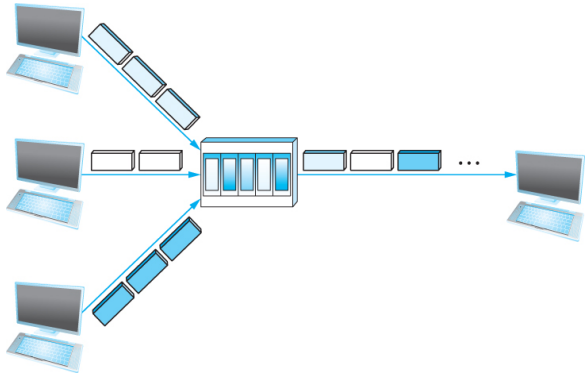
- Both have limits in efficiency and scalability



Multiplexing multiple logical flows over a single physical link

Cost-Effective Resource Sharing

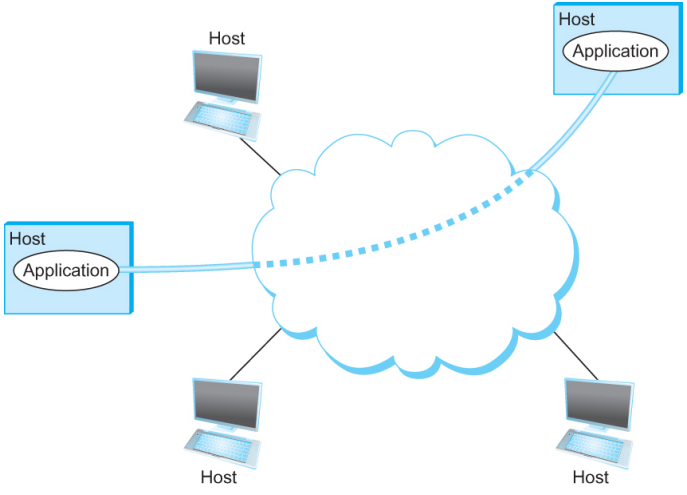
- Statistical Multiplexing
  - Data is transmitted based on demand of each flow.
  - What is a flow?
  - Packets vs. Messages
  - FIFO, Round-Robin, Priorities (Quality-of-Service (QoS))
  - Congested?
- LAN, MAN, WAN
- SAN (System Area Networks)



A switch multiplexing packets from multiple sources onto one shared link

Support for Common Services

- Programmers do not want to deal with single packets. They need *logical channels* between applications: application-to-application communication paths or pipes
- The network can be seen as a provider of logical channels, which can be established and maintained on the need
- Next problem is to define the functionalities for these logical channels
  - e.g.: reliable? Message size? Connection or connectionless?



Process communicating over an abstract channel

Common Communication Patterns

- By observing many applications, we can identify common communication patterns
- Client/Server applications are common
  - eg. File transfers/access, name resolutions…
  - But others are possible
- Two basic types of communication channel
  - Request/Reply Channels → scambio pacchetti FINITI, es. messaggi, che arrivino tutti certi (es. TCP)
    - Suitable for many client/server interactions
    - Can be reliable or not
  - Message Stream Channels → scambio pacchetti FINITI e il flusso è CONTINUO, l'importante è che arrivino più pacchetti possibili (UDP è un esempio)
    - Used often for video on demand, video conferences,…
  - But other types are possible

↳ es. del client server del prof:
- CONNECTIONLESS (UDP) ogni volta che il client mandava un messaggio al server era come se si collegasse
- CONNECTED (TCP) si collegava all'inizio il client e poi la connessione per i messaggi era stabilita

es. del postino che per mandare una lettera la consegna al destinatario e questo per rispondere salva l'indirizzo del suo amico e rispedisce al postino la risposta

al posto che il postino che mi fa da tramite, ho un tubo che mi collega direttamente con l'amico e comunichi da lì

Reliability affidabilità

Network should hide the errors

- Bits are lost
  - Bit errors (1 to a 0, and vice versa)
  - Burst errors – several consecutive errors
- Packets are lost (Congestion)
- Links and Node failures
- Messages are delayed
- Messages are delivered out-of-order → arrivano fuori ordine
- Third parties eavesdrop
  ↳ SNIFFING

Management

- Network admins have to manage (configure, monitor, repair) networks easily
- Configuration of large networks cannot be done by a single admin
- Sometimes, networks do not have expert administrators (e.g. home networks, ad hoc networks)
- Networks should be auto configurable, and resilient to failures

↳ si adatta ai cambiamenti

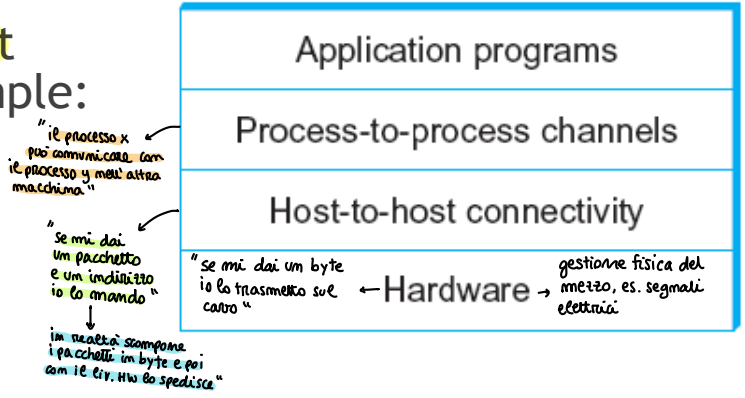
Network Architecture and Protocols

Network Architecture

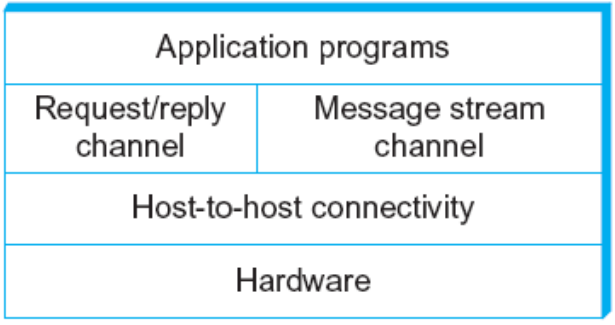
tutte le reti sono fatte a strati, le studiamo aggiungendo dei layer sopra lo strato HW

- Networks are complex
- It is difficult to implement all their functions in a single piece of code / hardware
- It is easier to adopt a *layered* architecture (like in layered operating systems)
- Each layer builds on that below it adding new functionalities. Example:

ogni strato implementa nuove funzionalità astratte utilizzando le funzionalità (API) dello strato sottostante



Network Architecture

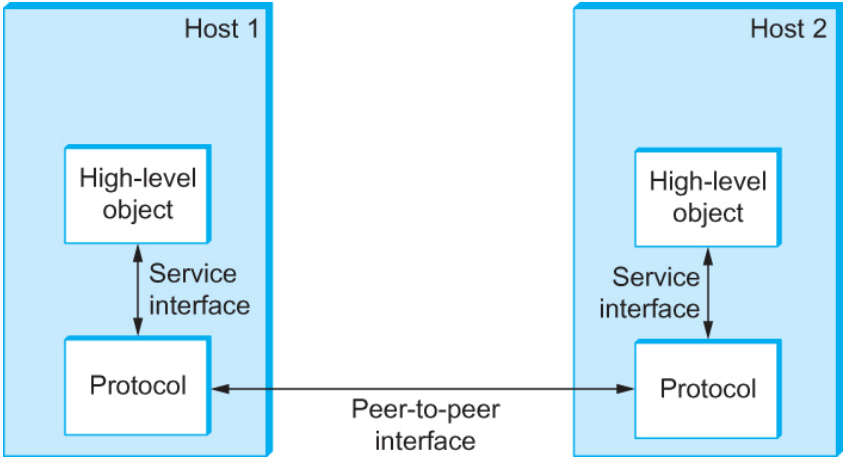


Layered system with alternative abstractions available at a given layer

Interfaces

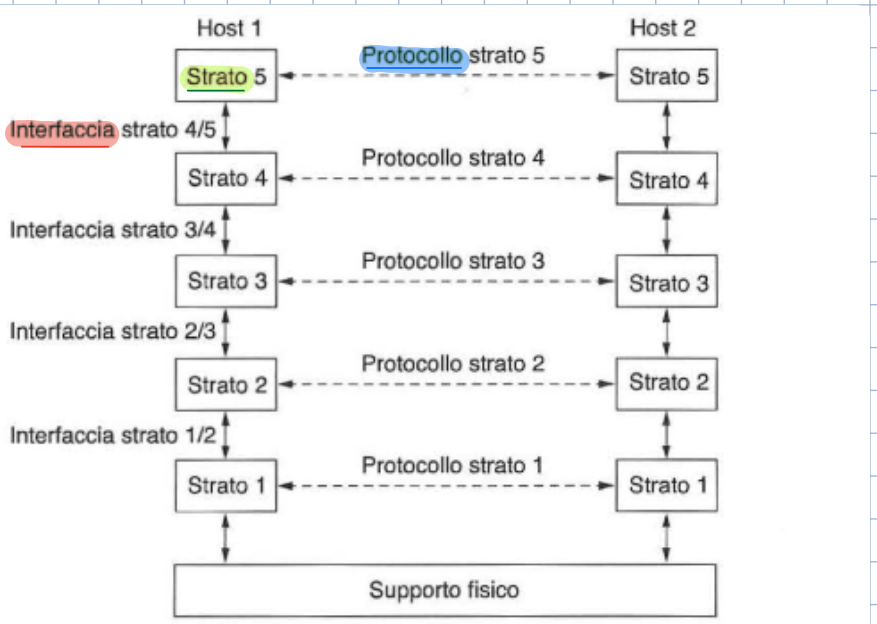
SERVIZIO: API offerte dal modulo ai livelli superiori

PEER TO PEER: il protocollo richiede dei servizi dallo stesso protocollo dall'altro host



Service and Peer Interfaces

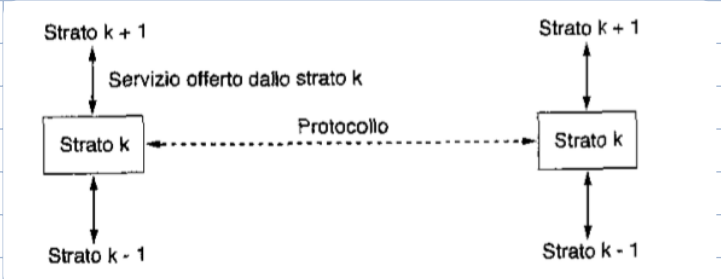
le implementaz. delle due macchine possono essere diverse (es. 32 vs 64 bit, o s.o. diversi) ma devono poter comunicare



- lo STRATO è il LIVELLO.

- l' INTERFACCIA rappresentata dal collegamento VERTICALE tra STRATI diversi, definisce le operazioni elementari ed i servizi che lo strato inferiore rende disponibili a quelli superiori. Un servizio è un insieme di primitive (operazioni) che uno strato offre a quello superiore senza spiegare come sono implementate (le operazioni), quello lo fa il protocollo

- i PROTOCOLLI funzionano tra strati dello stesso livello, sono degli accordi tra strati dello stesso livello sul modo in cui deve procedere la comunicazione, ovvero le entità usano i protocolli per implementare le definizioni dei loro servizi.



Gli strati possono offrire due tipi di servizi a quelli sopratanti

- CONNECTION-ORIENTED: si costruisce il canale della comunicazione (tubo) e si scelgono i parametri della comunicaz. Fatto ciò poi un utente carica i bit nel "tubo" e c'è chi li scarica. I pacchetti caramente arrivano disordinati
- CONNECTION LESS: come il postino, ogni messaggio ha l'indirizzo completo del destinatario ed è instradato verso di esso → possono arrivare i pacchetti in ordine diverso (DATAGRAM)

ogni volta che invii un messaggio con UDP devi dichiarare indirizzo del destinatario e porta
con TCP invece lo fai solo alla creazione della connessione

Spesso si preferisce CONNECTIONLESS perchè CONNESSO non è disponibile, ad es. Ethernet è NON CONNESSO perchè i pacchetti possono corrompersi ogni tanto, se ne devono occupare gli strati superiori.

Inoltre la creazione del "tubo connesso" può richiedere molto tempo

Protocols

↳ dice come vengono implementate le funzionalità

Un insieme di regole che definiscano i servizi offerti ai liv. superiori (VERTICALE) e ai livelli uguali (ORIZZONTALE) nelle altre macchine

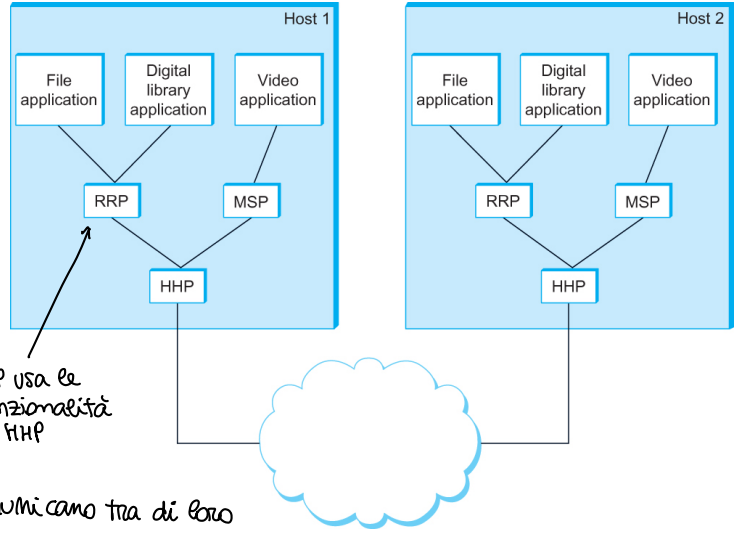
- A **protocol** defines the interfaces between the *layers* in the same system and with the layers of *peer* system
  ↳ ORIZZONTALE
- Building blocks of a network architecture
- Each protocol object has two different interfaces
  - service interface: operations on this protocol → VERTICALI
  - peer-to-peer interface: messages exchanged with peer → ORIZZONTALI
- Term "protocol" is overloaded
  - specification of peer-to-peer interface
  - module that implements this interface

Protocols

- Protocol Specification: prose, pseudo-code, state transition diagram
- Interoperable: when two or more protocols that implement the specification accurately → posso modificare/creare nuovi protocolli basta che mi attenga a delle specifiche e tutti gli strati sopra funzioneranno
- IETF: Internet Engineering Task Force: defines protocols
  - ↳ rilascia gli RFC: standard per i protocolli

Protocol Graph → dipendenze tra protocolli

- Example of a protocol graph: nodes are the protocols and links the "depends-on" relation
- RRP: request-reply protocol
- MSP: message stream protocol
- HHP: host-to-host protocol

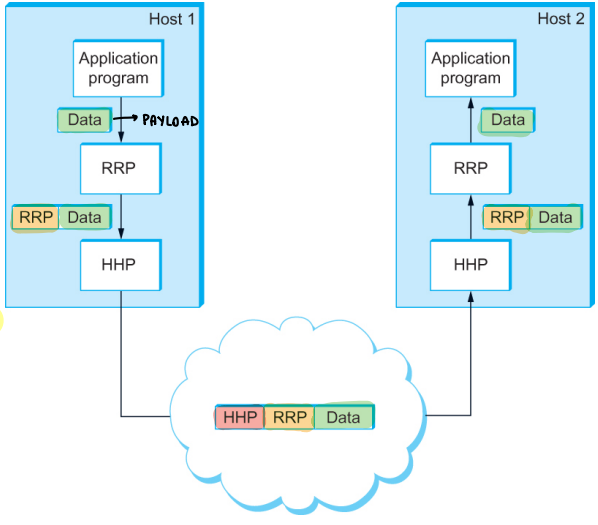


"virtualmente" anche, ad es., RAP e RRP comunicano tra di loro

Encapsulation

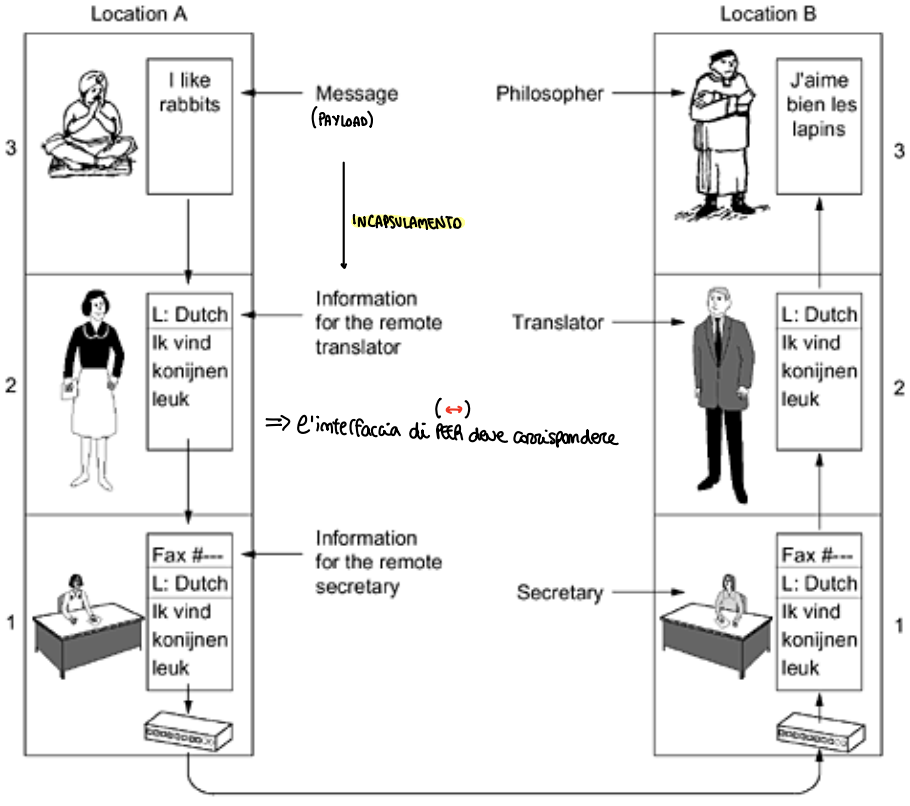
informazioni aggiunte (header) al file man mano che scende di livello

il PAYLOAD è il data che riceve lo strato (livello) x dal livello x+1



High-level messages are encapsulated inside of low-level messages

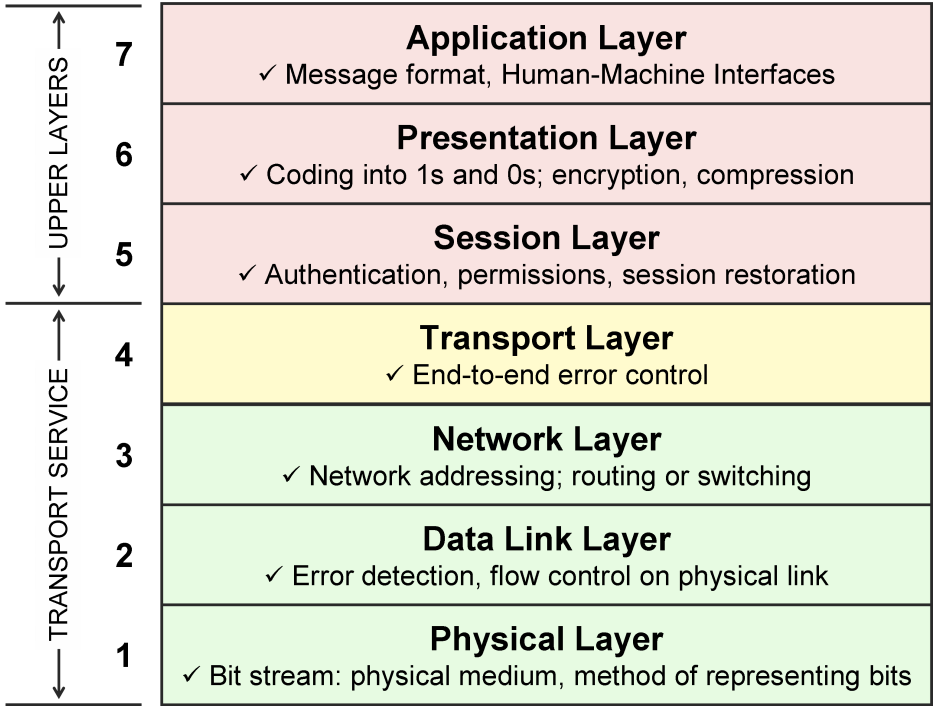
Example


Location A
Location B
I like rabbits
Message
(PAYLOAD)
Philosopher
J'aime bien les lapins
3
INCAPSULAMENTO
L: Dutch
Ik vind konijnen leuk
Information for the remote translator
Translator
2
⇒ l'interfaccia di PEER deve corrispondere (↔)
Fax #---
L: Dutch
Ik vind konijnen leuk
Information for the remote secretary
Secretary
1

The ISO OSI Architecture

OSI = Open Systems Interconnection

1, 2, 3, 4 → tutti protocolli già implementati nel S.O. (Kernel)
5, 6, 7 → implementati a livello applicativo



Description of Lower Layers

l'indirizzo MAC ci dice che interfaccia utilizza

1. **Physical Layer** → implementato nella scheda di rete; tutto ciò che riguarda i mezzi fisici, come i bit sono codificati nel segnale; "se mi dai dei byte io li faccio arrivare dall'altra parte"
   - Handles the transmission of raw **bits** over a communication link
   - Here we deal with encoding/decoding issues

1. FRAME → 0101→

2. **Data Link Layer** → gestire l'accesso al mezzo e controllare eventuali errori
   - **Collects a stream of bits into a larger aggregate called a frame**
   - Network adaptor along with device driver in OS implement the protocol in this layer → il livello 2 garantisce che il messaggio se arriva sbagliato me ne accorgo
   - Frames are actually delivered from host to hosts → c'è anche un meccanismo per regolare il traffico nel caso di trasmettitore veloce e ricevitore lento

2. PACCHETTO → FRAME

messaggio che viaggia da interfaccia a interfaccia

↳ se ad es. stacco il cavo mentre trasmetto allora è errore a livello 1
se ci sono delle interferenze e il cavo è attaccato è errore a livello 2

3. **Network Layer**
   - Handles routing among **nodes** within a packet-switched network
     ↳ ha anche funzione di controllo del congestionamento
   - Unit of data exchanged between nodes in this layer is called a **packet**
     ↳ "dammi un IP e i dati ed io te li mando" => jitter, QoS, ritardi, tempo di transito sono problemi del network

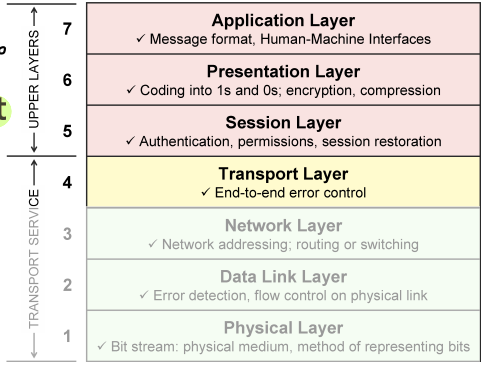
- The lower three layers are implemented on all network nodes

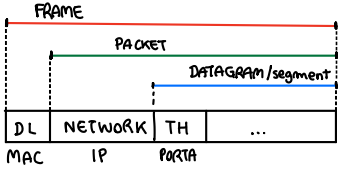
	7	**Application Layer** ✓ Message format, Human-Machine Interfaces
UPPER LAYERS	6	**Presentation Layer** ✓ Coding into 1s and 0s; encryption, compression
	5	**Session Layer** ✓ Authentication, permissions, session restoration
TRANSPORT SERVICE	4	**Transport Layer** ✓ End-to-end error control
	3	**Network Layer** ✓ Network addressing; routing or switching
	2	**Data Link Layer** ✓ Error detection, flow control on physical link
	1	**Physical Layer** ✓ Bit stream: physical medium, method of representing bits

DL	NETWORK	...
MAC	IP	

Description of Higher Layers

4. Transport Layer (es. TCP/UDP) → implementa l'ordinamento dei pacchetti → usa un nuovo indirizzo, la PORTA
   - Implements a process-to-process channel → è PROCESS to PROCESS perché è ASTRATTO per tutti gli strati sotto però ad esempio quando faccio una connessione devo dichiarare la porta
   - Unit of data exchanged in this layer is called datagram (↳ UDP) or segment (↳ TCP)
5. Session Layer
   - Provides a name space, authentication, coordination and general information among different transport streams that are part of a single application
6. Presentation Layer → i dati possono essere modificati, es. crittografati
   - Concerned about the format of data exchanged between peers
   - Data compression, conversion, encryption/decryption …
7. Application Layer → es. HTTP
   - Standardize common type of exchanges

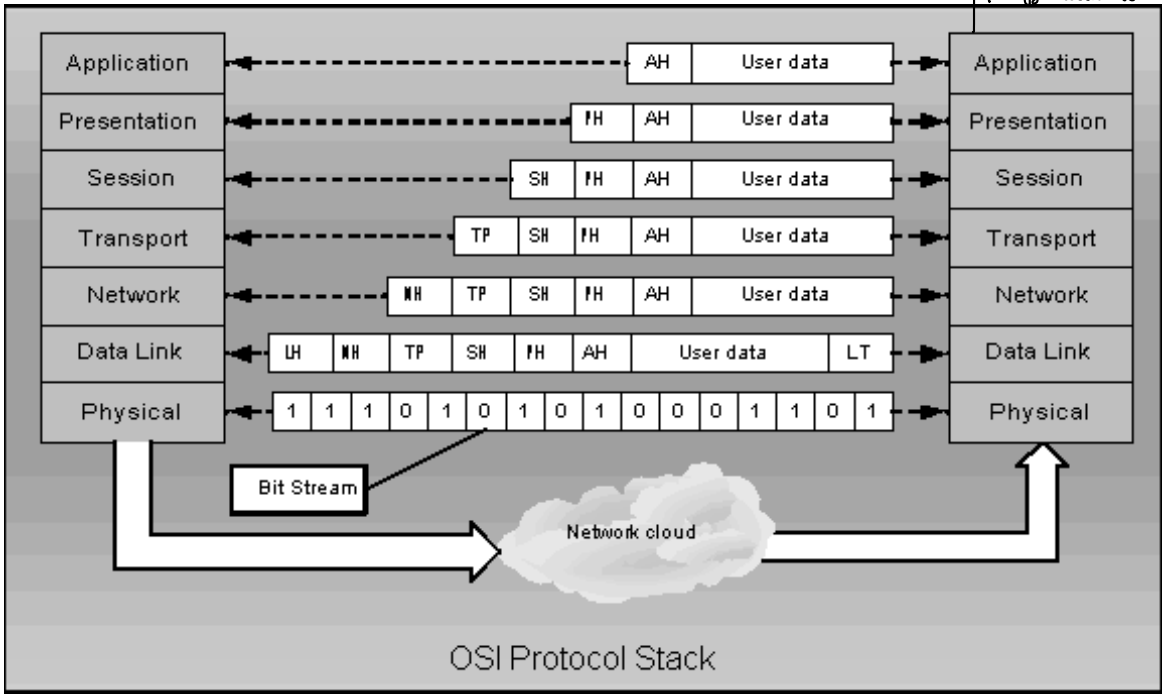




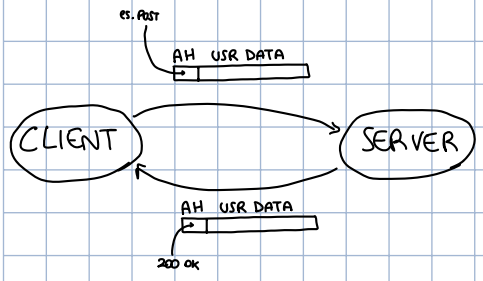
- The transport layer and the higher layers typically run only on end-hosts and not on the intermediate switches and routers

The ISO OSI Architecture

a livello applicativo io mando degli user data e imfostar. AH, ma ho astratto i passaggi sottostanti



OSI = Open Systems Interconnection



=> questo a livello applicativo, in realtà è astratto il canale
si utilizzano le API degli strati sottostanti

ad ogni livello ci sono i canali virtuali, in realtà tutto si appoggia
ai livelli sottostanti

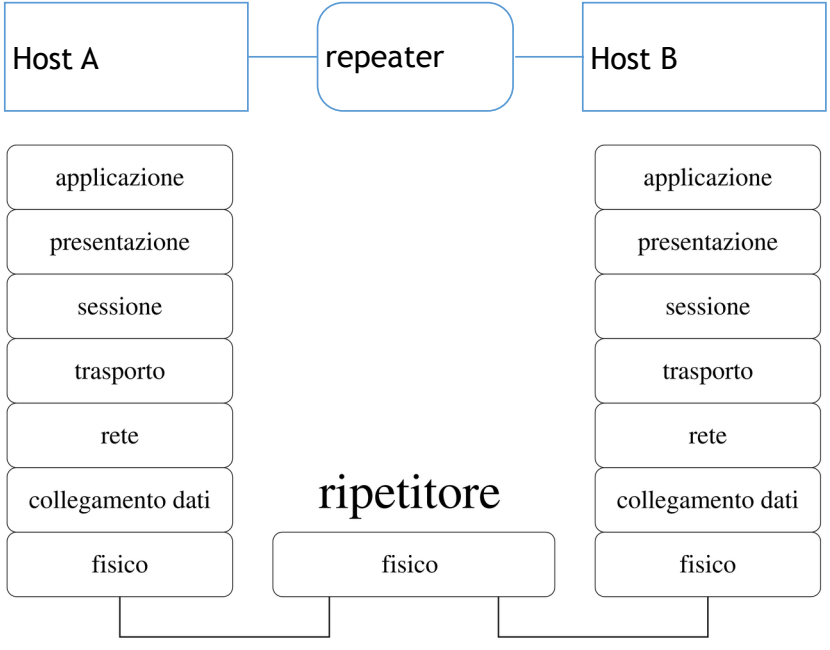
Ci sono alcuni dispositivi che implementano solo fino ad alcuni strati

es. repeater: prendono segnale fisico, lo decodificano e lo ricodificano dall'altra parte
↓
non amplificare
repeater che non sa rilevare errori

Repeater = level 1 switch

- Connects two physical media of the same type, which appear as a single one.
- Data units which are switched: single bits.
- Signals are basically the same on the two media, but not simply amplified: bits are decoded from one side and re-encoded on the other side.

→ vede solo bit



vediamo i frame ethernet, pacchetti wifi

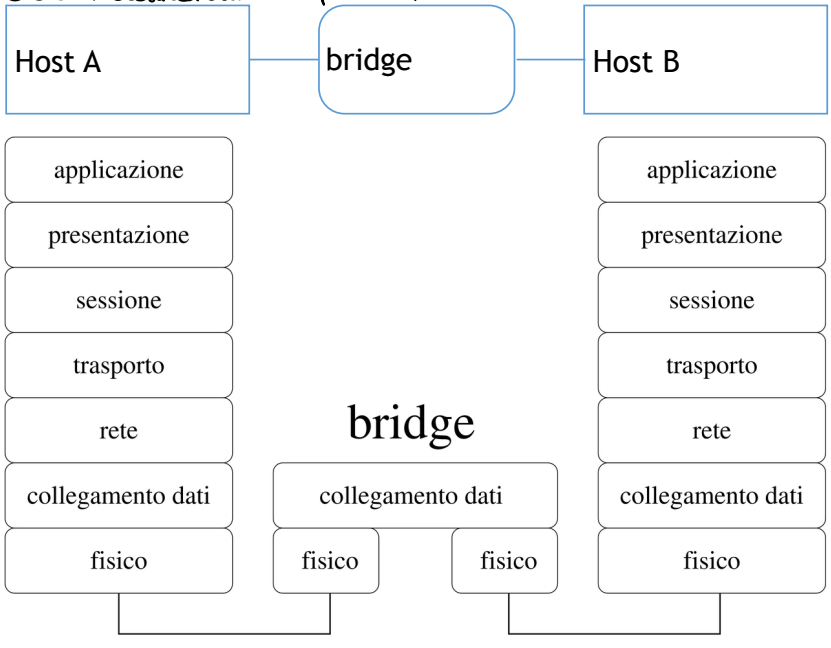
Bridge = level 2 switch

↳ prende la sequenza di bit, la interpreta come un frame e la ricodifica dall'altra parte → può trovare errori

- Connects two possibly different links, which appear as a single one.
- Data units which are switched: frames. Each frame is received from one side, checked for errors, and forwarded to the other.
- Frame headers are kept unchanged (usually), but physical encodings can differ.
- A *switch* is a multi-port bridge.

posso cambiare le codifiche dei frame

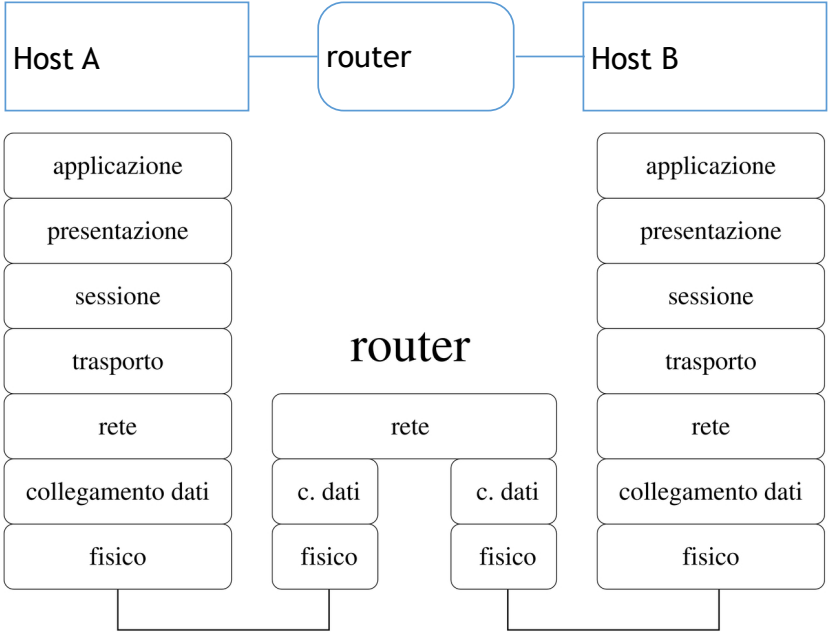
→ vede i frame



Router = level 3 switch

- Connects two different network segments, which appear as a single one.
- Data units which are switched: packets.
- Network header is not changed, but frame header is replaced => quando faccio il forwarding

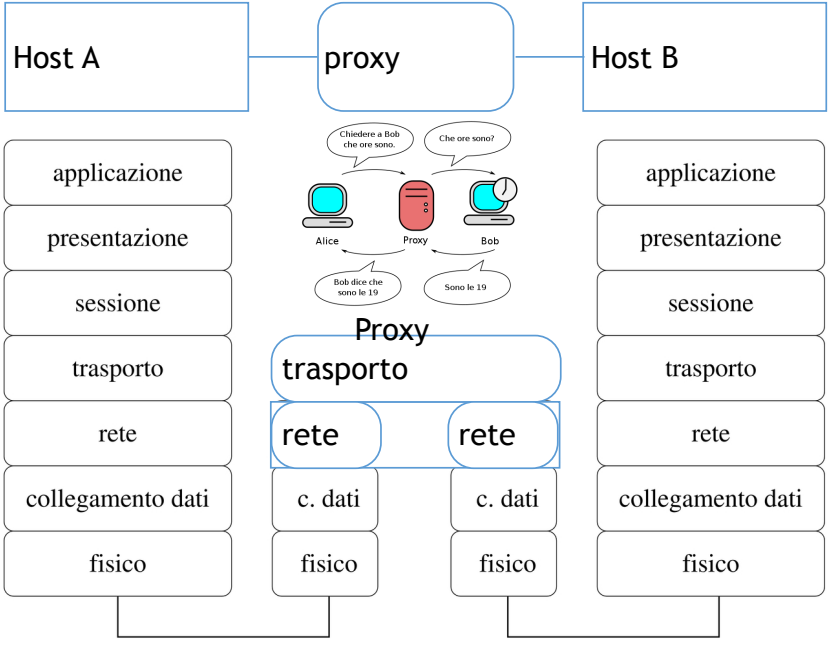
→ riesce ad implementare e' INSTRADAMENTO e scambia PACCHETTI



Proxy = level 4 switch

DATAGRAM: connectionless

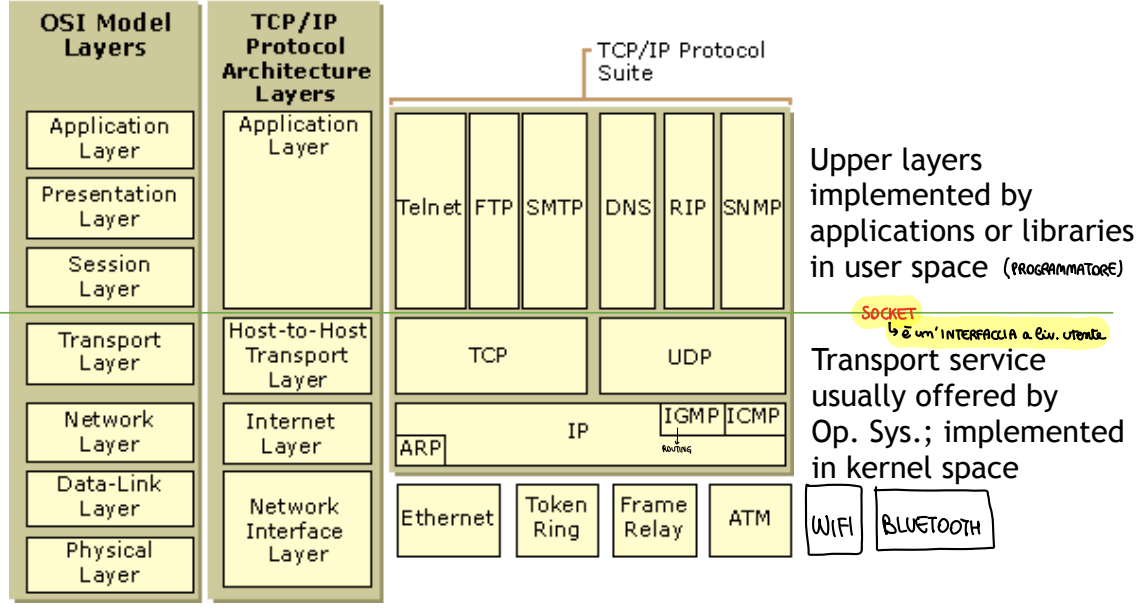
- Translates between different hosts. Data units which are switched: messages.
- Frame and network headers are replaced. Application protocol and transport header are maintained. ⇒ forwarding
- Allows to access a service (B) but connecting to a closer host (the proxy).→ a B la richiesta appare come effettuata dal proxy
- Useful for crossing firewalls, checking content …



Internet Architecture (aka the TCP/IP stack)

The Internet Architecture is simpler than ISO/OSI:

- Application layer covers Application, Presentation, Session
- Network interface layer covers Data-Link and Physical



=> IP non assicura che i canali sotto sono robusti e sicuri, il protocollo IP serve a far trasportare i dati

La STANZA è una rete

NOI siamo i nodi e possiamo scambiarci i messaggi solo passandoci la roba di mano → deve per forza passare agli adiacenti
↳ siamo i router, singole reti

ci divide in RIGHE e COLONNE, ognuno ha il suo identificatore (G 13)
↳ indirizzo IP, liv. 3 → se ci spostiamo cambia

l'indirizzo FISICO (nome cognome) non cambia se ti sposti

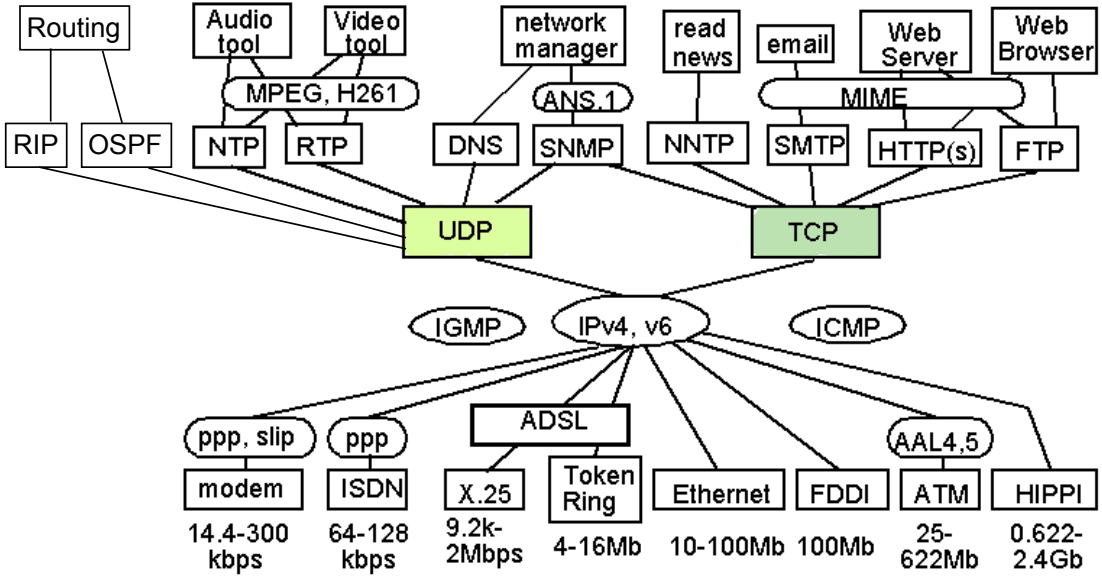
PASSAGGI:

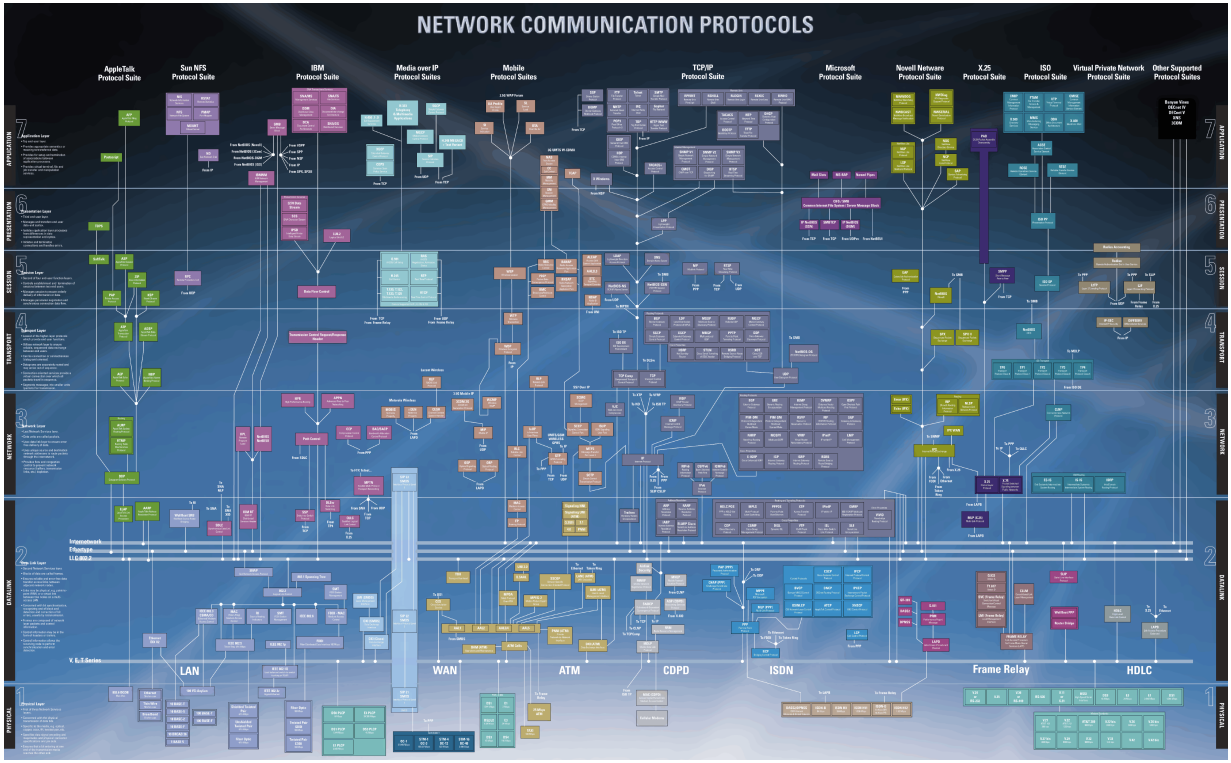
1. Divido il dato applicativo in tanti segmenti (a livello TRASPORTO creo i segmenti e me inserisco ciascuno in una busta (numerata)) con intestazioni di liv. 4 → INCAPSULAMENTO

2. inserisco un'altra intestazione con scritto l'indirizzo di destinaz. di liv 3 → INCAPSULAMENTO
   ↳ noi siamo dei router e dobbiamo reinstradare i pacchetti che non sono per noi.
      ↳ può succedere che un pacchetto si perda per errori di trasmissioni

3. il livello 4 deve "sbustare", riordinare e se manca qualcosa farsi rinviare tutto mandando un messaggio di rinvio una volta che c'è tutto e ricostruisce il messaggio complessivo

Internet Architecture: protocol graph





See at http://www.tehowners.com/info/Internet%20&%20Communication/Network%20Protocol.gif)

Internet Architecture

- Defined by IETF
- Three main features
  - **Does not imply strict layering.** The application can bypass the defined transport layers and to directly use IP or other underlying networks
  - An **hour-glass shape** – wide at the top, narrow in the middle and wide at the bottom. IP serves as the focal point for the architecture
  - In order for a new protocol to be officially included in the architecture, it must be provided both a protocol specification and at least one (preferably two) representative implementations

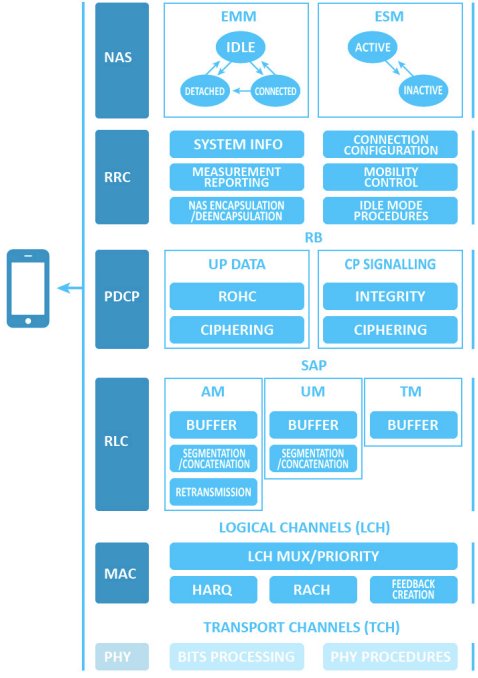
→ la TCP-IP ha vinto sulle altre scelte: (es. APPLE-TALK)

- architettura a CLESSIDRA: con un solo protocollo di rete (IPv4, IPv6) basta implementare solo IP nelle nuove tecnologie e tutto quello sopra IP automaticamente funziona. Nell'esempio fatto in classe se cambiassimo la busta che gira, tutto il resto poi funziona
  ↳ è facile aggiungere NUOVI PROTOCOLLI
- basata su protocolli APERTI: gratis

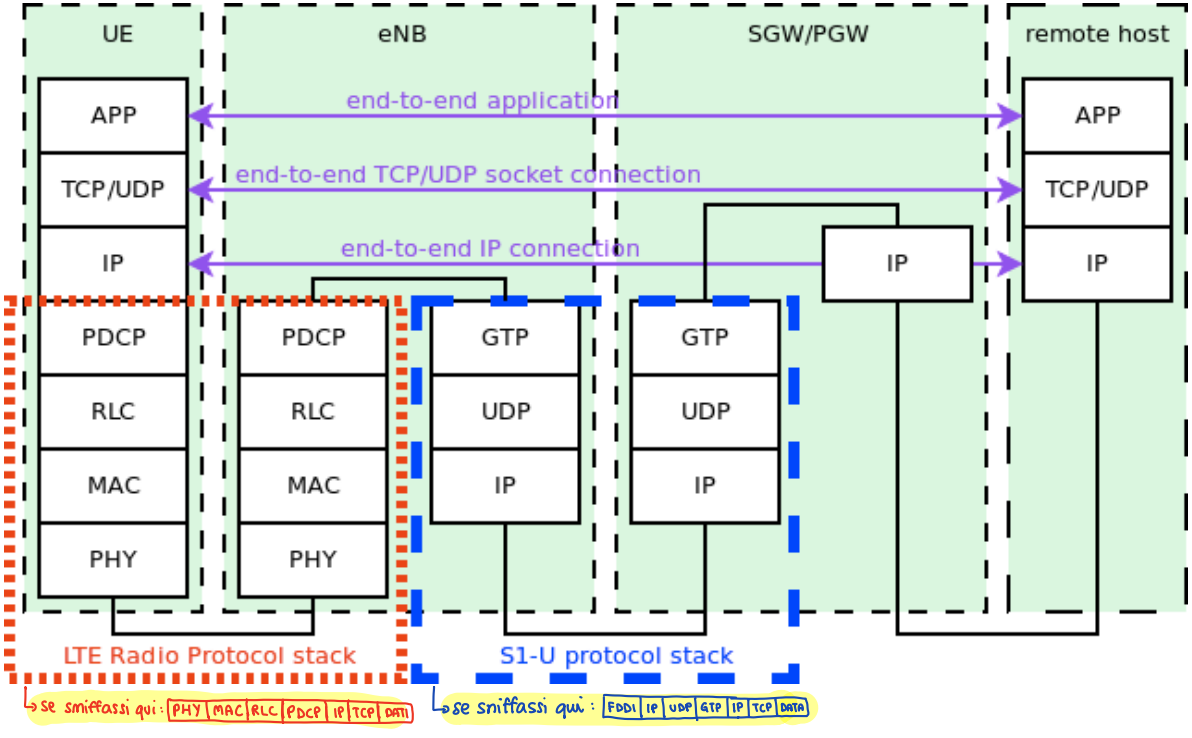
Other example: LTE-EPC (4G) stack

=> un altro tipo di STACK

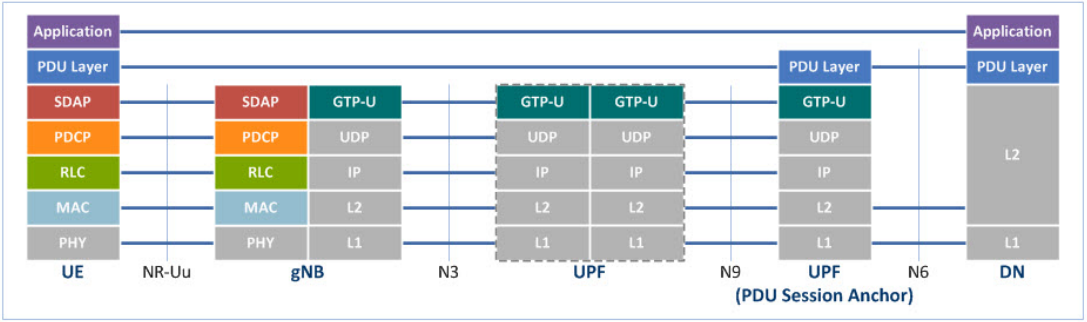
- LTE = Long Term Evolution
- Main goal: “to provide a high data rate, low latency and packet optimized radioaccess technology supporting flexible bandwidth deployments. Its network architecture has been designed with the goal to support packet-switched traffic with seamless mobility and great quality of service.”



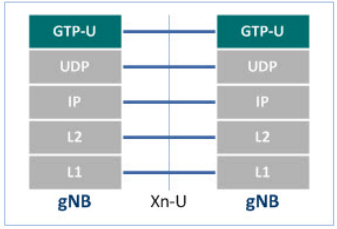
Other example: LTE-EPC (4G) stack

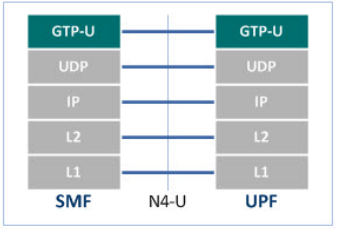


Other example: 5G stack



PDU Layer: IP, Ethernet, etc.





DN	: Data Network
gNB	: Next generation NodeB
GTP-U	: GPRS Tunneling Protocol User plane
MAC	: Medium Access Control
PDCP	: Packet Data Convergence Protocol
PDU	: Protocol Data Unit
RLC	: Radio Link Control
SDAP	: Service Data Adaptation Protocol
SMF	: Session Management Function
UE	: User Equipment
UPF	: User Plane Function
Xn-U	: Xn User plane

Stratified architectures: limits

- Not all services can fit easily in some layer
- Some services are *inherently* cross-layer
  - Security services
  - Quality of service

↳ quelli che riguardano la qualità, es. garantire che i pacchetti arrivino entro tot.

- In fact, these are "modalities", i.e. these describe how other services are offered
- In these cases, we implement them in one, some or even **all** layers...