



TechnoLabs S.p.A. R&D

# Reti di Calcolatori

La strada che porta a Internet



Università degli Studi di L'Aquila

Luigi Vetrano

# Reti di calcolatori

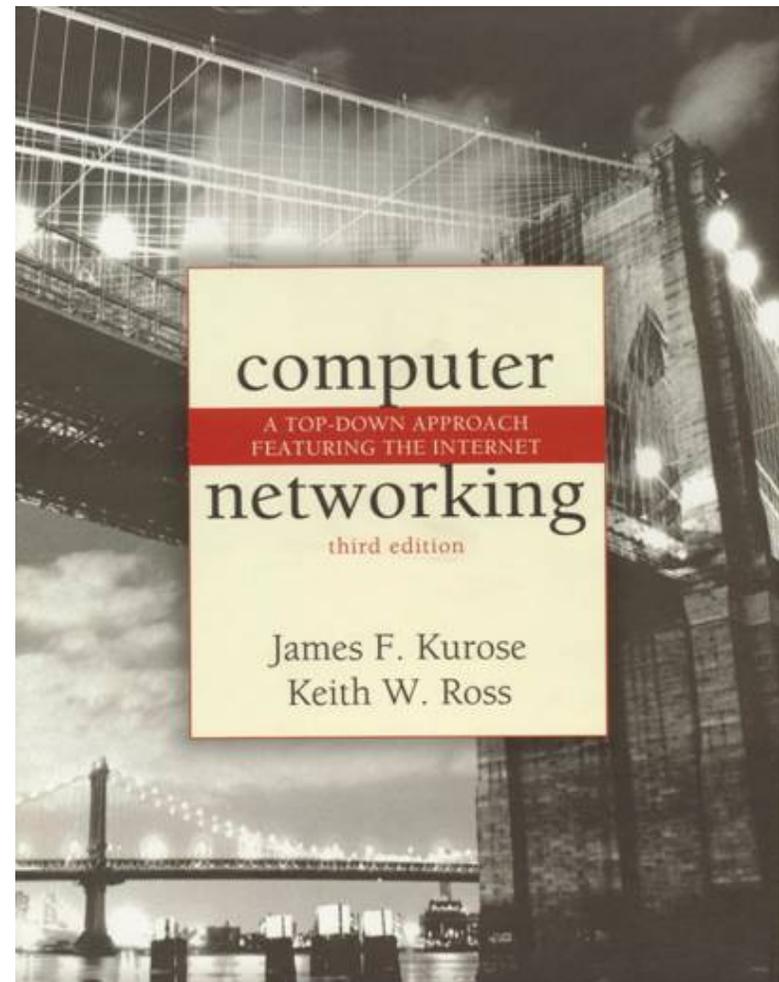
Parte di queste slides sono adattate dalle slides del libro:

*Computer Networking:  
A Top Down Approach  
Featuring the Internet,*

3rd edition.

Jim Kurose, Keith Ross

Addison-Wesley, 2005.



# Overview



- **Cosa è una rete**
- **Cosa è Internet**
  - ◆ Componenti
    - ◆ Cosa è un protocollo
  - ◆ Servizi
    - ◆ Client/server e peer-to-peer
    - ◆ Connectionless e connection-oriented
- **Network core**
  - ◆ Circuit/packet switching
    - ◆ TDM/FDM
    - ◆ packet network e VC
- **Perdite e ritardi in packet-switched network**
- **Struttura a livelli**
- **Infrastruttura di Internet e ISP**
- **Accessi alla rete e mezzi fisici**



# Obiettivo del Networking



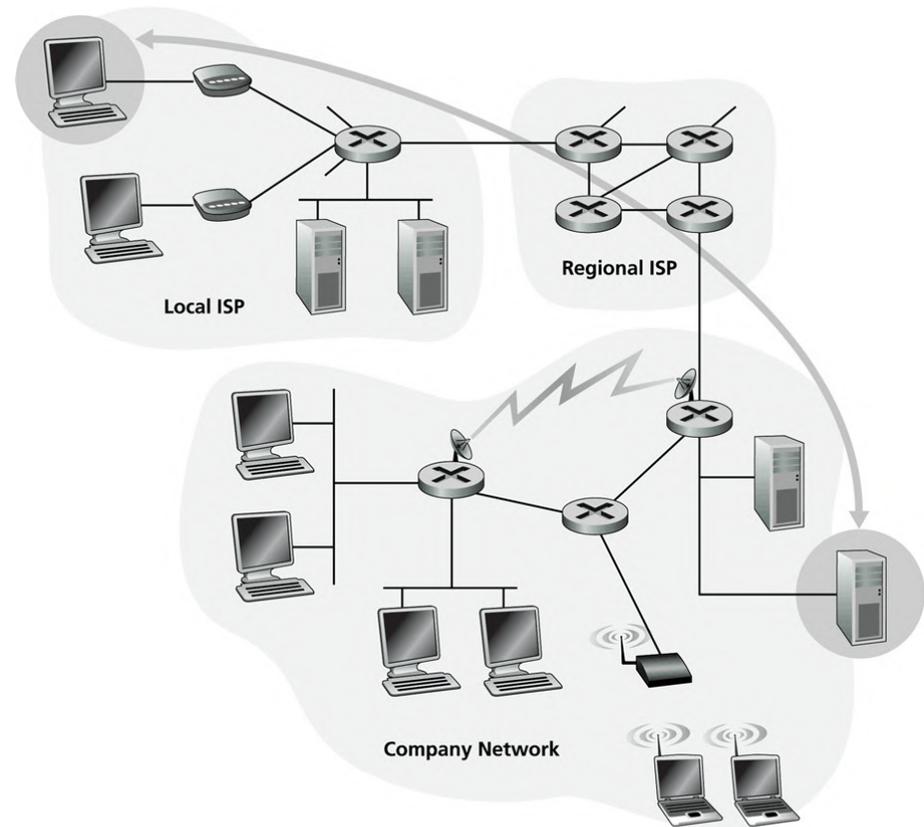
“Le persone e le loro macchine dovrebbero poter avere accesso alle informazioni e comunicare tra loro **facilmente** e **in sicurezza**, utilizzando **qualunque medium** o una combinazione di **media** – voce, dati, immagini, video, o multimedia – **in qualsiasi momento, dovunque**, in un **tempo ragionevole** e a **basso costo**.”

Oct 1992



# Cosa è una rete ?

- ☞ Un insieme **interconnesso** di computer **autonomi**
  - ◆ Accesso alle informazioni
  - ◆ Condivisione di risorse
  - ◆ Facilitazione della comunicazione
  
- ☞ Fatta di hardware:
  - ◆ apparati di interconnessione
  - ◆ apparati per il controllo della trasmissione
  
- ☞ ... e di software:
  - ◆ codifica e formattazione dei dati
  - ◆ rilevazione di errori e correzione
  - ◆ Instradamento delle informazioni fra i computer in rete
  - ◆ Servizi e applicazioni



# Cosa fa una rete ?

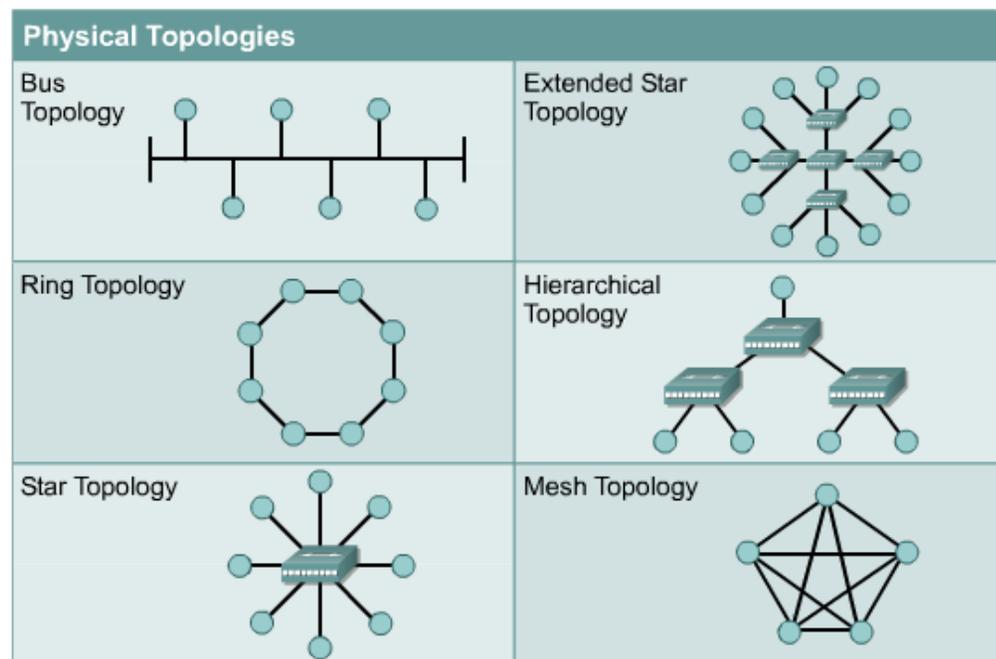
- ☞ Mette in comunicazione ambienti applicativi diversi:
  - ◆ Affidabile
  - ◆ Efficiente
  
- ☞ Rileva e corregge automaticamente:
  - ◆ Dati corrotti
  - ◆ Dati persi
  - ◆ Duplicazioni di dati (se si perde l'ack)
  - ◆ Distribuzione corretta dei pacchetti ricevuti con ordine diverso
  
- ☞ Trova cammini ottimali dalla sorgente alla destinazione (**routing**).



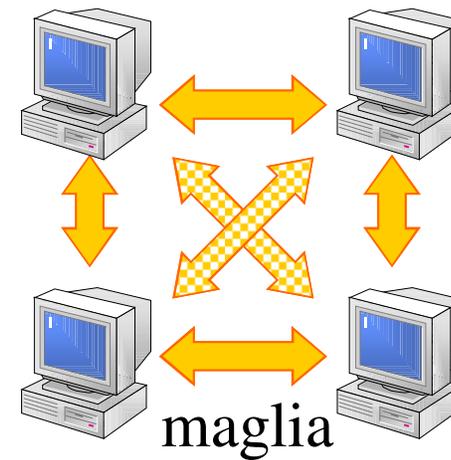
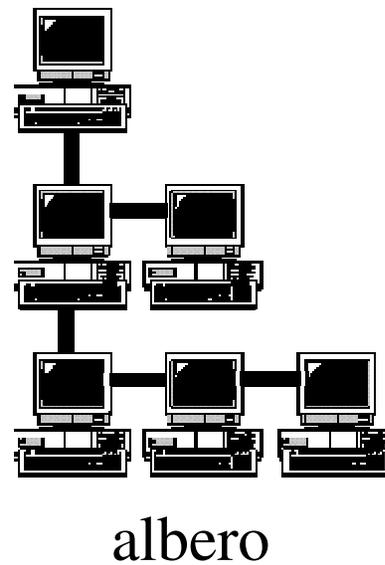
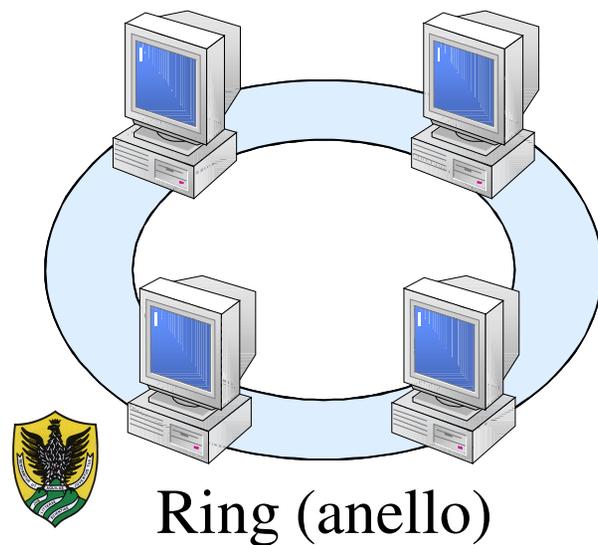
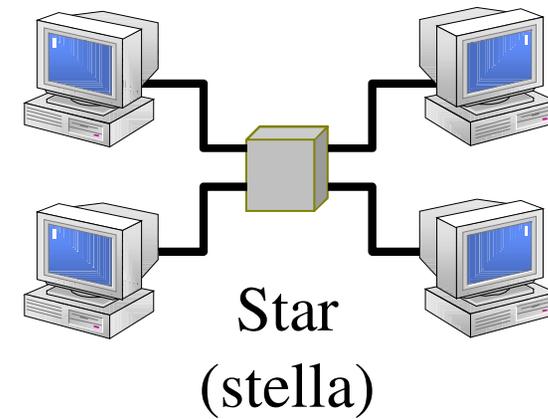
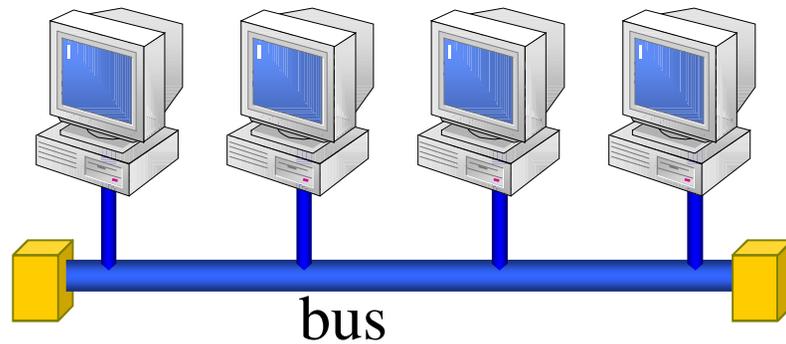
# Topologia di rete

## ☞ Definisce la struttura della rete

- ◆ La topologia fisica, che è il reale layout dei fili e/o dei media, e
- ◆ La topologia logica, che definisce come i media vengono usati dagli hosts per spedire i dati.

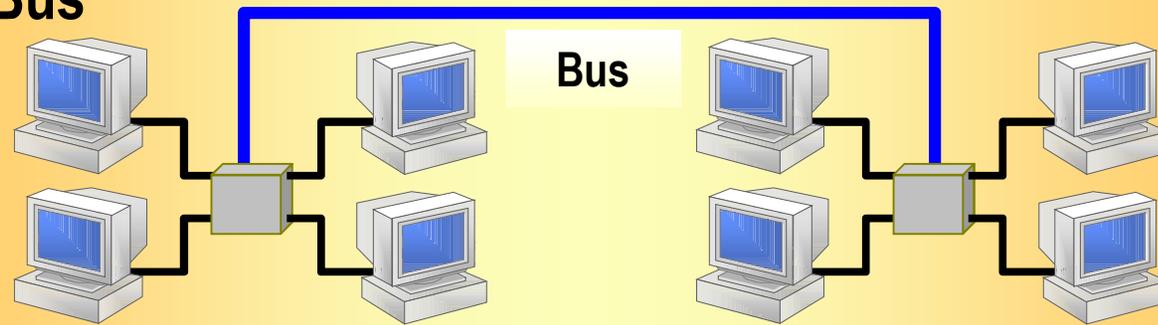


# Topologie di rete

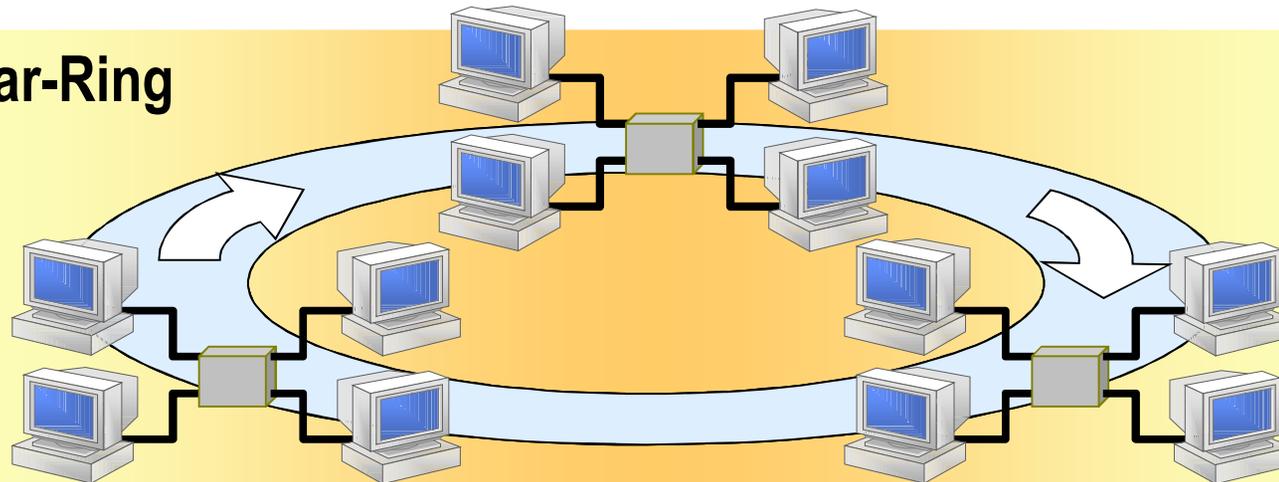


# Topologie ibride

## Star-Bus



## Star-Ring

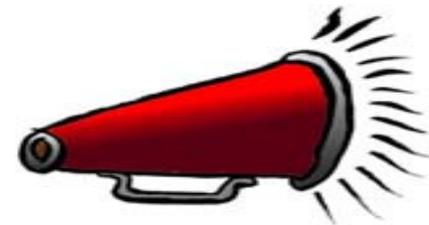


# Modalità di trasmissione

Punto-Punto



Broadcast

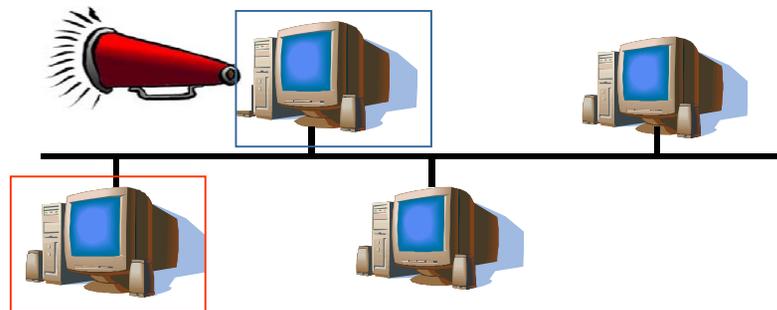




# Modalità di Trasmissione

## ☞ Broadcast

- ◆ Il canale di comunicazione è condiviso fra tutte le macchine
- ◆ I messaggi inviati (**pacchetti**) sono ricevuti da tutte le macchine
- ◆ Nel pacchetto è indicato l'indirizzo del destinatario
- ◆ Solo la macchina destinataria accetta il pacchetto trasmesso (esistono comunque indirizzi **broadcast** e **multicast**)



# Mezzi di trasmissione



## 'Wired'

- ☞ Rame
  - ◆ Doppini
  - ◆ Cavi coassiali
- ☞ Vetro
  - ◆ Fibre ottiche

## 'Wireless'

- ☞ Suoni
- ☞ Luce
- ☞ Raggi infrarossi
- ☞ Radiofrequenza
- ☞ Microonde



# La scala

Distanza fra le unità  
di elaborazione

Ambiente

0.1m	Circuito
1m	Sistema
10m	Stanza
100m	Edificio
1Km	Campus
10Km	Città
100Km	Nazione
1000Km	Continente
10000Km	Pianeta

Calcolatori  
multiprocessore

Rete Locale  
(LAN)

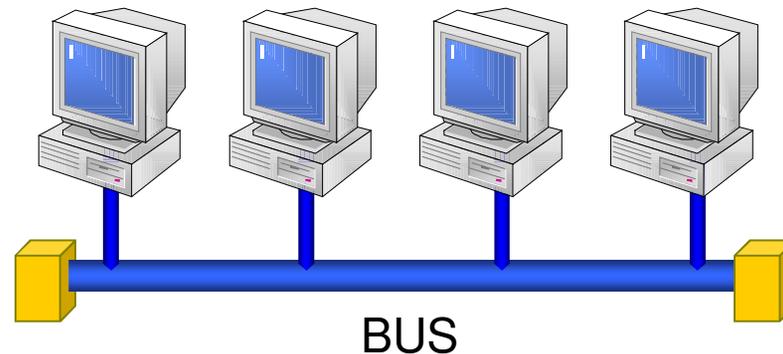
Rete Metropolitana (MAN)

Rete Geografica (WAN)

Internet



# Reti locali: LAN



- Rete di comprensorio, senza attraversamento di suolo pubblico
- Caratteristiche
  - ◆ Velocità trasmissiva molto elevata ( $> 10\text{Mb/s}$ )
  - ◆ Distanze ridotte ( $< \text{km}$ )
  - ◆ Mezzi trasmissivi anche non conformi alle raccomandazioni CCITT (standard internazionali per la telefonia)
  - ◆ Realizzata e gestita dal proprietario del comprensorio



# Reti metropolitane: MAN



- ☞ Rete in ambito urbano con disponibilità di canali trasmissivi veloci: sono “grosse” LAN e hanno simile tecnologia
- ☞ Caratteristiche
  - ◆ Velocità trasmissiva  $\underline{v}$  elevata ( $2 \text{ Mb/s} < \underline{v} < 140 \text{ Mb/s}$ )
  - ◆ Limitata ad una città o a poche città vicine
  - ◆ Mezzi trasmissivi conformi agli standard CCITT (telefonici), tipicamente in fibra ottica
  - ◆ Realizzata e gestita dalle società di telefonia e trasmissione dati pubblica
- ☞ Es: Standard IEEE 802.6 (DQDB)
  - ◆ Distributed Queue Dual Bus
  - ◆ Esempio: 160 Km - 45 Mbps



# Reti geografiche: WAN



- Rete in ambito nazionale o internazionale utilizzando qualsiasi mezzo trasmissivo (satellite nel caso peggiore)
  
- Interconnettono LAN/MAN
  
- Caratteristiche
  - ◆ Velocità trasmissiva  $\underline{v}$  medio-bassa ( $9.6 \text{ kb/s} < \underline{v} < 34 \text{ Mb/s}$ )
  - ◆ Distanze elevate (che determinano alti costi e basse velocità)
  - ◆ Mezzi trasmissivi conformi agli standard CCITT (telefonici), talvolta gli stessi usati per la rete telefonica convenzionale

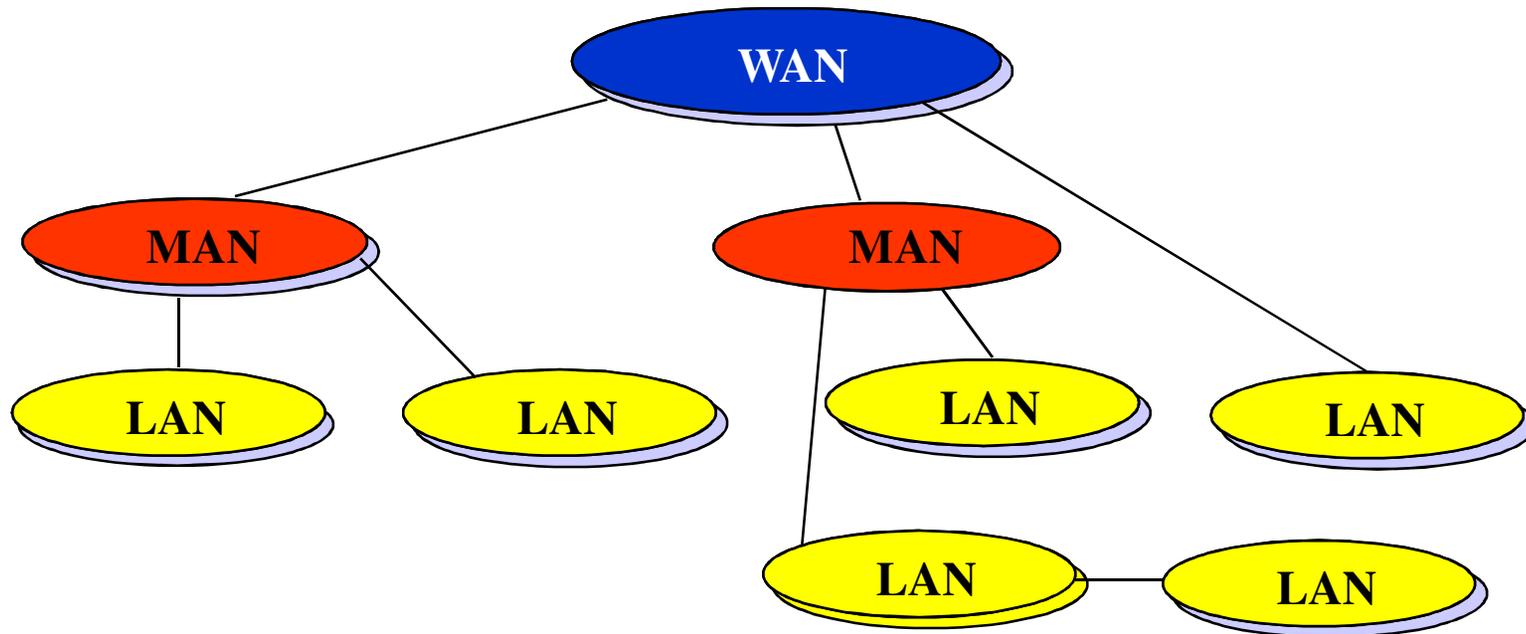


# Esempi di estensioni WAN

- Modem
- ISDN
- ADSL
- Frame Relay
- T1/E1
- T3
- STS-1, STS-3, STS-48 (SONET)
- STM-1, STM-4, STM-16 (SDH)
- GPRS/UMTS
- HiperLan/WiMax



# Struttura



- ☞ LAN: Local Area Network
- ☞ MAN: Metropolitan Area Network
- ☞ WAN: Wide Area Network



# Overview



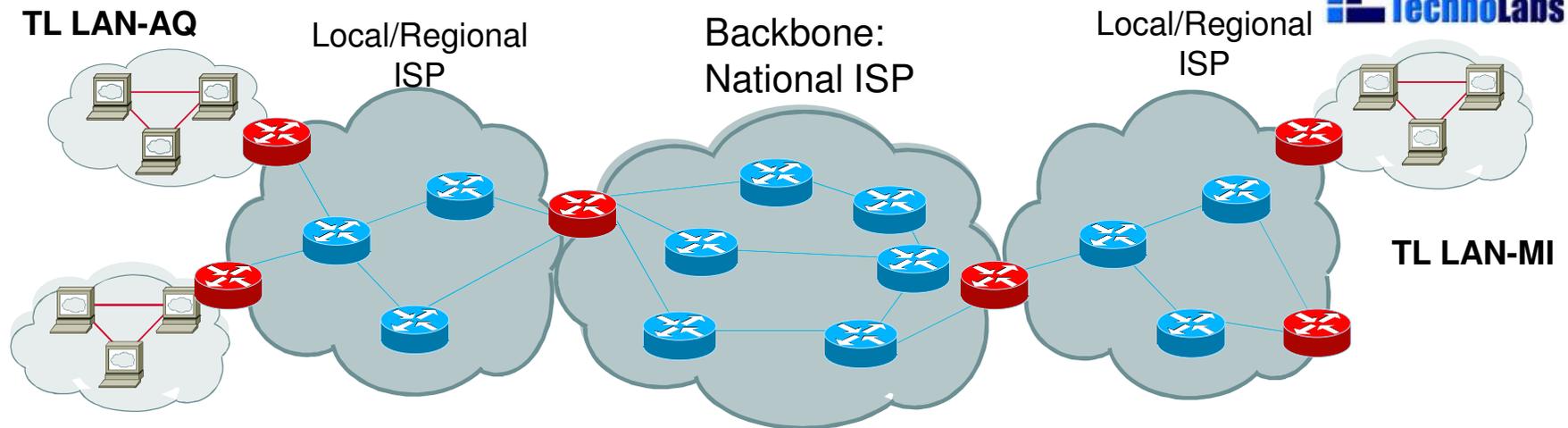
- Cosa è una rete
- **Cosa è Internet**
  - ◆ Componenti
    - ◆ Cosa è un protocollo
  - ◆ Servizi
    - ◆ Client/server e peer-to-peer
    - ◆ Connectionless e connection-oriented
- Network core
  - ◆ Circuit/packet switching
    - ◆ TDM/FDM
    - ◆ packet network e VC
- Perdite e ritardi in packet-switched network
- Struttura a livelli
- Internet structure and ISPs
- Accessi alla rete e mezzi fisici



# Cosa è 'Internet' ?



# Internet: Infrastruttura



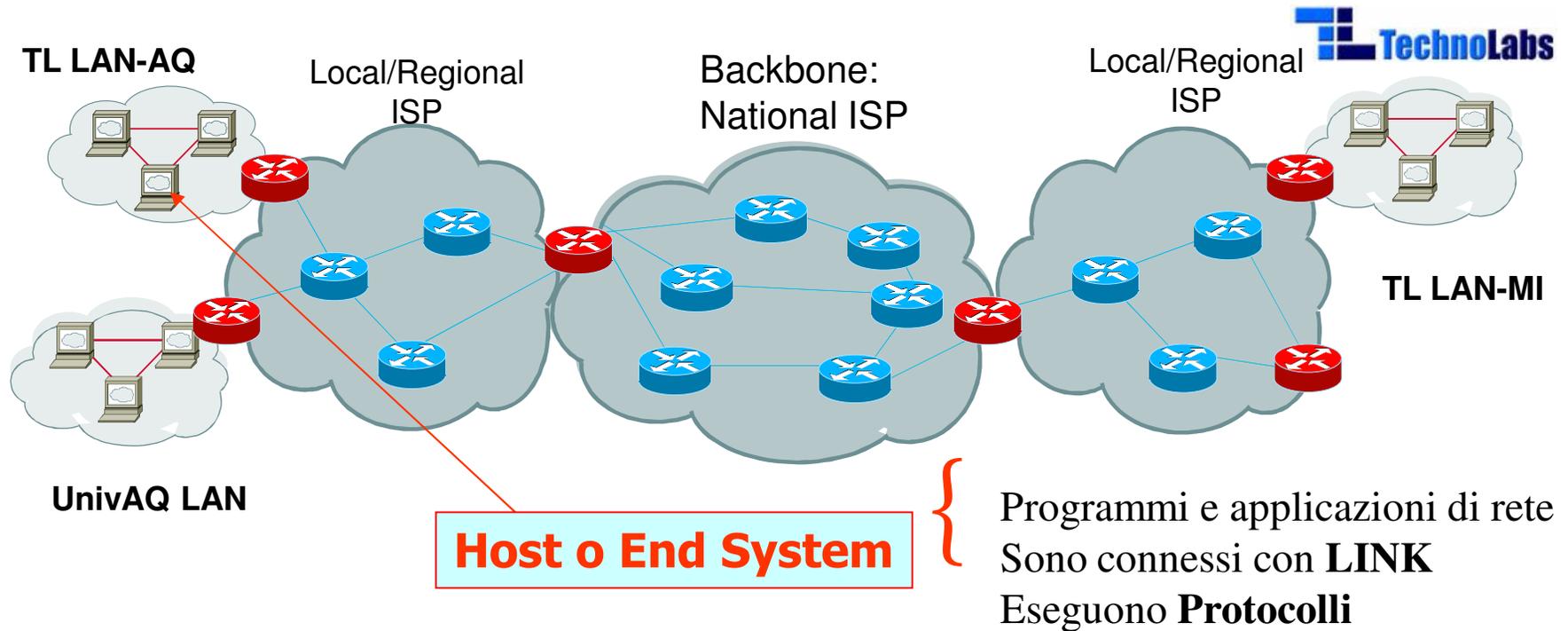
- Residential Access
  - ◆ Modem
  - ◆ DSL
  - ◆ Cable modem
- Campus network access
  - ◆ Ethernet
  - ◆ FDDI
  - ◆ Wireless



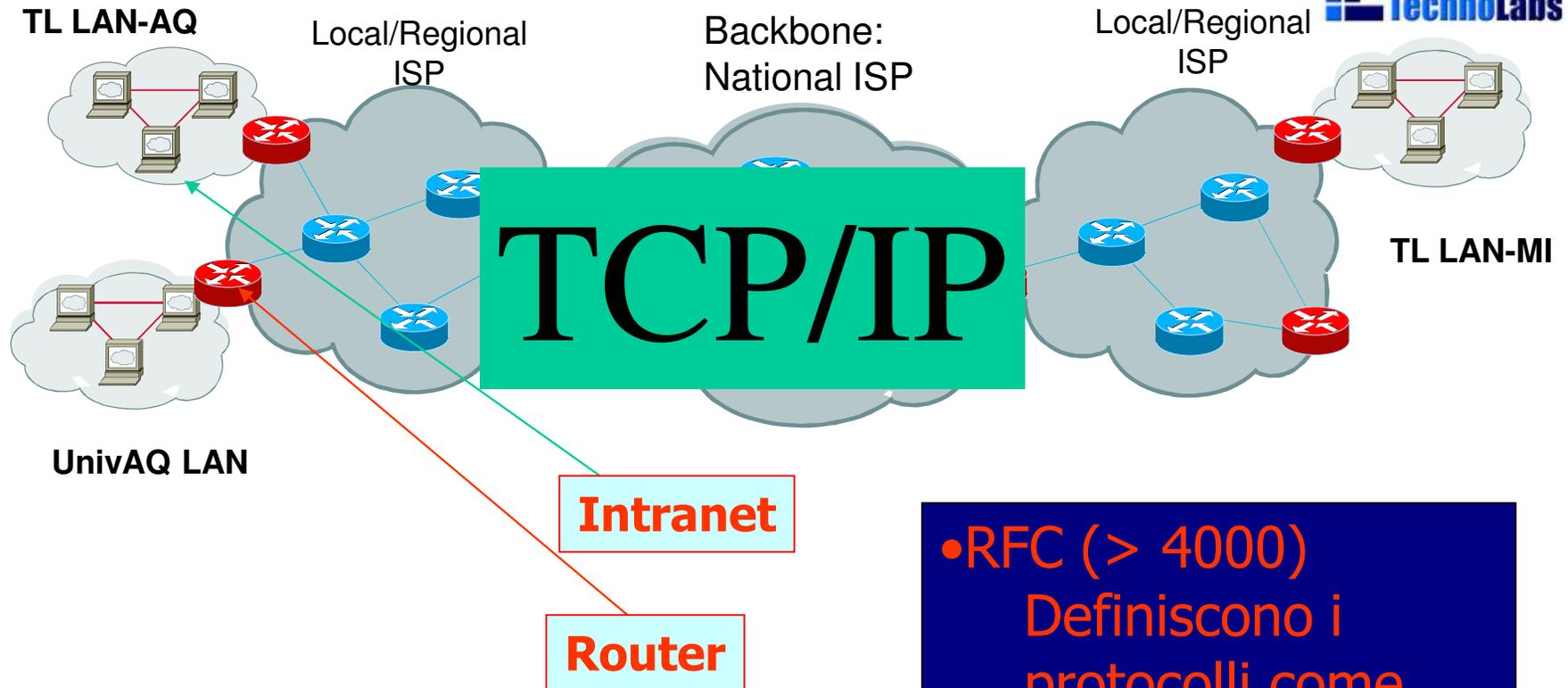
- Access to ISP, Backbone transmission
  - ◆ T1/T3, OC-3, OC-12
  - ◆ ATM, SONET, WDM

- Internet Service Providers
  - ◆ Local/Regional/National
  - ◆ They exchange packets at Point of Presence (POP)

# Internet: Infrastruttura



# Internet: Infrastruttura



- RFC (> 4000) Definiscono i protocolli come HTTP, TCP, UDP, FTP, IRC, RTP, SMTP, SNMP



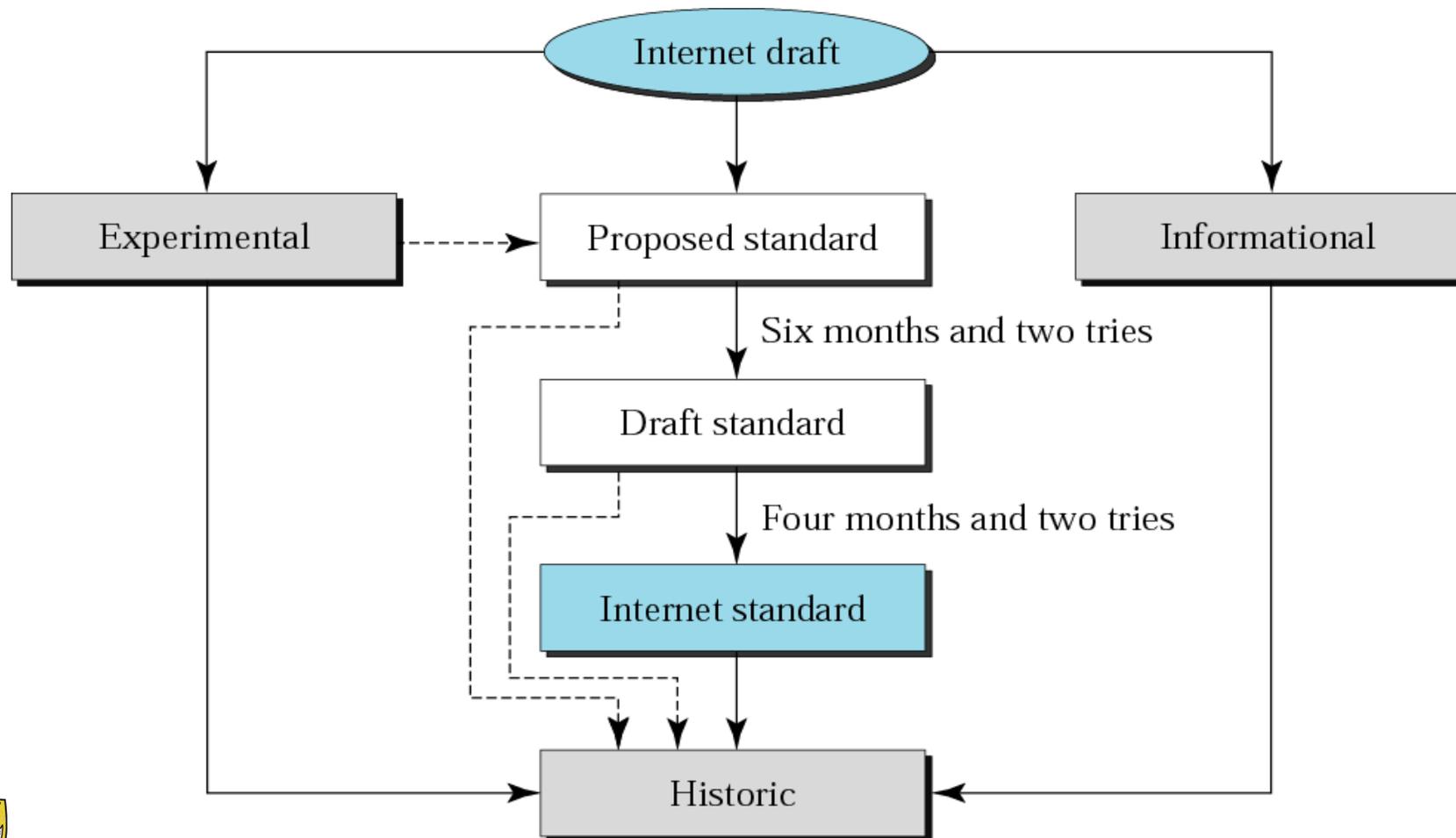
# RFC: Wikipedia



- Un **Request for Comments** (RFC) è un documento che riporta informazioni o specifiche riguardanti nuove ricerche, innovazioni e metodologie dell'ambito informatico o più nello specifico, di **internet**.
- Attraverso l'Internet Society gli ingegneri o gli esperti informatici possono pubblicare dei memorandum, sottoforma di RFC, per esporre nuove idee o semplicemente delle informazioni che una volta vagliate dall'**IETF** possono diventare degli standard Internet.
- Gli RFC furono inizialmente pubblicati nel 1969 come parte del progetto **ARPANET**.
- **RFC Editor** è il manutentore degli RFC ed è responsabile della loro edizione e del loro ciclo di vita.



# Ciclo di vita di un RFC



# Internet: una visione a servizi

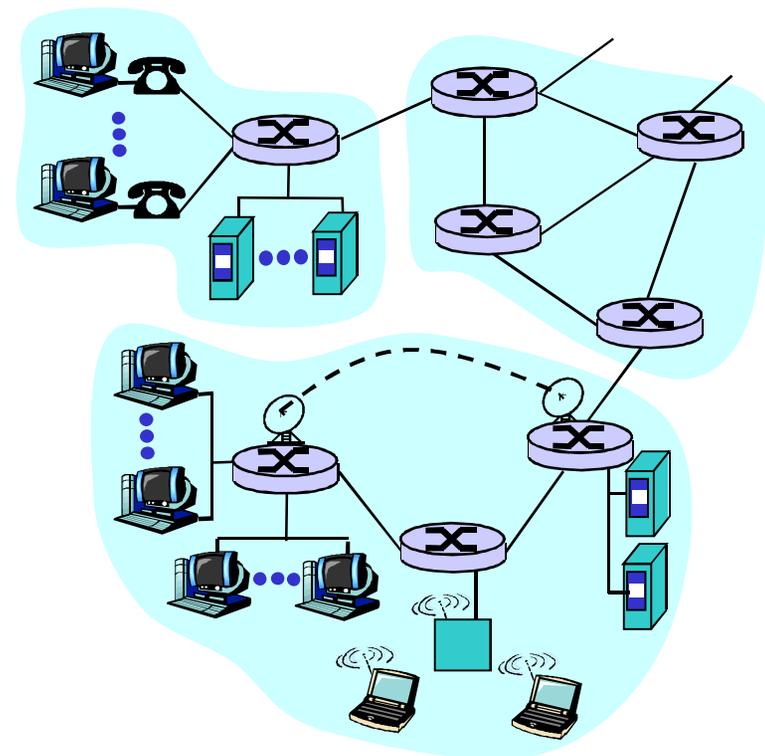
☞ **Infrastruttura di comunicazione che permette l'esecuzione di applicazioni distribuite:**

- ◆ WWW, email, e-commerce, Information Retrieval, game,
- ◆ altro?

☞ **Servizi:**

- ◆ connectionless
- ◆ connection-oriented

☞ **Nessuna garanzia sul tempo richiesto (non ancora)**



# Internet vista dal lato servizi



Internet come infrastruttura che permette **applicazioni distribuite**  
Queste applicazioni hanno a disposizione due classi di servizi:

☞ **Connection-oriented** services

- ◆ Garanzia che i dati trasmessi da Tx raggiungano Rx completamente e in ordine

☞ **Connectionless** services

- ◆ Nessuna garanzia della consegna

Problema: attualmente Internet non offre nessun servizio che garantisca il tempo di attraversamento

Sia ben chiaro che in un servizio connection-oriented, l'idea di connessione è implicitamente nota ai soli end-systems, gli apparati intermedi non sanno nulla della connessione né hanno una qualche memoria dello stato precedente.

Un servizio connection-oriented prevede sempre una fase di handshaking prima di iniziare una sessione (serve, tra l'altro, ad allocare le risorse necessarie alla manutenzione della connessione).



# Quindi cosa è Internet ?



☞ Internet è una infrastruttura per applicazioni distribuite in cui nuove applicazioni sono costantemente inventate e provate.

- ◆ E-Mail
- ◆ Chat Sessions
- ◆ Usenets and Listservs
- ◆ Web
- ◆ Ftp
- ◆ P2P
- ◆ SOAP
- ◆ GRID Computing
- ◆ Social Networking



# L'origine del nome

- ☞ Curiosa è la circostanza secondo cui la definizione ufficiale di Internet viene coniata e pubblicata solo nel 1995, (circa 35 anni dopo i primi esperimenti), dopo approvazione all'unanimità, da parte della FNC (Federal Networking Council).
- ☞ Tra i partecipanti di quella sessione della FNC troviamo **Vinton Cerf** e **Robert Kahn**, giustamente ritenuti tra gli artefici della invenzione di Internet.
- ☞ Questa definizione, l'unica ufficiale e reperibile su Internet stessa recita, in modo deliziosamente ricorsivo così:



# Internet : definizione ufficiale



- ☞ Sistema di informazione globale che
  - ◆ è logicamente interconnesso da un address space unico e globale, basato sull'Internet Protocol (IP) o le sue successive estensioni/sviluppi;
  - ◆ è in grado di supportare la comunicazione tramite la suite Transmission Control Protocol/Internet Protocol (TCP/IP) o le sue successive estensioni/sviluppi, e/o altri protocolli compatibili con l'IP; e
  - ◆ fornisce, utilizza o rende accessibili, sia pubblicamente che privatamente, servizi di comunicazione di alto livello stratificati e basati sulla correlata infrastruttura qui descritta.

(Notare come si sia volutamente fatto uso di una definizione ricorsiva)



# Internet: l'HW

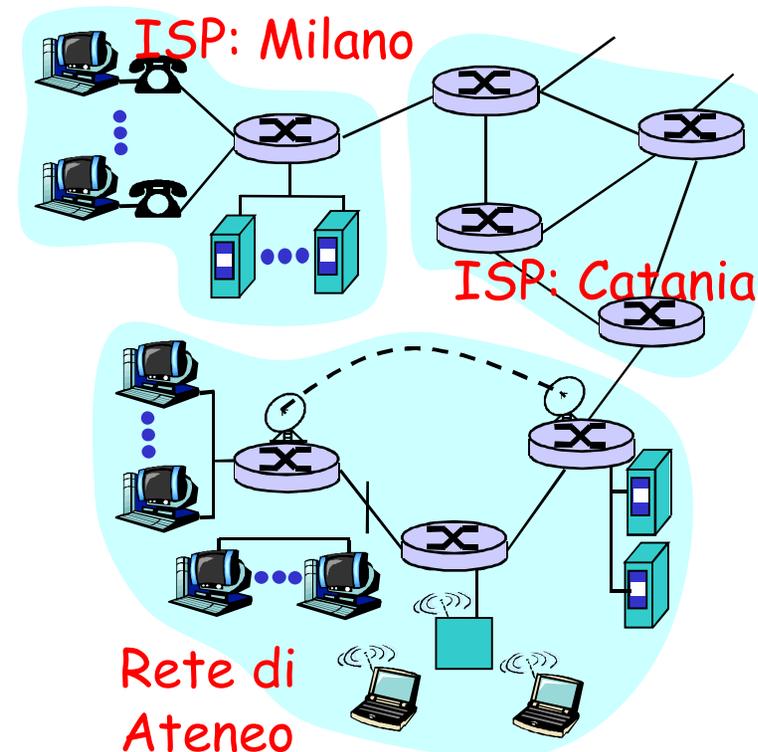
➔ Milioni di dispositivi computazionali connessi in rete: *hosts, end-systems*

- ◆ Pc, workstation, server
- ◆ PDA, cellulari, frigoriferi

➔ *Collegamenti*

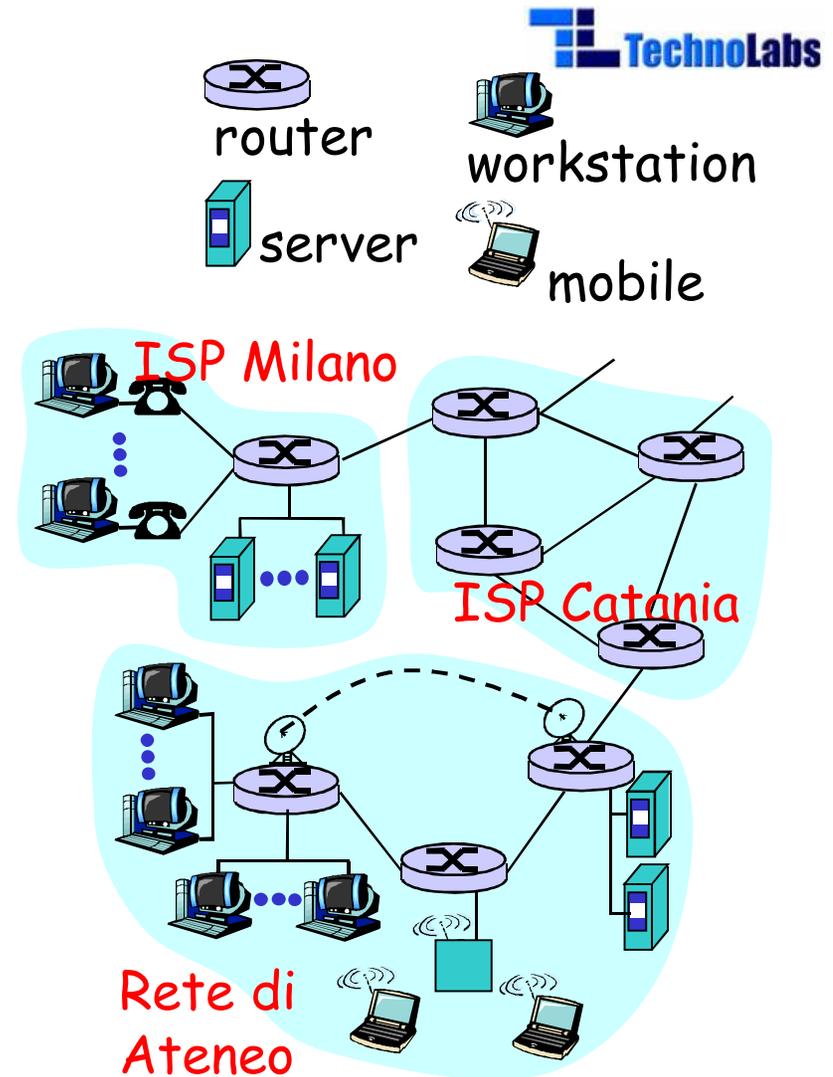
- ◆ Fibre ottiche, ponti radio, satellite

➔ *router*: compito di inoltrare pezzi di dati (*pacchetti*) lungo la rete



# Internet: SW

- ➔ **Protocolli di comunicazione:**  
meccanismi per la trasmissione dei messaggi
  - ◆ TCP, IP, HTTP, FTP, PPP
- ➔ **Internet: “network of networks”**
  - ◆ gerarchica
- ➔ **Internet: standard**
  - ◆ RFC: Request for comments
  - ◆ IETF: Internet Engineering Task Force



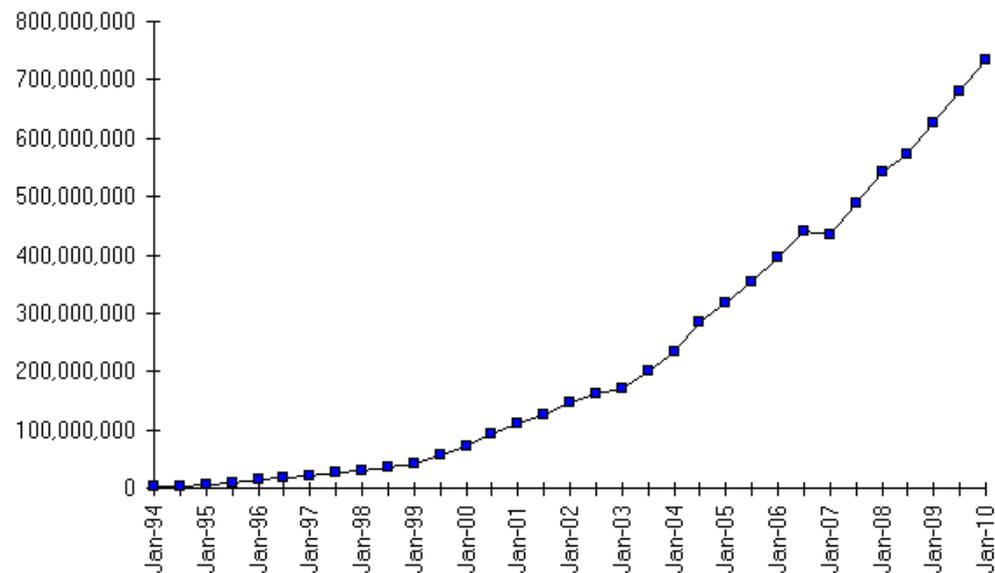
# Crescita di Internet



## Number of Hosts on the Internet:

Aug. 1981	213
Oct. 1984	1,024
Dec. 1987	28,174
Oct. 1990	313,000
Jul. 1993	1,776,000
Jul. 1996	19,540,000
Jul. 2000	93,047,000
Jul. 2002	162,128,493

Internet Domain Survey Host Count



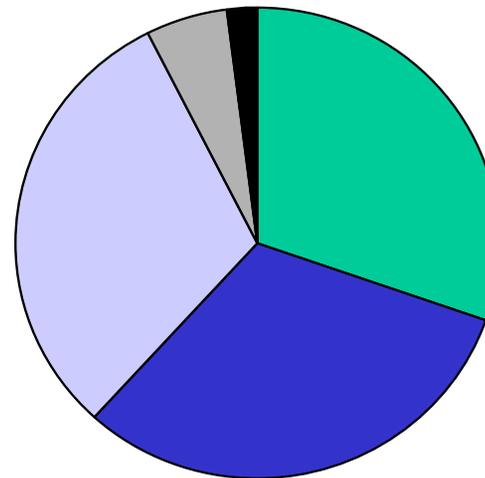
Source: Internet Systems Consortium ([www.isc.org](http://www.isc.org))



# Utenti Internet – Sett. 2002



CAN/US -	182.67M
Europe -	190.92M
Asia/Pac -	187.24M
Latin Am -	33.35M
Africa -	6.31M
Mid-east -	5.12M
-----	
Total -	605.6 M



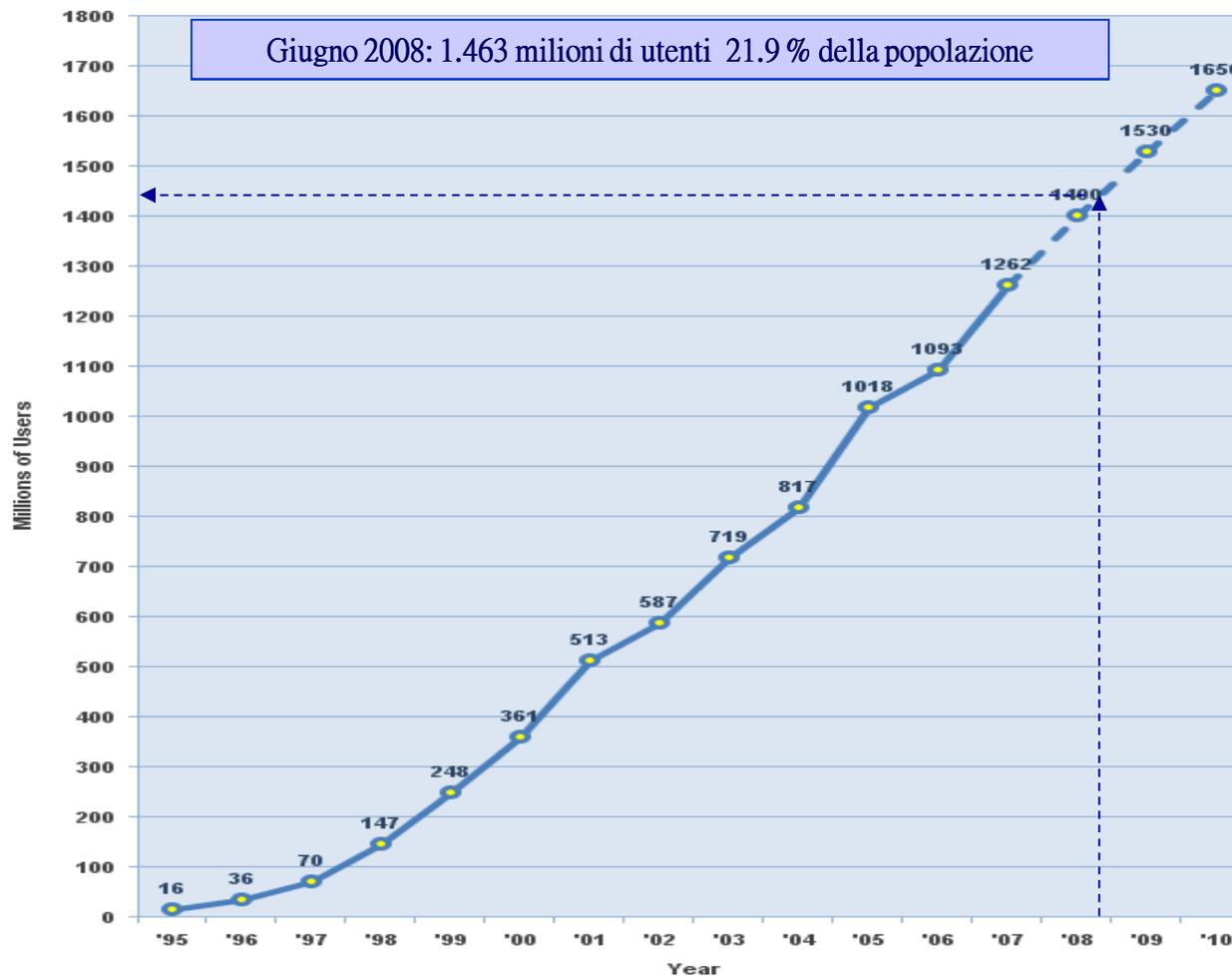
- CAN/US**
- Europe**
- Asia/Pac**
- Latin Am**
- Africa**
- Mid East**



*(Source [www.nua.ie](http://www.nua.ie))*

# And the "Global Village" became a Reality

Internet Users in the World  
Growth 1995 - 2010

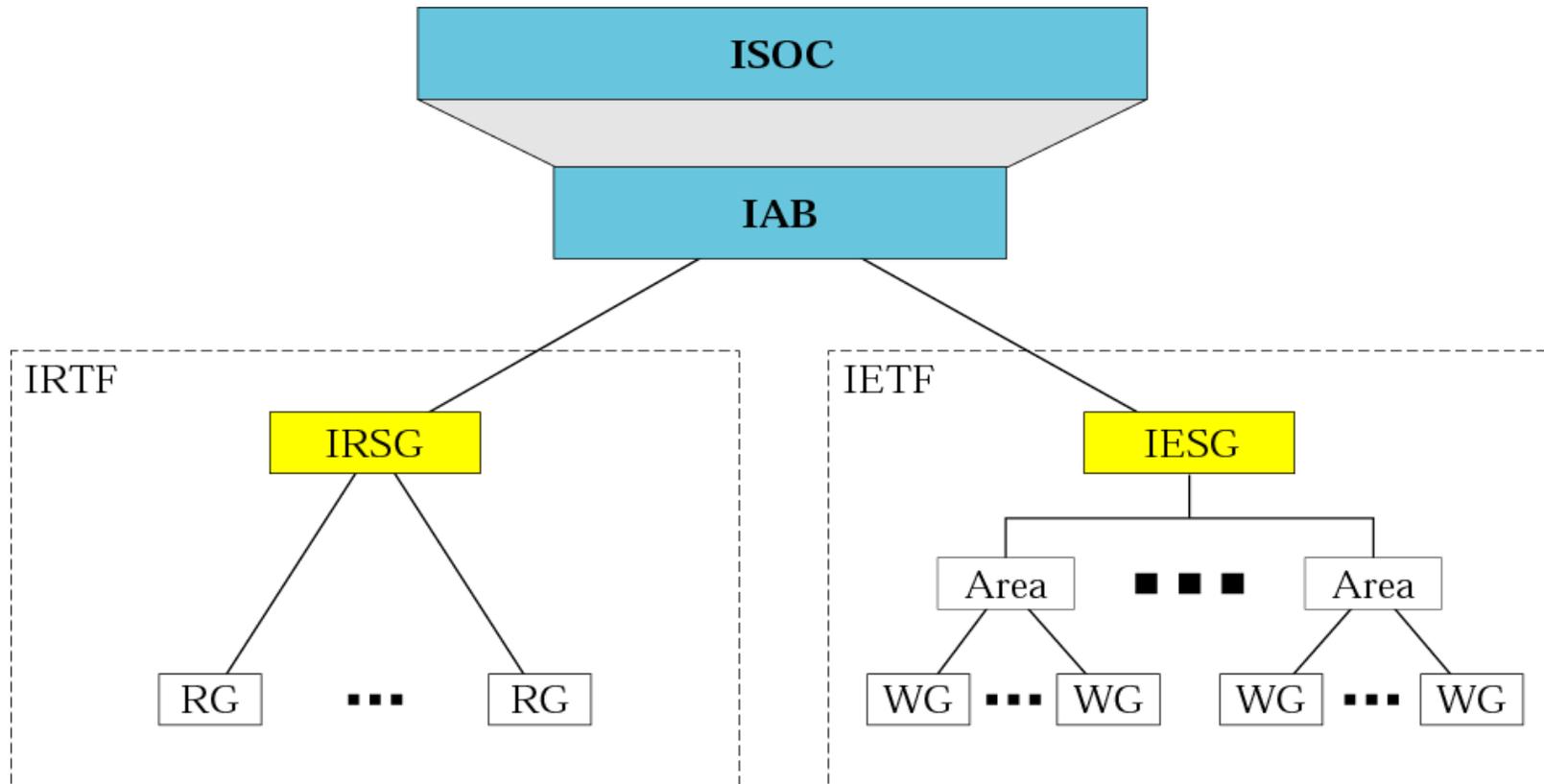


Giugno 2008: 1.463 milioni di utenti 21.9 % della popolazione



Source: [www.internetworldstats.com](http://www.internetworldstats.com) - January, 2008  
Copyright © 2008, Miniwatts Marketing Group

# Chi amministra Internet



Internet Society (ISOC)  
Internet Engineering Task Force (IETF)  
Internet Assigned Numbers Authority (IANA)  
Network Information Center (NIC)

Internet Architecture Board (IAB)  
Internet Research Task Force (IRTF)  
Names and Numbers (ICANN)



# Cos'è un protocollo?



## Protocolli umani:

Gli umani eseguono continuamente dei protocolli

- ☞ “pronto..”
- ☞ “pronto, sono Pippo, come stai?”

... regole che governano la condotta delle persone (azioni - reazioni) nello scambio dei messaggi

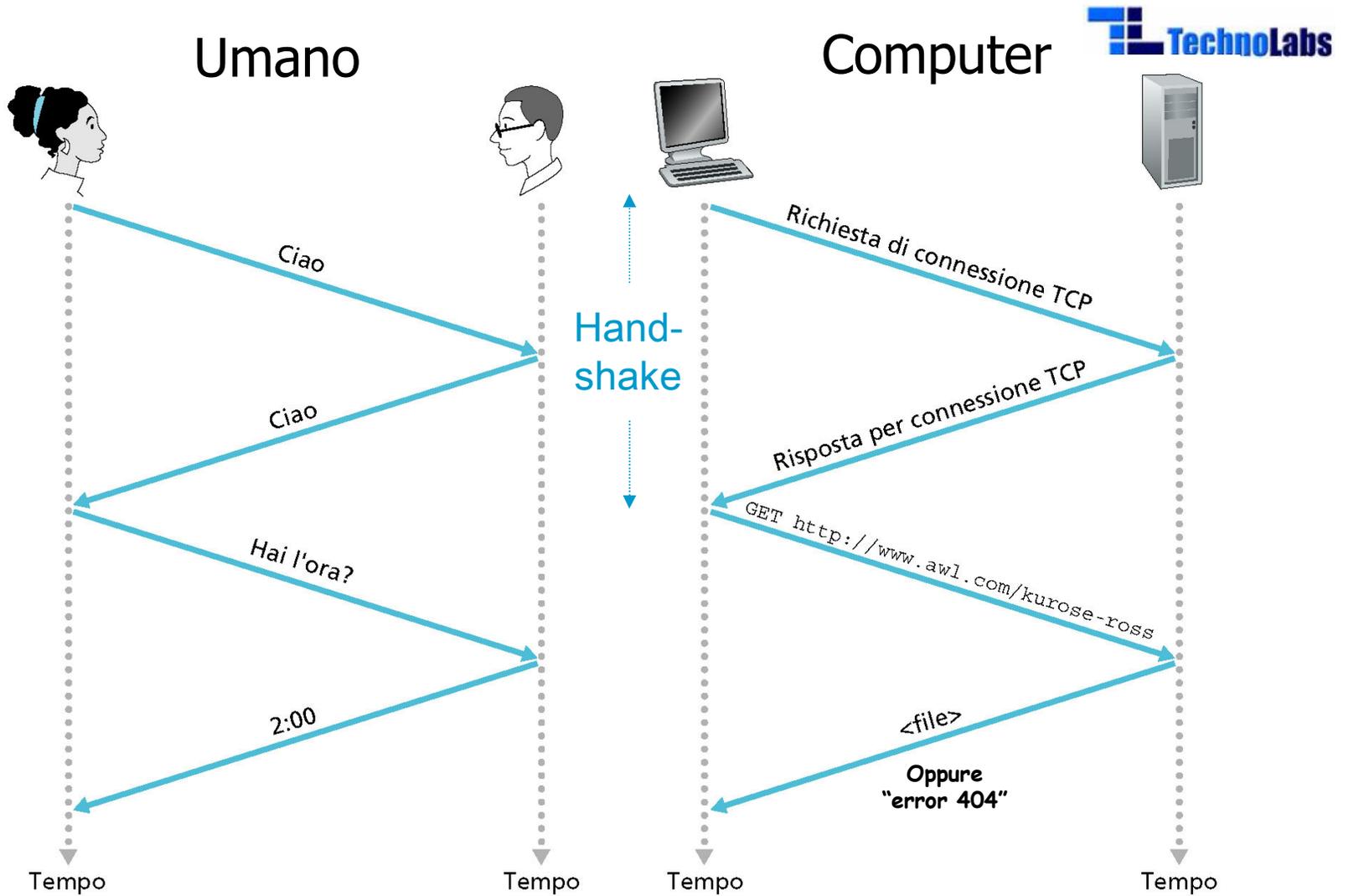


## Protocolli di rete:

- ☞ calcolatori invece che persone
- ☞ tutte le attività di comunicazione in Internet sono governate da protocolli

*Definizione: I protocolli sono un insieme formalizzato di regole che definiscono il **formato** e l'**ordine**, dei messaggi inviati e ricevuti tra entità della rete e le **azioni** che vengono fatte per la trasmissione e ricezione dei messaggi stessi*

# Esempi di protocollo



# Internet: edge

## 👉 Hosts (A.K.A. End-systems):

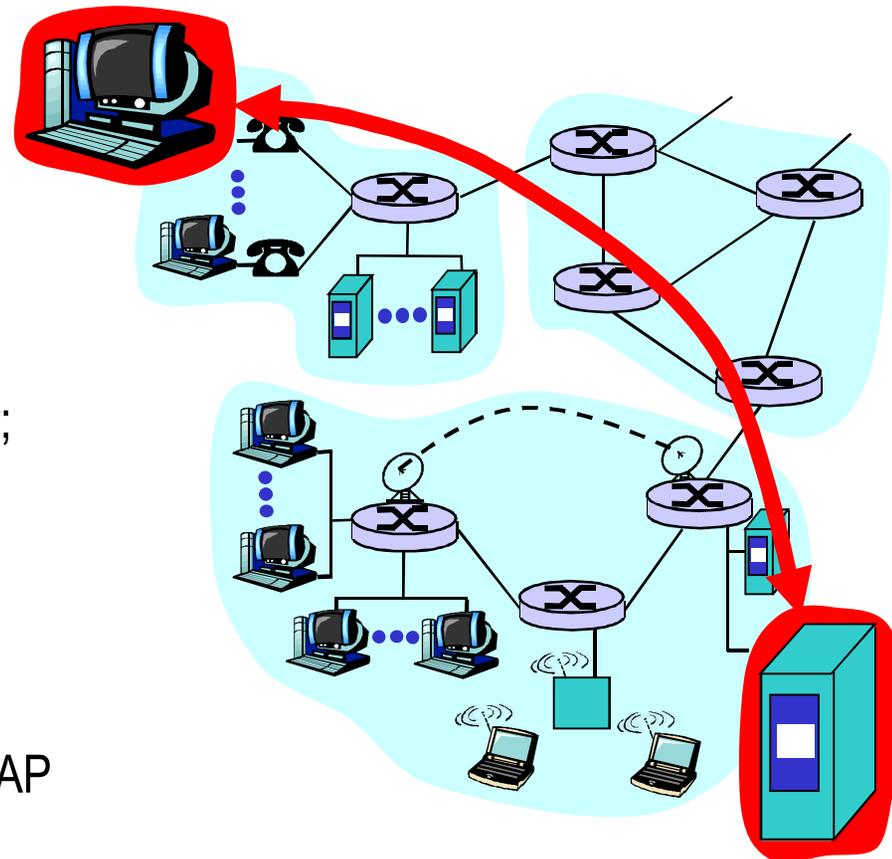
- ◆ Applicazioni e servizi di rete
- ◆ e.g., WWW, email

## 👉 Modello client/server

- ◆ Host (client) invia una richiesta di servizio, host (server) fornisce il servizio
- ◆ e.g., WWW client (browser)/ server; email client/server

## 👉 Modello peer2peer

- ◆ Non ci sono server dedicati
- ◆ L'interazione è simmetrica
- ◆ e.g.: FreeNet, GNUTELLA, OpenNAP



# Network edge: servizio orientato alla connessione



Obiettivo: trasferire dati tra end system.

☞ *handshaking*: scambio di informazione di controllo prima della comunicazione

- ◆ Hello, hello ( protocollo umano)
- ◆ *viene creato uno "stato"* nei due host che comunicano

☞ TCP - Transmission Control Protocol

- ◆ Servizio orientato alla connessione in Internet



## Servizio TCP [RFC 793]

☞ trasferimento *affidabile (reliable)* e *in ordine* di flussi di byte

- ◆ perdita: conferma (acknowledgement) e ritrasmissioni

☞ *controllo di flusso (flow control)*:

- ◆ il sender non "inonda" il receiver

☞ *Controllo della congestione (congestion control)*:

- ◆ Si diminuisce il ritmo (rate) di trasmissione se la rete è congestionata

# Network edge: servizio connectionless



Obiettivo: trasferimento dati tra host

- ◆ Lo stesso di prima!
- ☞ **UDP** - User Datagram Protocol [RFC 768]: il servizio connectionless di Internet
  - ◆ trasferimento dati non affidabile
  - ◆ no controllo di flusso
  - ◆ no controllo della congestione



Appl. che usano TCP:

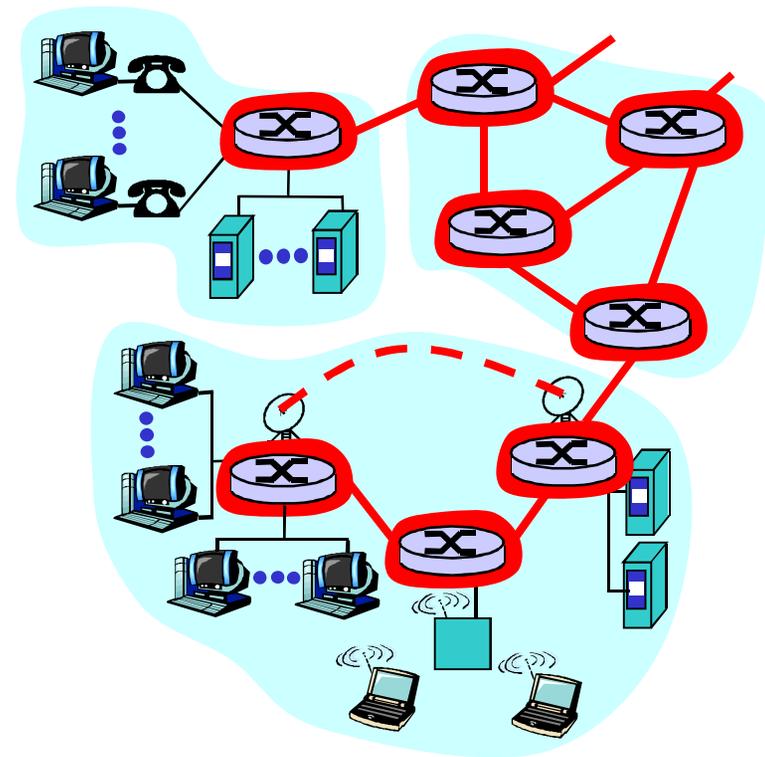
- ☞ HTTP (WWW), FTP (file transfer), Telnet (remote login), SMTP (email)

Appl. che usano UDP:

- ☞ streaming audio/video teleconferenza, telefonia su Internet, SNMP (management)

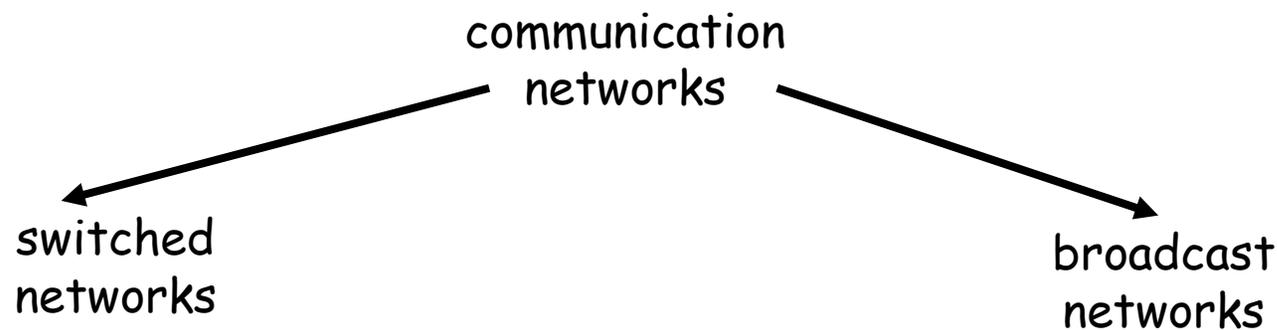
# Internet: il cuore della rete

- ☞ Ragnatela di router
- ☞ La domanda fondamentale: Come avviene il trasferimento dei dati nelle reti?
  - ◆ **Commutazione di circuito:** circuito dedicato per ogni chiamata (es. rete telefonica) le risorse sono assegnate (riservate) per tutta la durata della sessione
  - ◆ **Commutazione di pacchetto:** i dati sono inviati in rete scomponendoli in “pezzi” detti ‘pacchetti’ e le risorse di rete vengono assegnate su richiesta per cui può capitare che bisogna aspettare (analogia del ristorante) per accedere al link ovvero alla risorsa trasmissiva



# Tassonomia delle reti di Comunicazione

- ☞ Cerchiamo di catalogare una rete dal punto di vista delle problematiche di comunicazione e non solo dalla sua topologia (i.e. Internet)
- ☞ Una rete di comunicazione è stata vista come una struttura magliata (**mesh**) di router interconnessi
- ☞ **La questione è:** come vengono trasferiti i dati attraverso la rete ?



# Reti Broadcast vs reti Switched

## Reti Broadcast

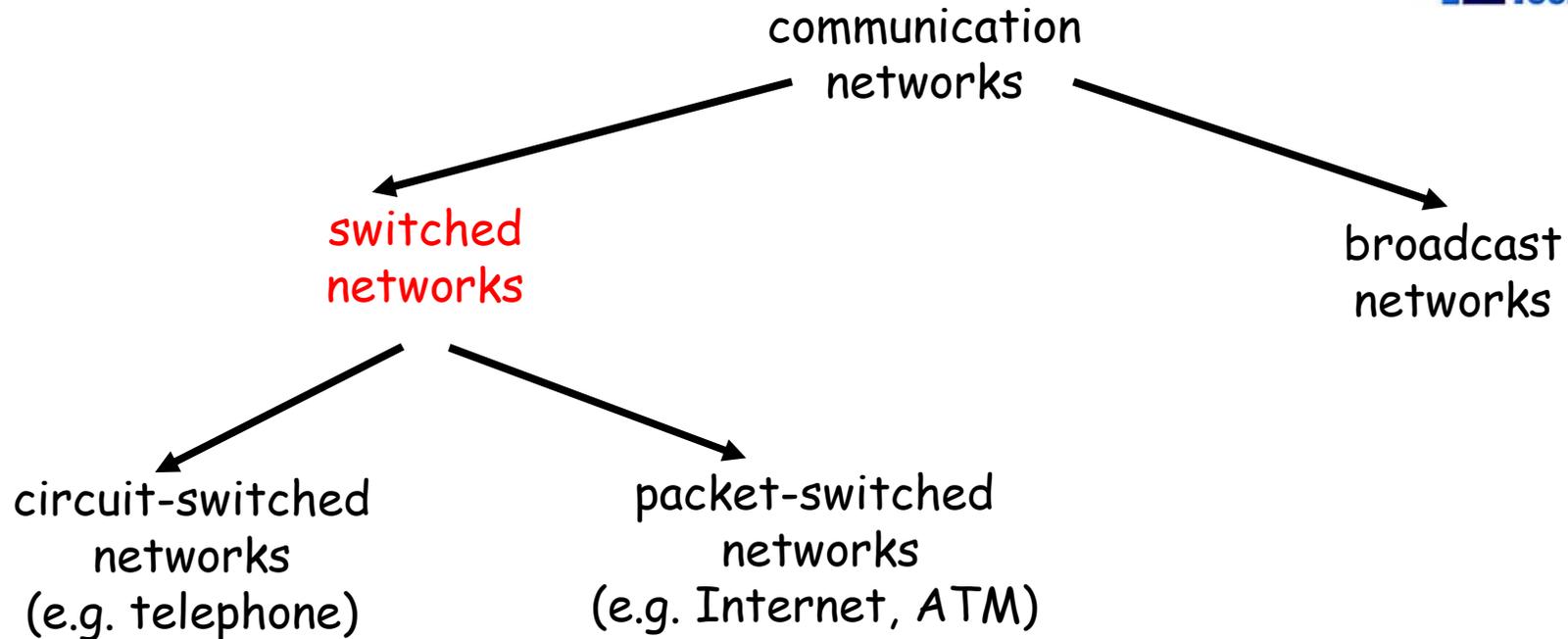
- ◆ I nodi condividono un canale comune; l'informazione trasmessa da un nodo è ricevuta da **tutti** gli altri nodi nella rete
- ◆ Esempio: TV, radio

## Reti Switched

- ◆ L'informazione è trasmessa a un piccolo sub-set (eventualmente solo uno) dei suddetti nodi



# Tassonomia delle reti switched



- ☞ **Switching di circuito:** un circuito dedicato per ogni chiamata:
  - ◆ e.g., telefono, Integrated Services Digital Networks (ISDN)

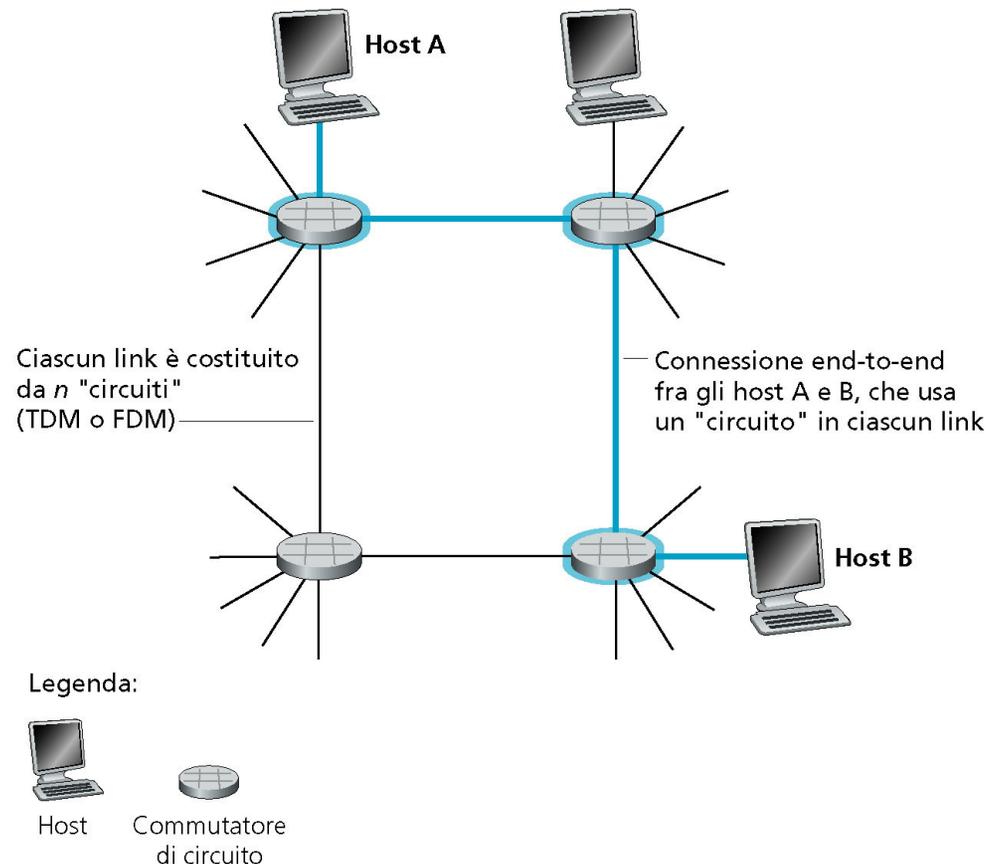


- ☞ **Switching di pacchetto:** i dati vengono spediti attraverso la rete in “atomi” discreti (“chunks”, ‘pacchetti’)

# Switching di circuito

Le risorse sono riservate  
End-to-end alla chiamata

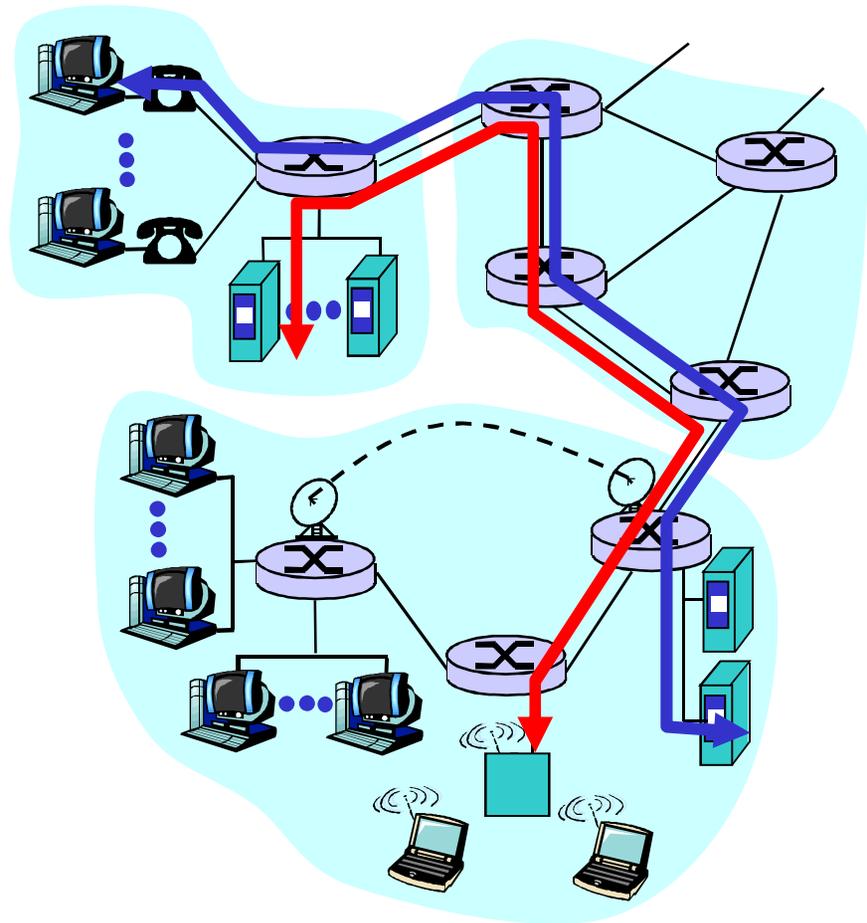
- ➔ Prestazioni garantite
- ➔ È richiesta la fase di “Call setup” (handshaking)
  - ◆ Le risorse sono assegnate alla chiamata
  - ◆ Le risorse sono inattive (*idle*) se non usate dal chiamante (*no sharing*)



# Commutazione di circuito

## Allocazione delle risorse per la gestione della chiamata

- Banda di trasmissione
- Risorse dedicate
- Performance elevata
- Necessita di una fase di inizializzazione
  - ◆ Creazione **circuito**



# Commutazione di circuito

Le risorse di comunicazione di rete (bandwidth) sono suddivise in “parti” allocate alle chiamate

➤ Una parte della risorsa rimane inattiva (*idle*) se non viene utilizzata (*no sharing*)

- Due soluzioni possibili
- ◆ frequency division (FDM)
  - ◆ time division (TDM)

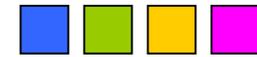


# Circuit Switching: FDMA and TDMA

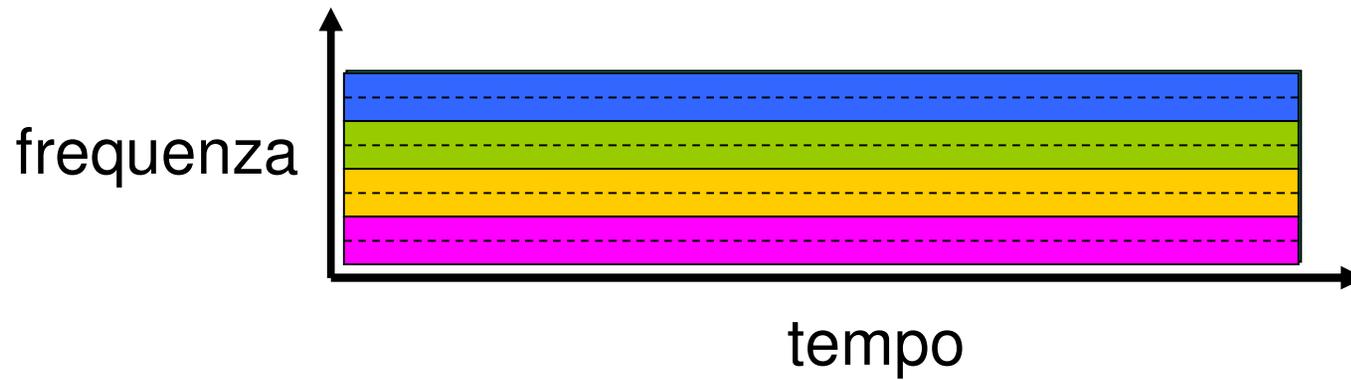
Esempio:



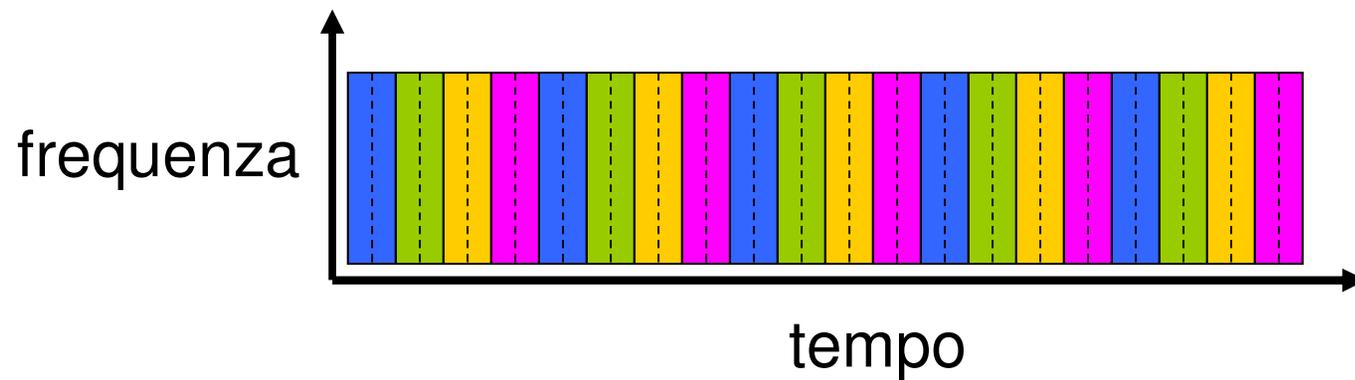
4 utenti



FDMA



TDMA



# Switching di pacchetto

Ogni flusso dati end-to-end è diviso in **pacchetti**

☞ Ogni Pacchetto ha la seguente struttura:



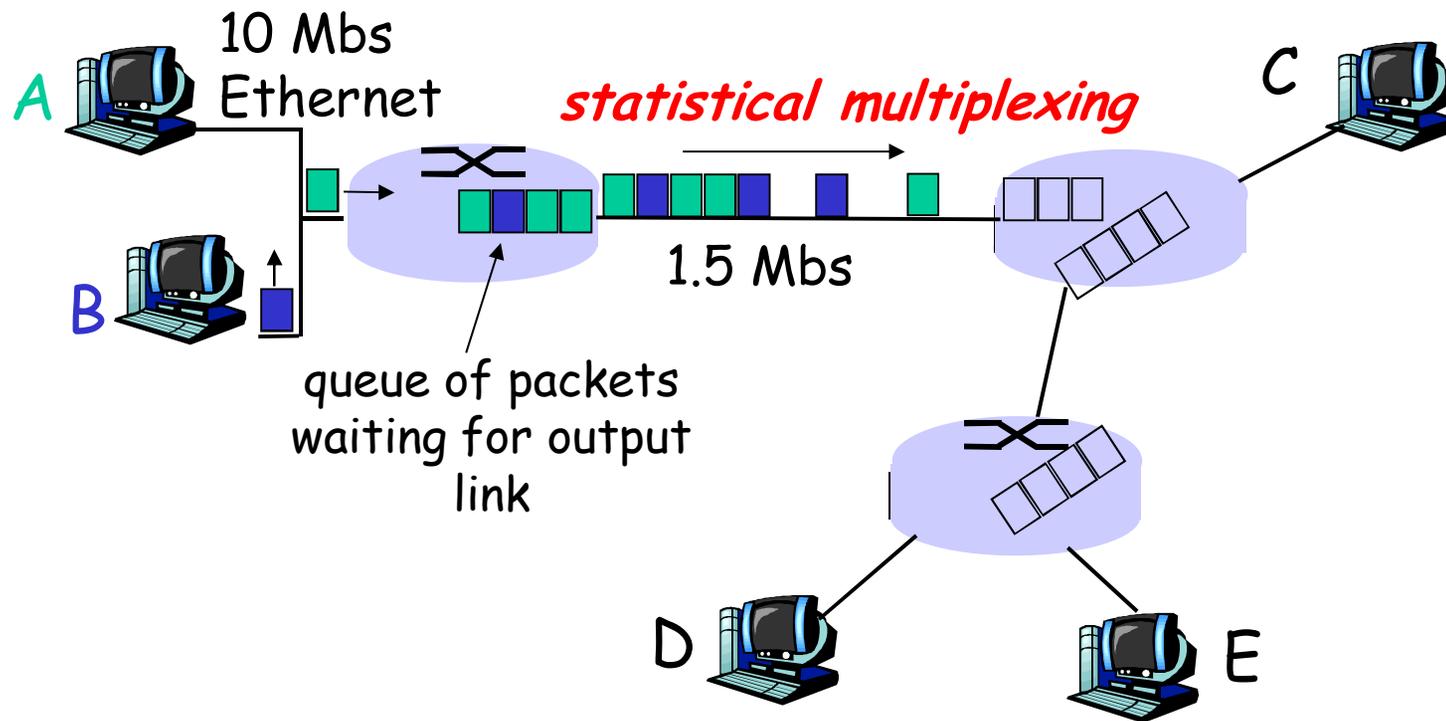
- ◆ Header e Trailer trasportano informazioni di controllo (e.g., indirizzo destinazione, check sum) (dov'è questa informazione nel caso del circuito?)

☞ Ad ogni nodo viene ricevuto l'intero pacchetto, memorizzato per un tempo breve, e poi spedito al prossimo nodo (**Store-and-Forward Networks**)



☞ Ogni pacchetto è inviato attraverso la rete da nodo a nodo seguendo dei cammini (**Routing**)

# Statistical Multiplexing



Le sequenze di pacchetti di A e B non hanno un pattern fisso → **statistical multiplexing**.



# Packet Switching

## Dati sono suddivisi in *packets*

- ☞ Packet degli utenti A e B condividono le risorse di rete
- ☞ ogni packet utilizza la banda al massimo della sua capacità
- ☞ “resources used as *needed*” . Non è necessaria una allocazione iniziale di tutte le risorse

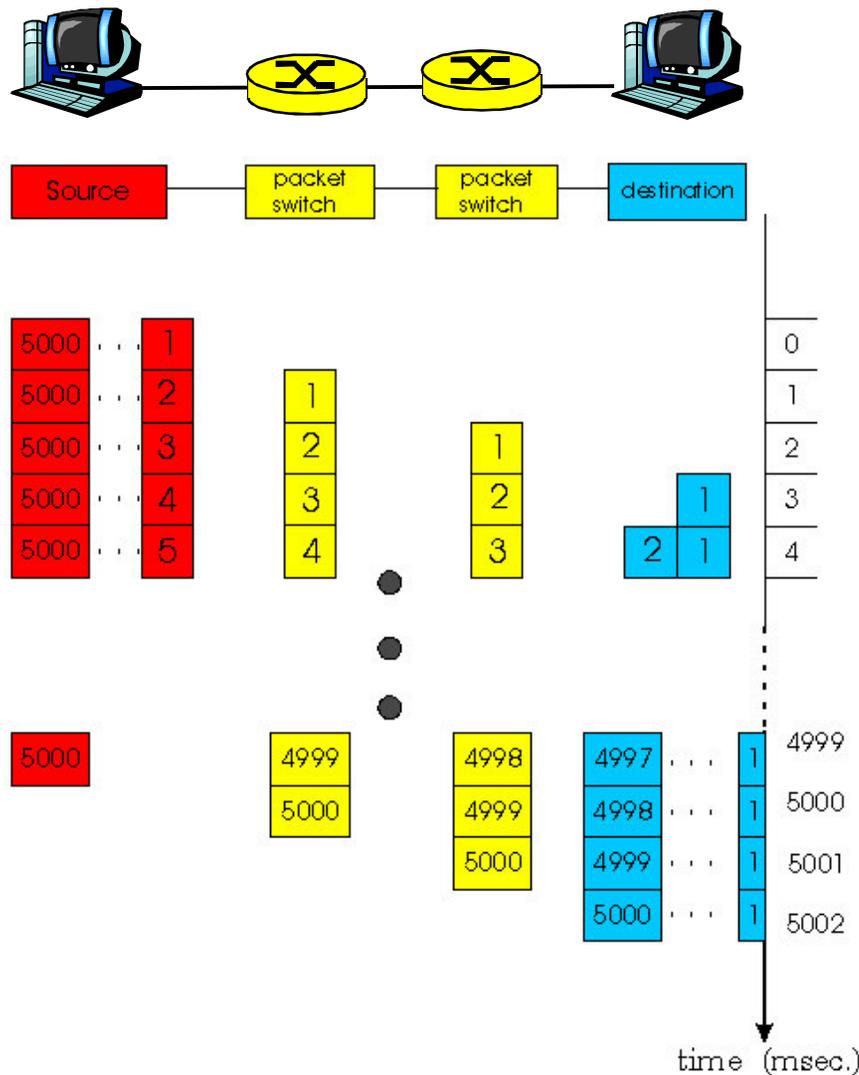


## Problematiche:

- ☞ Richiesta di risorse può essere superiore della disponibilità
- ☞ congestione: code
- ☞ “store and forward”: packet fanno un passo alla volta
  - ◆ Trasmissione su un link
  - ◆ Attesa al link successivo

Può tornare utile l'esempio del ristorante !

# Network Core: Packet Switching



## Esempio

- Messaggio di 7.5 Mbit
- Suddivisione in 5000 pacchetti da 1.5 Kbit
- Capacità dei link: 1.5 Mbps
- Tempi di processamento nei router trascurabili
- **Esercizio:** calcolare tempo di trasferimento se il messaggio non fosse diviso

Attenzione: 1 Mbit=1000 Kbit !!



# Switching di pacchetto

- ☞ Esistono due tipologie di reti a switching di pacchetto
  - ◆ **datagram network**
    - ◆ Ogni pacchetto di un flusso viene instradato (switched) **indipendentemente** dagli altri
  - ◆ **virtual circuit network:**
    - ◆ Tutti i pacchetti di un flusso vengono spediti su un cammino **pre-stabilito** (circuito virtuale).



# Datagram Packet Switching

- ☞ Esempio: reti IP
- ☞ Ogni pacchetto viene indirizzato indipendentemente
  - ◆ ogni pacchetto (**header**) contiene tutte le informazioni relative alla destinazione (**complete destination address**)
  - ◆ Ricevuto un pacchetto, un router guarda al campo “destination address” e cerca nella sua tabella di routing per determinare il prossimo passo (**next hop**)
  - ◆ I cammini possono cambiare durante la sessione
  - ◆ I routers non mantengono alcuna informazione sullo stato della sessione



# Virtual-Circuit Packet Switching

- Esempio: Rete Asynchronous Transfer Mode (ATM)
- Ibrido di circuito e datagram switching

- ◆ ogni pacchetto trasporta una **tag** (virtual-circuit VC#), la tag determina il next hop
- ◆ Un cammino fissato al *Virtual Circuit setup time*, rimane fissato per tutto il tempo
- ◆ I routers mantengono informazioni di stato per-flow

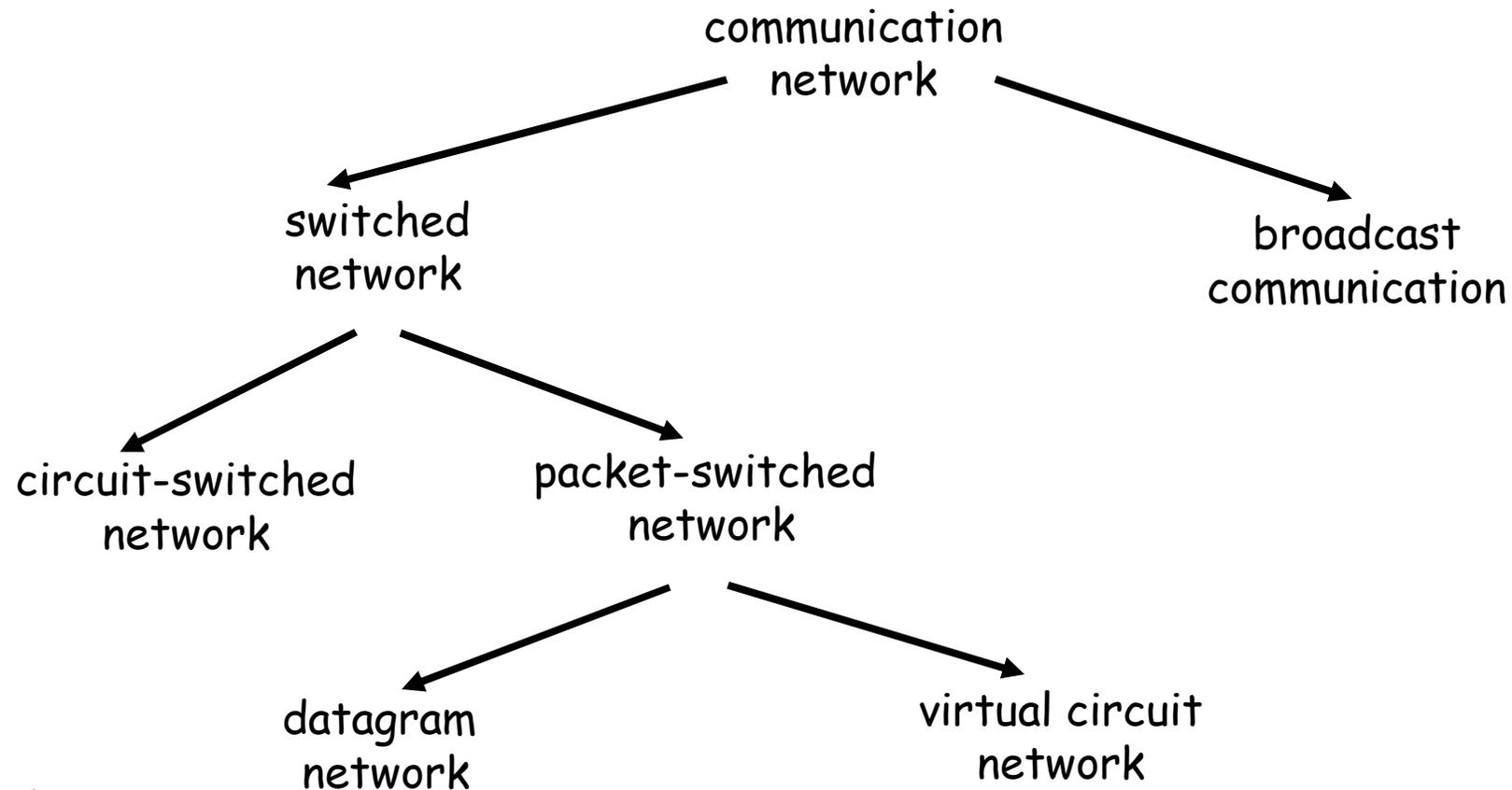
Incoming Interface	Incoming VC#	Outgoing Interface	Outgoing VC#
1	12	2	22
1	16	3	1
2	12	3	22
...			

- Quali vantaggi offrono i circuiti virtuali rispetto ai datagrammi ?

- ◆ Consegna dei pacchetti con **garantita integrità e in-sequence**
- ◆ Tuttavia: Pacchetti da **differenti** circuiti virtuali possono essere **interleaved**



# Tassonomia delle reti di Comunicazione

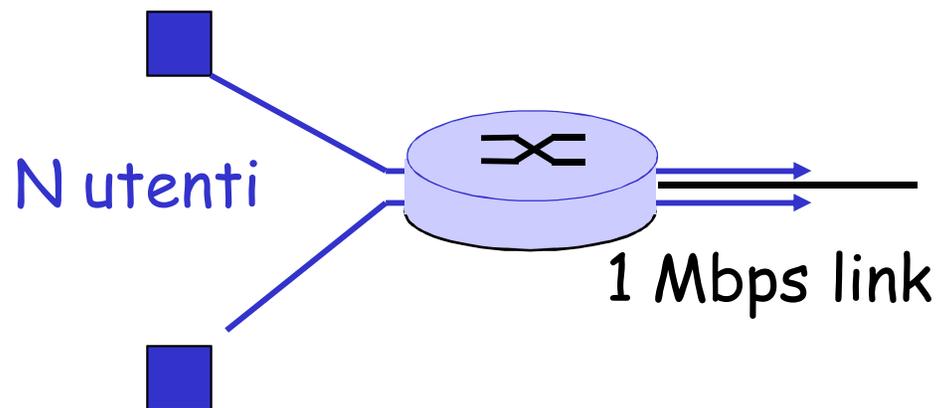


# Valutazione



La commutazione di pacchetto permette di avere un maggior numero di utenti a parità di risorse

- ☞ Collegamento ad 1 Mbit
- ☞ Utente generico:
  - ◆ 100Kbps se “attivo”
  - ◆ Attivo solo 10% del tempo
- ☞ Commutazione di circuito:
  - ◆ 10 utenti
- ☞ Commutazione di pacchetto:
  - ◆ 35 utenti, probabilità di avere un numero di utenti attivi maggiore di 10 è minore di 0.0017 (**provare**)



La tecnica Packet switching quasi sempre ha le stesse prestazioni della circuit-switching ma lo fa permettendo un numero di utenti 3 volte superiore

# Packet switching versus circuit switching



- ☞ Ottimo per dati a raffica (bursty)
  - ◆ Condivisione di risorse
  - ◆ Nessuna instaurazione di chiamata

Ma :

- ☞ **Possibilità di congestione:** ritardo e perdita di pacchetti
  - ◆ Servono protocolli per il trasporto affidabile e per gestire la congestione



# Packet-switched networks: instradamento (routing)



- ☞ **Obiettivo:** trasferire i pacchetti da sorgente a destinazione seguendo un cammino nella rete
  - ◆ Molti algoritmi di selezione dei cammini (cap. IV)
- ☞ **Reti a datagramma (datagram networks):**
  - ◆ Prossimo salto (hop) determinato dall'indirizzo di destinazione
  - ◆ Il percorso può mutare nel corso della sessione
  - ◆ analogia: servizio postale
- ☞ **Reti a circuito virtuale (virtual circuit networks):**
  - ◆ Ogni pacchetto contiene un identificatore che determina il prossimo salto
  - ◆ Il cammino è fissato una volta per tutte in fase di instaurazione
  - ◆ I router attraversati mantengono informazione su ogni chiamata



Attenzione: circuito virtuale e circuit switching sono cose diverse!!

# Overview



- Cosa è una rete
- Cosa è Internet
  - ◆ Componenti
    - ◆ Cosa è un protocollo
  - ◆ Servizi
    - ◆ Client/server e peer-to-peer
    - ◆ Connectionless e connection-oriented
- Network core
  - ◆ Circuit/packet switching
    - ◆ TDM/FDM
    - ◆ packet network e VC
- Perdite e ritardi in packet-switched network
- Struttura a livelli
- Internet structure e ISPs
- Accessi alla rete e mezzi fisici

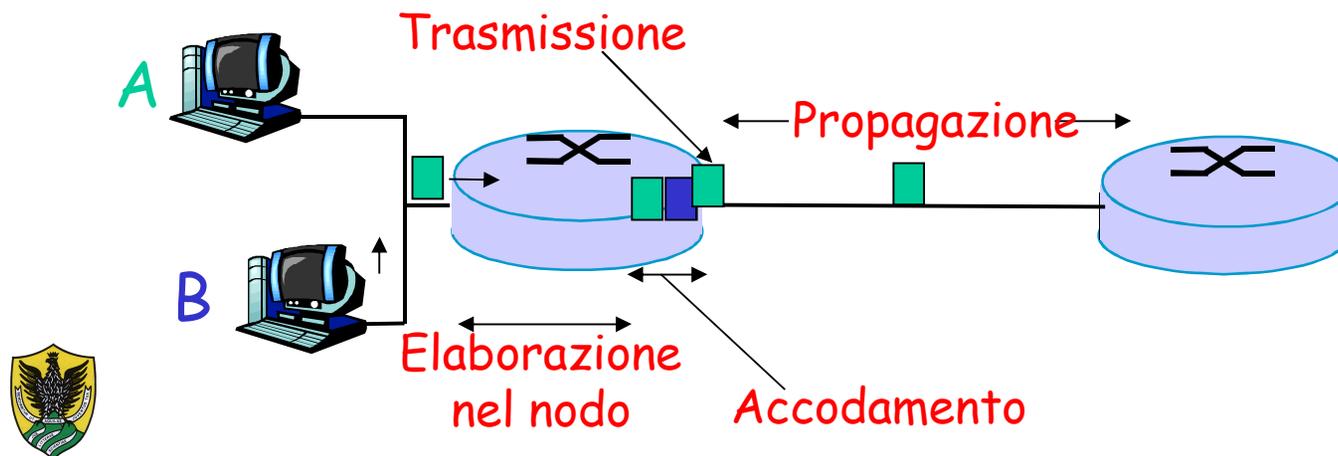


# Ritardo di trasferimento nelle reti a pacchetto



## Quattro sorgenti di ritardo ad ogni salto

1. **Processamento al nodo:**
  - ◆ Correzione di errore sui bit
  - ◆ calcolo link di uscita
2. **Attesa in coda**
  - ◆ Attesa per la trasmissione
  - ◆ Dipende dalla congestione nel router



# Ritardo di trasferimento nelle reti a pacchetto-2

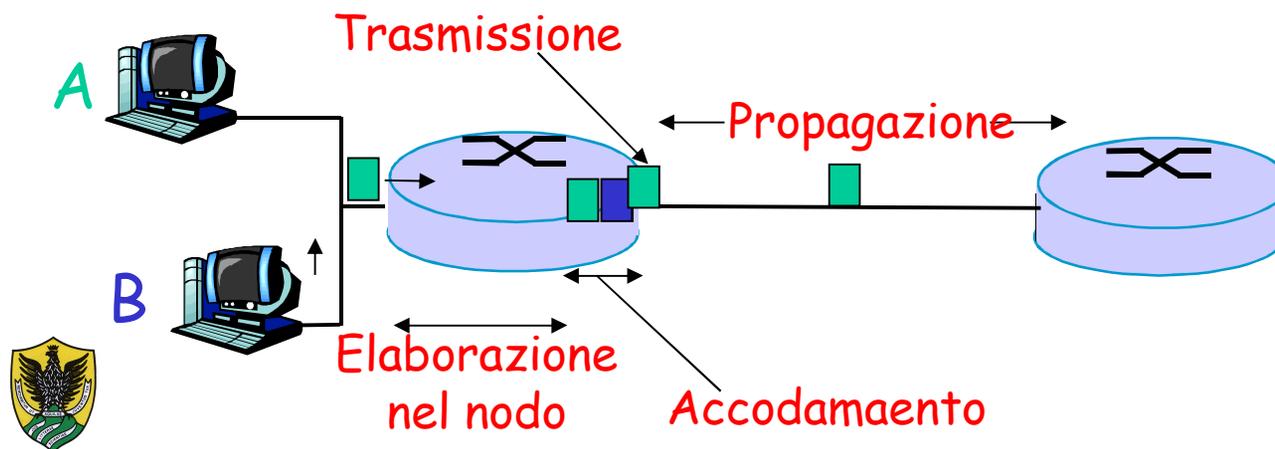
## 3. Ritardo di trasmissione

- ◆  $R$  = banda sul link (bps)
- ◆  $L$  = lunghezza pacchetto (bit)
- ◆ Tempo per trasmettere pacchetto sul link =  $L/R$

## 4. Ritardo di propagazione:

- ◆  $d$  = lunghezza link fisico
- ◆  $s$  = vel. propagazione nel mezzo ( $\sim 2 \times 10^8$  m/sec)
- ◆ Ritardo di propagazione nel mezzo =  $d/s$

**Attenzione:** 3 e 4 sono quantità diverse!

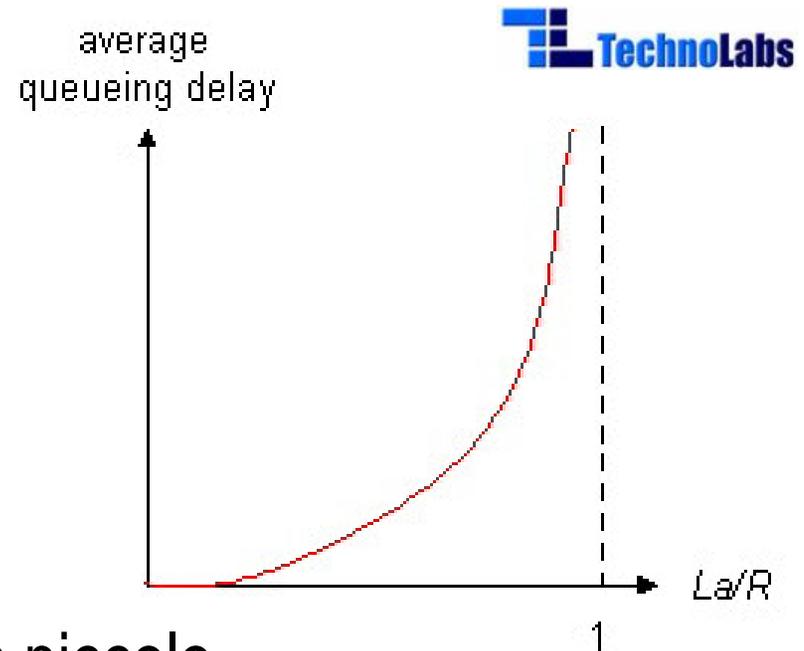


# Ritardo di coda nelle reti a pacchetto (3)

- ➡  $R$ =banda del link (bps)
- ➡  $L$ =lunghezza pacchetto (bit)
- ➡  $a$ =frequenza (rate) di arrivo dei pacchetti (packets/sec)

Intensità del traffico =  $\lambda a/R$

- ➡  $\lambda a/R \sim 0$ : ritardo medio di coda piccolo
- ➡  $\lambda a/R \rightarrow 1$ : ritardo medio di coda grande
- ➡  $\lambda a/R > 1$ : più pacchetti di quanti possano essere smaltiti, il tempo di attesa in coda tende a diventare infinito! (in realtà perdita)



# Ritardo in un Nodo

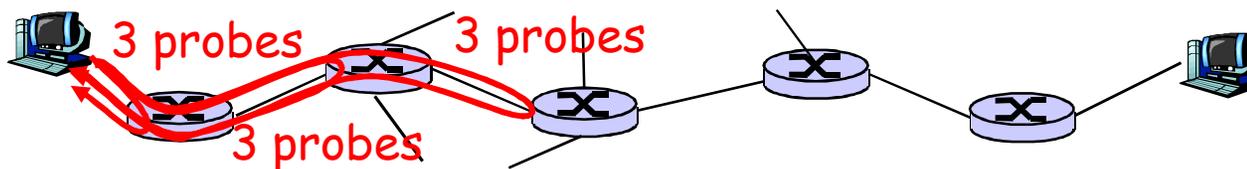
$$d_{\text{nodo}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

- ☞  $d_{\text{proc}}$  = processing delay
  - ◆ tipicamente qualche microsecondo o meno
- ☞  $d_{\text{queue}}$  = queuing delay
  - ◆ dipende dalla congestione
- ☞  $d_{\text{trans}}$  = transmission delay
  - ◆  $= L/R$ , significativo per links a bassa velocità
- ☞  $d_{\text{prop}}$  = propagation delay
  - ◆ Da qualche microsecondo fino a decine di msecs



# Cammini e ritardi nella “Rete Internet”

- ➡ Possiamo usare il ‘coltellino svizzero’ **TRACEROUTE**
- ➡ Il programma traceroute\* fornisce una misura di ritardo dalla sorgente al prossimo router lungo il cammino end-to-end fino alla destinazione finale.
  - ◆ Per ogni router  $i$ :
    - ◆ Invia 3 pacchetti verso il router  $i$  sul cammino verso la destinazione
    - ◆ Il router  $i$  ritornerà i pacchetti al mittente
    - ◆ L’End-System mittente calcola i tempi di andata-ritorno.



\* Nota: In windows questo comando prende il nome tracert



# Cammini e ritardi nella “Rete Internet”



traceroute: gaia.cs.umass.edu to www.eurecom.fr

3 misure di ritardo da  
gaia.cs.umass.edu a cs-gw.cs.umass.edu

1	cs-gw (128.119.240.254)	1 ms	1 ms	2 ms
2	border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)	1 ms	1 ms	2 ms
3	cht-vbns.gw.umass.edu (128.119.3.130)	6 ms	5 ms	5 ms
4	jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)	16 ms	11 ms	13 ms
5	jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)	21 ms	18 ms	18 ms
6	abilene-vbns.abilene.ucaid.edu (198.32.11.9)	22 ms	18 ms	22 ms
7	nycm-wash.abilene.ucaid.edu (198.32.8.46)	22 ms	22 ms	22 ms
8	62.40.103.253 (62.40.103.253)	104 ms	109 ms	106 ms
9	de2-1.de1.de.geant.net (62.40.96.129)	109 ms	102 ms	104 ms
10	de.fr1.fr.geant.net (62.40.96.50)	113 ms	121 ms	114 ms
11	renater-gw.fr1.fr.geant.net (62.40.103.54)	112 ms	114 ms	112 ms
12	nio-n2.cssi.renater.fr (193.51.206.13)	111 ms	114 ms	116 ms
13	nice.cssi.renater.fr (195.220.98.102)	123 ms	125 ms	124 ms
14	r3t2-nice.cssi.renater.fr (195.220.98.110)	126 ms	126 ms	124 ms
15	eurecom-valbonne.r3t2.ft.net (193.48.50.54)	135 ms	128 ms	133 ms
16	194.214.211.25 (194.214.211.25)	126 ms	128 ms	126 ms
17	* * *			
18	* * *			
19	fantasia.eurecom.fr (193.55.113.142)	132 ms	128 ms	136 ms

trans-oceanic link

\* Significa no reponse (probe perso, router not replying)



# Testare la Connettività con Ping



- ➔ Ping è un programma utile per verificare l'installazione corretta del TCP/IP.
- ➔ Funziona inviando pacchetti multipli IP ad una specificata destinazione.
- ➔ Ogni pacchetto inviato è una richiesta di una 'reply' (echo) Ogni 'reply' contiene la percentuale di successi e il round-trip time per la destinazione.

```
C:\WINNT\System32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
<C> Copyright 1985-2000 Microsoft Corp.

C:\> ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<10ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Recieved = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
```



# Perdita di Pacchetti



- ☞ Code (aka buffer): ai link di uscita sono associati uno o più buffer che hanno finita capacità
- ☞ Quando un pacchetto arriva ad una coda piena, il pacchetto viene buttato (lost)
- ☞ I pacchetti persi possono essere ritrasmessi dal nodo precedente, dalla sorgente (end system), o non essere ritrasmessi affatto



# Overview



- Cosa è una rete
- Cosa è Internet
  - ◆ Componenti
    - ◆ Cosa è un protocollo
  - ◆ Servizi
    - ◆ Client/server e peer-to-peer
    - ◆ Connectionless e connection-oriented
- Network core
  - ◆ Circuit/packet switching
    - ◆ TDM/FDM
    - ◆ packet network e VC
- Perdite e ritardi in packet-switched network
- **Struttura a livelli**
- Internet structure and ISPs
- Accessi alla rete e mezzi fisici



# OSI (Open System Interconnect) Reference Model

Il modello di  
riferimento che regola  
le comunicazioni di  
rete



# Stratificazione protocollare (Protocol “Layering”)



## Le reti sono complesse!

☞ Molti elementi:

- ◆ host
- ◆ router
- ◆ link fisici dalle caratteristiche diverse
- ◆ applicazioni
- ◆ protocolli
- ◆ hardware, software

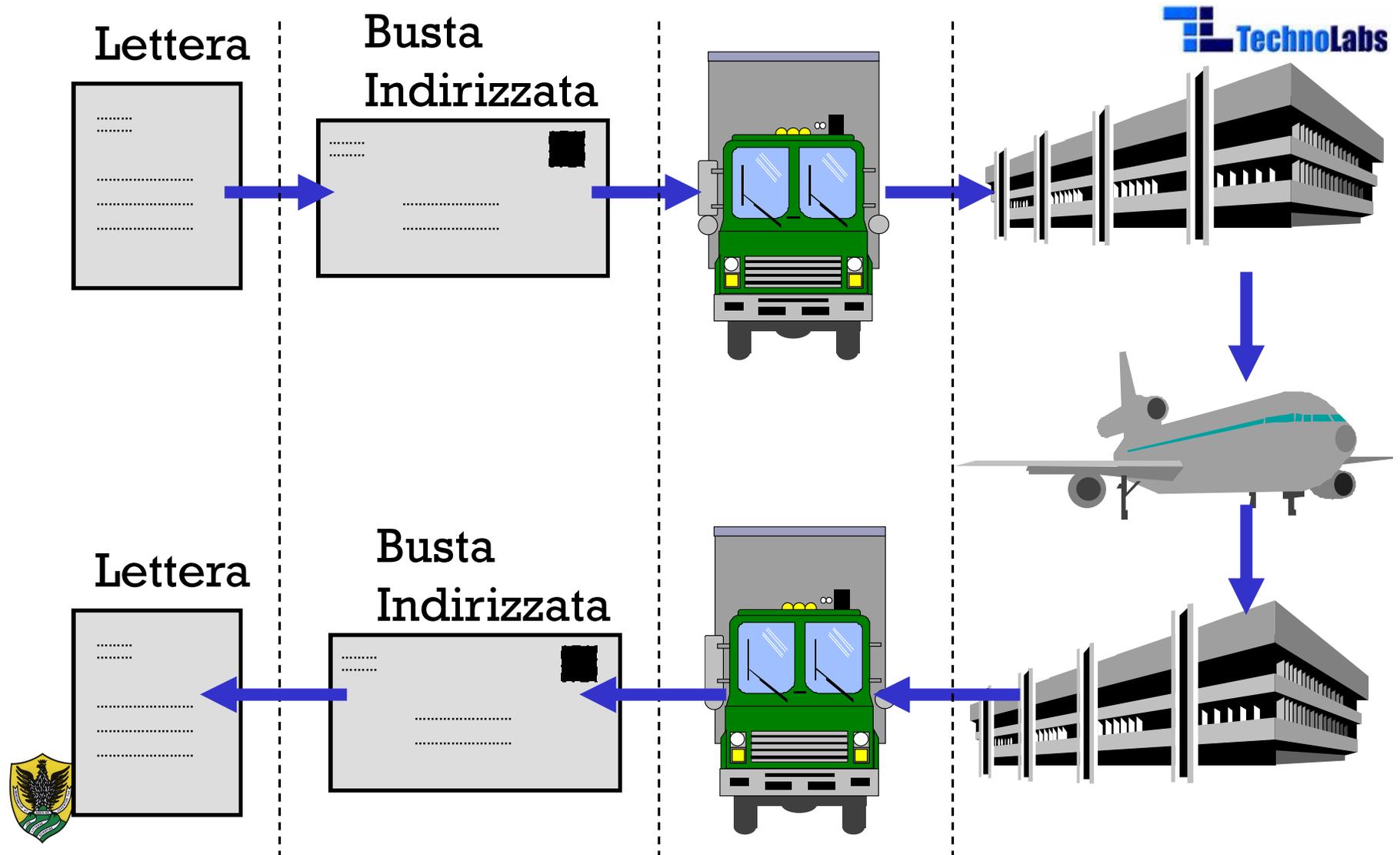


## Domanda:

Come organizzare la struttura della rete?

O almeno la sua discussione?

# Analogia Postale



# Esempio di stratificazione: servizio postale



➔ Una sequenza di passi

# Servizio postale: una prospettiva diversa



Sorgente

Destinazione

Lettera (consegna)	Lettera (cons. a dest.)
Controllo destinazione (consegna a corriere)	Controllo destinazione (ric. da corriere)
Corriere	Corriere (consegna)
Trasporto (aereo, treno...)	

**Strati:** ogni strato implementa un servizio

- ◆ Mediante funzionalità interne
- ◆ Usando i servizi messi a disposizione dagli strati inferiori



# Stratificazione del servizio postale



Consegna da mittente a destinatario

Consegna da ufficio postale sorgente a ufficio postale di destinazione

Trasferimento da sede locale a sede destinazione del corriere

Trasporto



# Perché la stratificazione?



I sistemi sono complessi:

- ☞ La stratificazione permette una più facile organizzazione e individuazione delle funzionalità
- ☞ La modularità facilita la manutenzione e la modifica dei sistemi
  - ◆ La modifica **dell'implementazione** dei servizi resi da uno strato è **trasparente**, purchè non si modifichi **l'interfaccia**
  - ◆ Es., cambiare il corriere non altera il funzionamento complessivo del servizio postale
- ☞ Ogni livello espone le interfacce ma nasconde la sua implementazione



## Separation of Concern

Separazione degli interessi e delle responsabilità, fare ciò che compete, delegando ad altri tutto ciò che è delegabile

## Information Hiding

Nascondere tutte le informazioni che non sono indispensabili a che il committente possa compiutamente definire l'operazione



# ISO OSI reference model

- ☞ Un set di protocolli è **aperto** se
  - ◆ i dettagli del protocollo sono disponibili pubblicamente
  - ◆ i cambiamenti sono gestiti da un'organizzazione la cui partecipazione è aperta al pubblico
  
- ☞ Un sistema che implementa protocolli aperti è un **sistema aperto** (open system)
- ☞ L'International Organization for Standards (ISO) prescrive uno standard per l'interconnessione di sistemi aperti
  - ◆ open system interconnect (OSI)
- ☞ OSI ha molto influenzato il modo di pensare ai protocolli stratificati



# ISO OSI



## *Reference model*

- ◆ definisce formalmente cosa si intende per strato (layer) e servizio etc.

## *Service architecture*

- ◆ descrive il servizio fornito da ogni layer ed il punto di accesso al servizio (SAP: Service Access Point)

## *Architettura protocollare*

- ◆ insieme di regole che implementano l'architettura del servizio
- ◆ architetture di servizio diverse ma compatibili possono usare architetture protocollari non compatibili



# Modello ISO/OSI

Nel modello OSI sono previsti 7 livelli

☞ **3 LIVELLI DI RETE**

- ◆ fisico,
- ◆ link,
- ◆ rete

☞ **4 LIVELLI DI UTENTE**

- ◆ trasporto,
- ◆ sessione,
- ◆ presentazione,
- ◆ applicazione



# ISO: il modello di riferimento OSI

- ◆ ISO:  
International  
Standard  
Organization
- ◆ OSI:  
Open Systems  
Interconnection

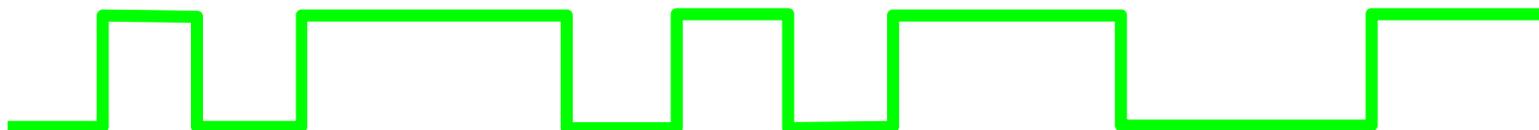


# LIVELLO FISICO



- ☞ **Compito:** trasmettere sequenze binarie (bit) sul canale trasmissivo
- ☞ **Lo standard prescrive**
  - ◆ le caratteristiche fisiche dell'interfaccia e del mezzo
  - ◆ codifica per rappresentare i bit
  - ◆ frequenza dei dati
  - ◆ configurazione della linea
  - ◆ topologia
  - ◆ modalità di trasmissione (HD, FD, S)
  - ◆ la forma e la dimensione dei connettori
  - ◆ la sincronizzazione dei bit

Applicazione
Presentazione
Sessione
Trasporto
Rete
Dati
Fisico



# LIVELLO DATI (Datalink)

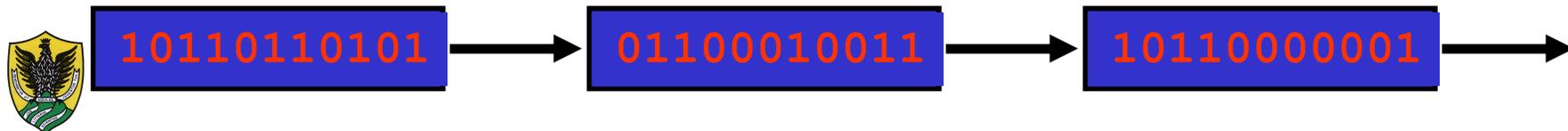


## 👉 **Compito:**

- ◆ creare trame (*frame*) e trasmetterle con "sufficiente" affidabilità tra due entità direttamente connesse, rilevare errori di trasmissione ed eventualmente correggerli
  - ◆ smistamento
  - ◆ controllo flusso
  - ◆ controllo errore
  - ◆ controllo accesso

## 👉 Rilevazione e correzione errori

- ◆ codici autocorreggenti, ritrasmissione, etc.



# LIVELLO RETE



- ☞ **Compito:** gestire l'instradamento (routing) di trame attraverso sistemi intermedi, ed eventualmente trovare percorsi alternativi in caso di guasti
- ☞ Indirizzamento
  - ◆ logico
- ☞ Algoritmi di instradamento
  - ◆ definizione e/o apprendimento (completo o parziale) della topologia della rete, calcolo del percorso su base locale e/o globale, riconfigurazione in caso di guasti, etc.
- ☞ Non necessariamente garantisce
  - ◆ affidabilità della trasmissione delle trame, non duplicazione alla destinazione, rispetto alla destinazione del loro ordine di invio
- ☞ È responsabile della consegna mittente-destinatario di un pacchetto



Applicazione
Presentazione
Sessione
Trasporto
<b>Rete</b>
Dati
Fisico

# LIVELLO TRASPORTO



- ☞ **Compito:** trasferire l'informazione (il messaggio) *end-to-end* affidabilmente e trasparentemente, ottimizzando l'uso delle risorse
- ☞ **Affidabilità'**
  - ◆ tutte le trame arrivano a destinazione, in copia unica e in ordine
- ☞ **Trasparenza**
  - ◆ "forma" dell'informazione qual era alla sorgente conservata a destinazione
- ☞ **Ottimizzazione**
  - ◆ traffico ripartito sui canali disponibili, prevenzione della congestione della rete
- ☞ **Funzioni:** segmentazione, riassettaggio, controllo della connessione, controllo flusso, controllo errore



# LIVELLO SESSIONE



☞ **Compito:** gestire il dialogo *end-to-end* tra due programmi applicativi che debbono comunicare

☞ Dialogo

- ◆ garantire la mutua esclusione nell'utilizzo di risorse condivise, intercalare domande e risposte garantendo la consequenzialita'

☞ Sincronizzazione

- ◆ stabilire punti intermedi nella comunicazione rispetto ai quali entrambe le parti abbiano la garanzia che quanto accaduto "prima" sia andato a buon fine



[\*] Many protocol suites do not include a session layer



# LIVELLO PRESENTAZIONE



☞ **Compito:** gestire la sintassi dell'informazione lungo l'intero percorso *end-to-end*, convertendo l'uno nell'altro i vari formati

- ◆ traslazione,
- ◆ crittografia,
- ◆ compressione



Many protocol suites do not include a Presentation Layer.

# LIVELLO APPLICAZIONE



- ☞ **Compito**: definire i servizi attraverso cui l'utente utilizza la rete, con tutte le relative interfacce di accesso
- ☞ Servizi di utente
  - ◆ terminale virtuale, trasferimento di file, posta elettronica, servizi di directory, etc.
- ☞ Servizi di sistema operativo
  - ◆ risoluzione di nomi, localizzazione di risorse, sincronizzazione degli orologi tra sistemi diversi, controllo di diritti di accesso, etc.

Applicazione
Presentazione
Sessione
Trasporto
Rete
Dati
Fisico



# Riepilogando



- ☞ Data-Link: comunicazione tra macchine sulla stessa rete.
- ☞ Network: comunicazione tra macchine su reti differenti.
- ☞ Trasporto: comunicazione tra processi (su macchine ubicate su reti differenti).

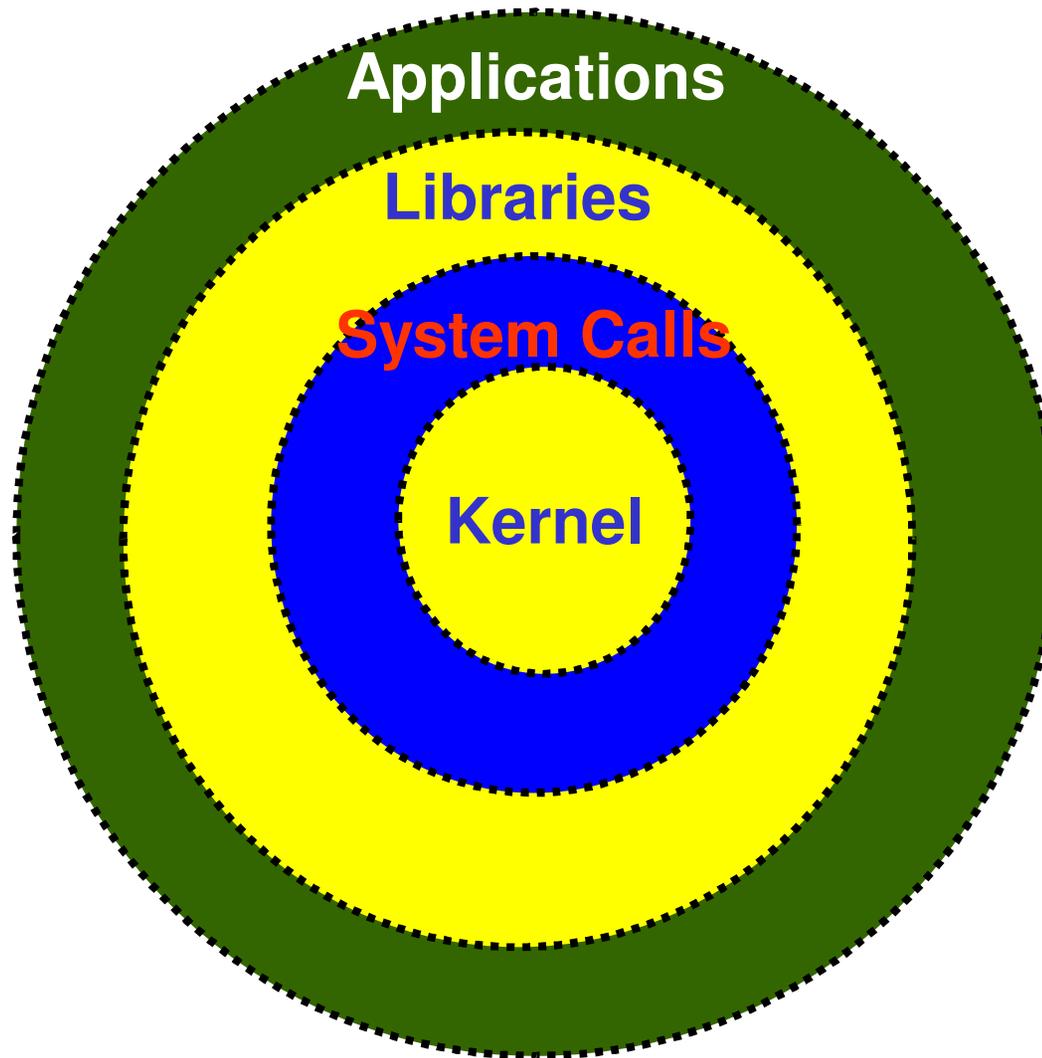


# Ricordare i livelli OSI

❖ <b>A</b>	7	Applicazione
❖ <b>P</b> arigi	6	Presentazione
❖ <b>S</b> i	5	Sessione
❖ <b>T</b> rovano	4	Trasporto
❖ <b>R</b> istoranti	3	Rete
❖ <b>D</b> a	2	Data Link
❖ <b>F</b> avola	1	Fisico



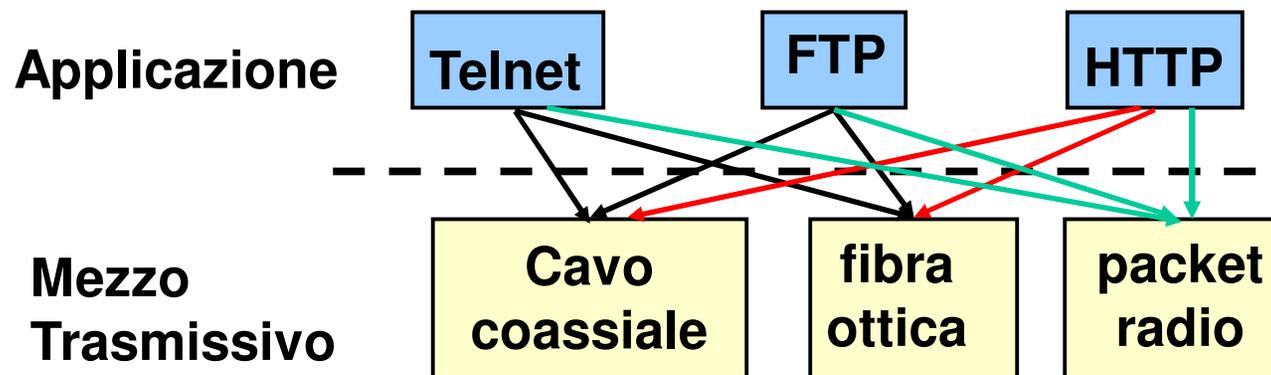
# Perché strutturare a livelli ?



Il sistema operativo UNIX è layered



# Un Esempio: No Layering

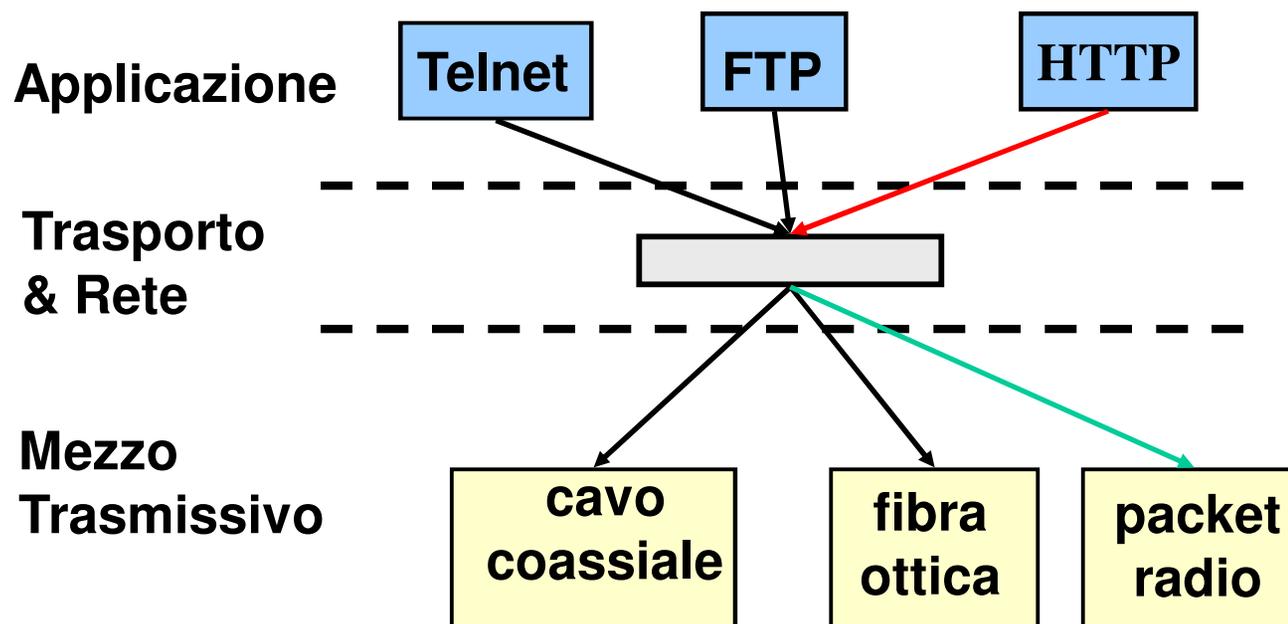


- ☞ In un approccio 'No layered': ogni nuova applicazione deve essere **ri-scritta** per ogni nuova tecnologia trasmissiva !



# Un Esempio: Benefici del Layering

- ☞ Soluzione: introdurre un livello intermedio che fornisca una astrazione **comune** per le varie tecnologie



# Quale funzionalità mettere in ogni layer ?



- ☞ Se l'applicazione può implementare una certa funzionalità, fatelo e non demandatelo ad un livello più basso – l'applicazione sa meglio di chiunque altro ciò di cui ha bisogno
- ◆ Aggiungere funzionalità in livelli più bassi se e solo se:
  - 1) È usata per migliorare le prestazioni di un gran numero di (correnti e future) applicazioni, e
  - 2) Non interferisce (troppo) con altre applicazioni



Success story: Internet



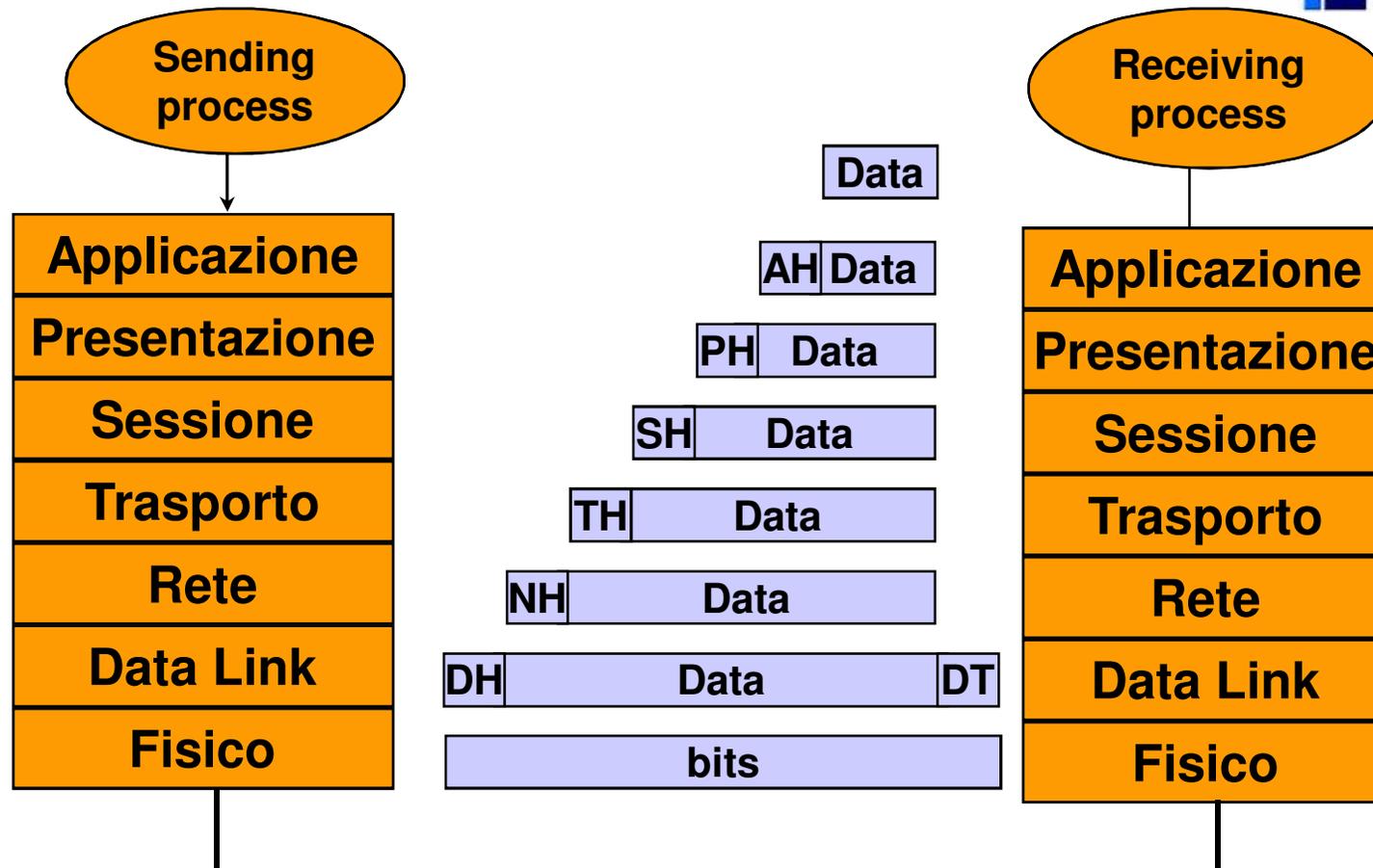
# Layering : Regola aurea



- ➔ Implementare una funzionalità ad un livello più basso deve avere un impatto **minimo** (in termini di prestazioni) su quelle applicazioni che non fanno uso di quella specifica funzionalità



# Trasmissione dei dati



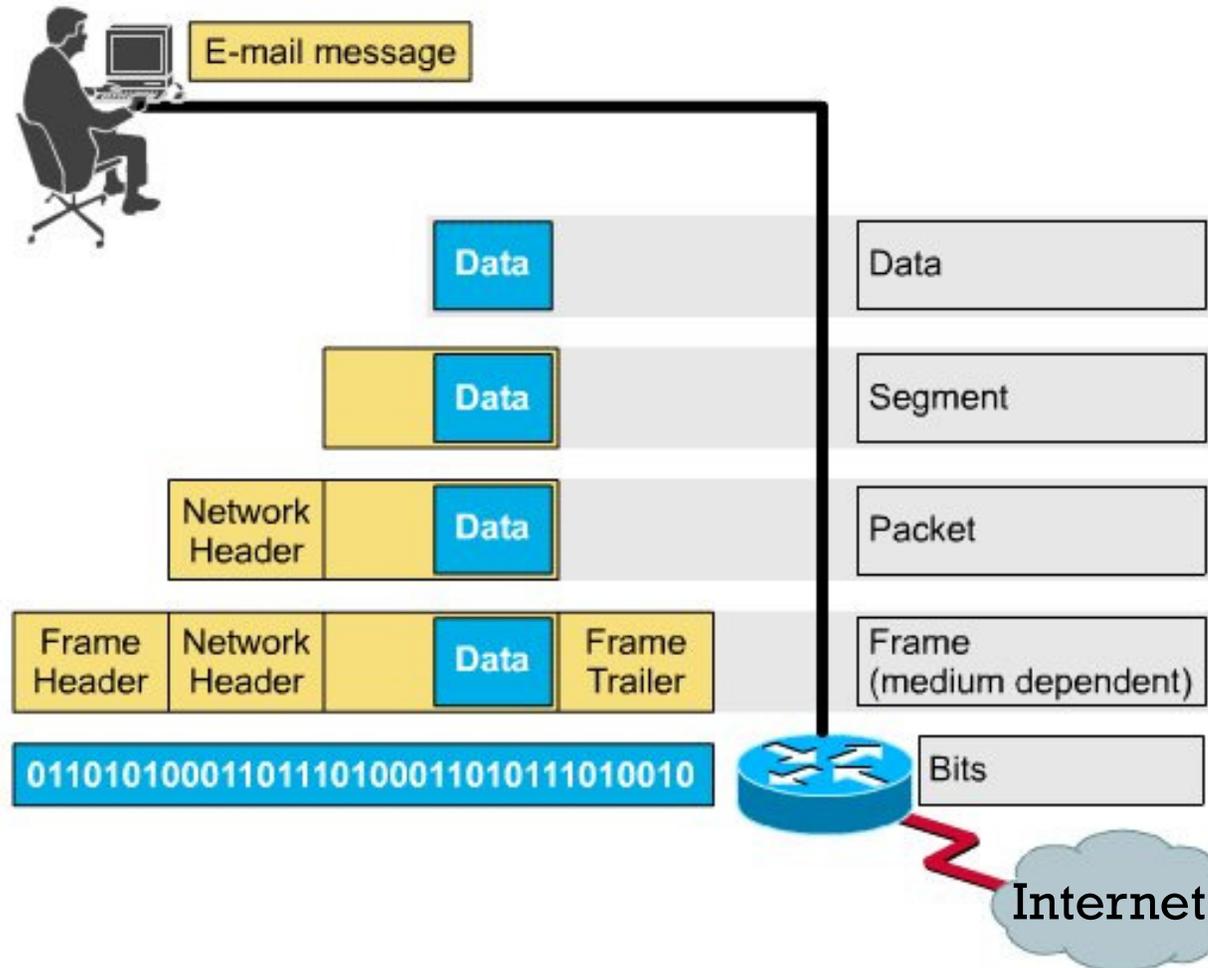
# Data Encapsulation



- Data Encapsulation è il processo di aggiungere un header ai dati che fluiscono giù per il modello OSI.
- Ogni livello OSI può aggiungere il proprio header ai dati ricevuti dal livello superiore (o dal programma software 'sopra' il livello Applicazione).
- I 5 Steps di Data Encapsulation sono:
  - ◆ I livelli Application, Presentation e Session creano DATA dall'input degli utenti.
  - ◆ Il livello Transport converte i DATA in SEGMENTS
  - ◆ Il livello Network converte i SEGMENTS in PACKETS (o datagrams)
  - ◆ Il livello Data Link converte i PACKETS in FRAMES
  - ◆ Il livello Physical converte i FRAMES in BITS.
- Lato TX l'informazione fluisce dall'alto in basso ed ogni livello spezzetta l'informazione ricevuta dal livello superiore e aggiunge l'header.
- Lato RX l'informazione fluisce verso l'alto del modello strippando l'header ad ogni livello e ri assemblando i pezzi di informazione.



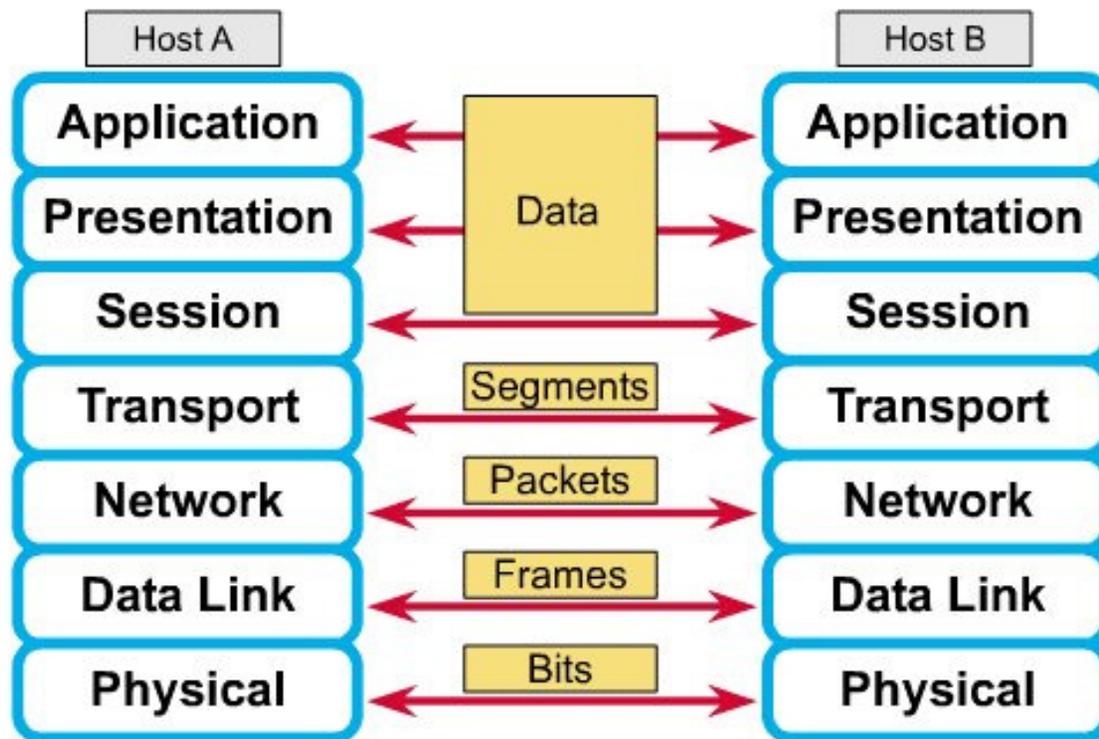
# Processo di Data Encapsulation



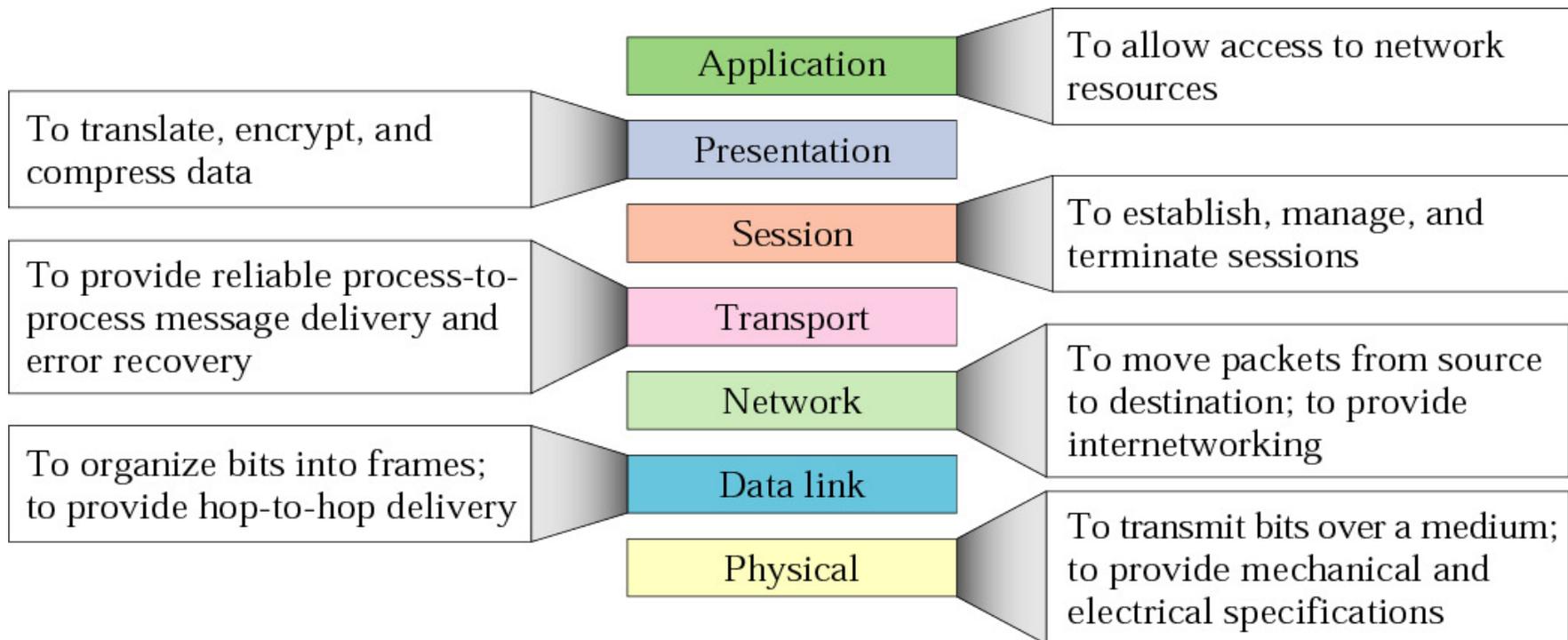
# Cosa sono le PDU's ?

Ogni livello contiene una Protocol Data Unit (PDU).

Le PDU's sono usate per la conversazione peer-to-peer.



# Summary of layers



# Perché 7 livelli?



- È necessario una cima ed un fondo -- 2
- è necessario nascondere il collegamento fisico, dunque c'è bisogno di un datalink -- 3
- sono necessarie sia azioni end-to-end che nodo-nodo; così sono necessari almeno sia un livello di rete che uno di trasporto -- 5
- sessione e presentazione non sono così importanti e sono spesso ignorati
- così sono necessari almeno 5 livelli, e 7 sembrano eccessivi
- ... e se lasciassimo fuori il livello fisico?



# Pila protocollare di Internet



- ➔ **applicazione**: supporta le applicazioni di rete (qui si gestisce sessione e presentazione)
  - ◆ ftp, smtp, http
- ➔ **trasporto**: trasferimento dati host-host
  - ◆ tcp, udp
- ➔ **rete (Internet)**: instradamento dei datagrammi dalla sorgente alla destinazione
  - ◆ IP, ICMP
- ➔ **link (accesso alla rete)**: trasferimento dati tra elementi di rete vicini
  - ◆ ppp, ethernet, ... qualunque cosa



# Il modello TCP/IP

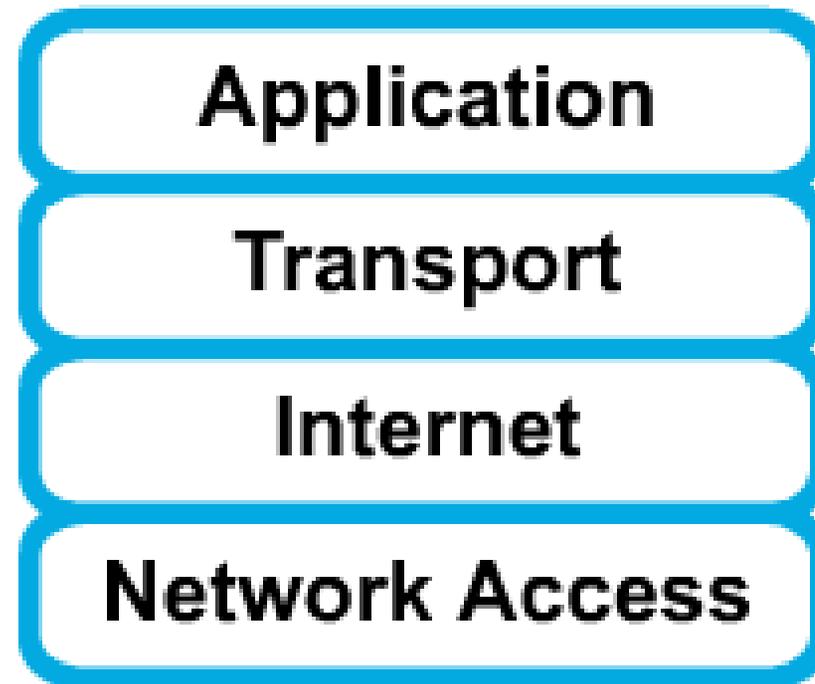


Il modello **TCP/IP** è stato sviluppato dal Department of Defense (DoD) negli anni 60 come un progetto militare atto ad assicurare la comunicazione dei dati anche sotto attacco.

Da allora, TCP/IP è diventato lo standard de facto usato per la comunicazione dei dati su **Internet**.



# The TCP/IP Reference Model

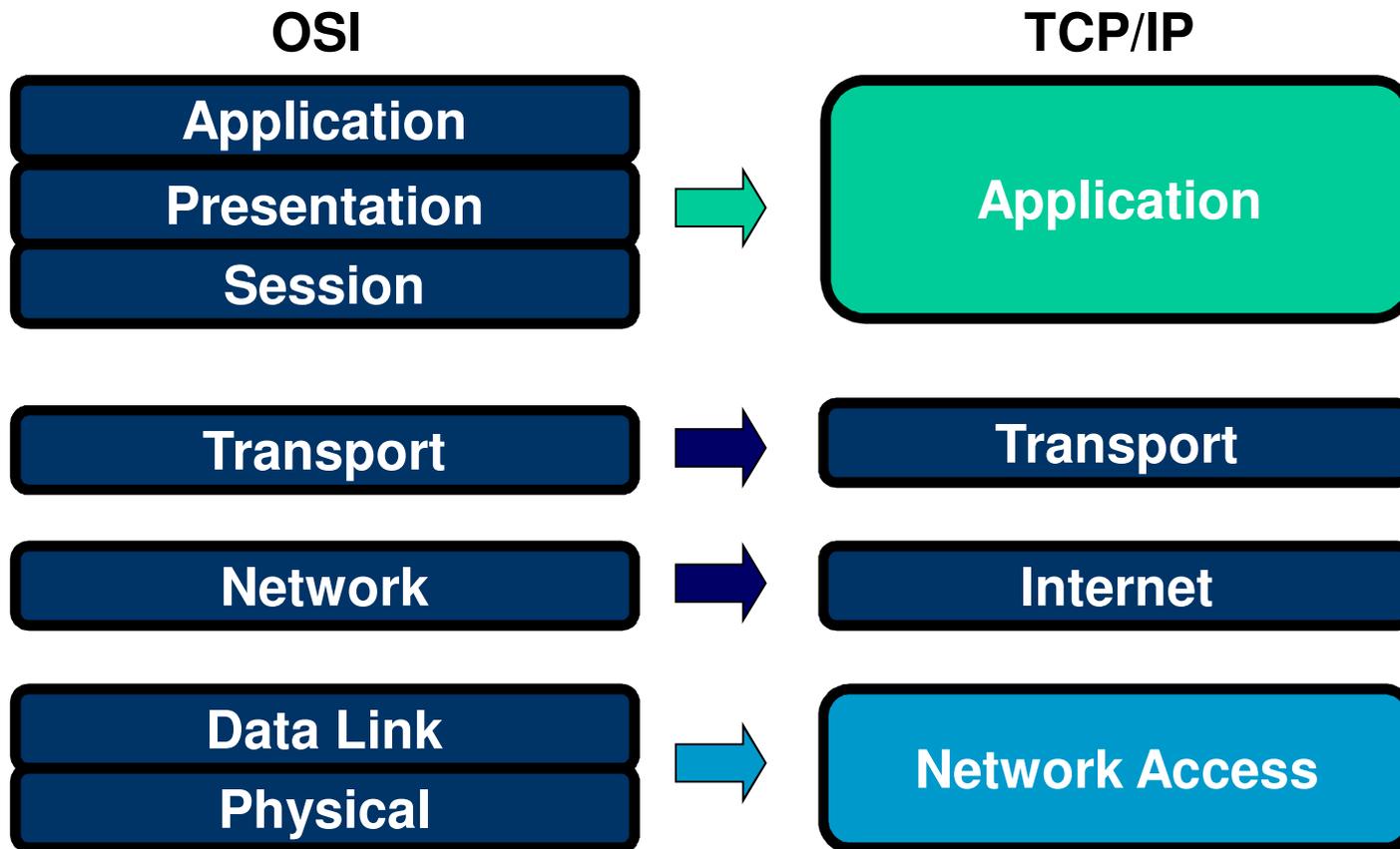


# TCP/IP vs OSI

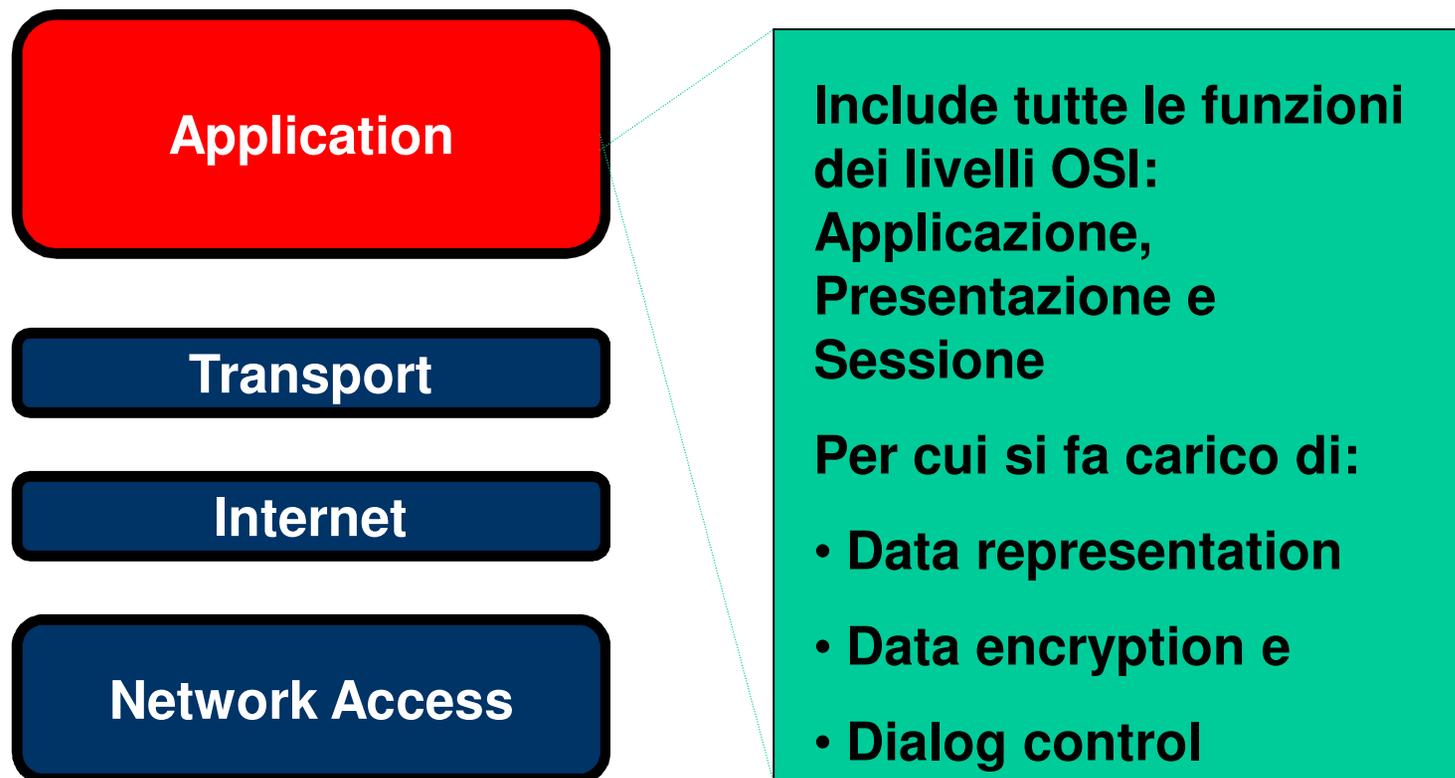
- ☞ TCP/IP è una descrizione di protocolli già esistenti ed è quindi meno generale e flessibile
- ☞ I protocolli TCP/IP erano già largamente usati in ambito accademico quando fu proposto il modello OSI
- ☞ Il modello e i protocolli OSI hanno dei difetti
  - ◆ Il livello sessione è poco utile e quello presentazione poco significativo
  - ◆ I livelli rete e datalink sono molto complessi e richiedono una strutturazione in sottolivelli
  - ◆ Lo standard è complesso e i protocolli difficili da implementare
- ☞ TCP/IP non distingue bene fra servizi, interfacce e protocolli



# Comparazione tra i 2 Modelli



# Esplorando i 4 Layers del modello TCP/IP



# Esempi di protocolli Applicativi



**Application**

**Transport**

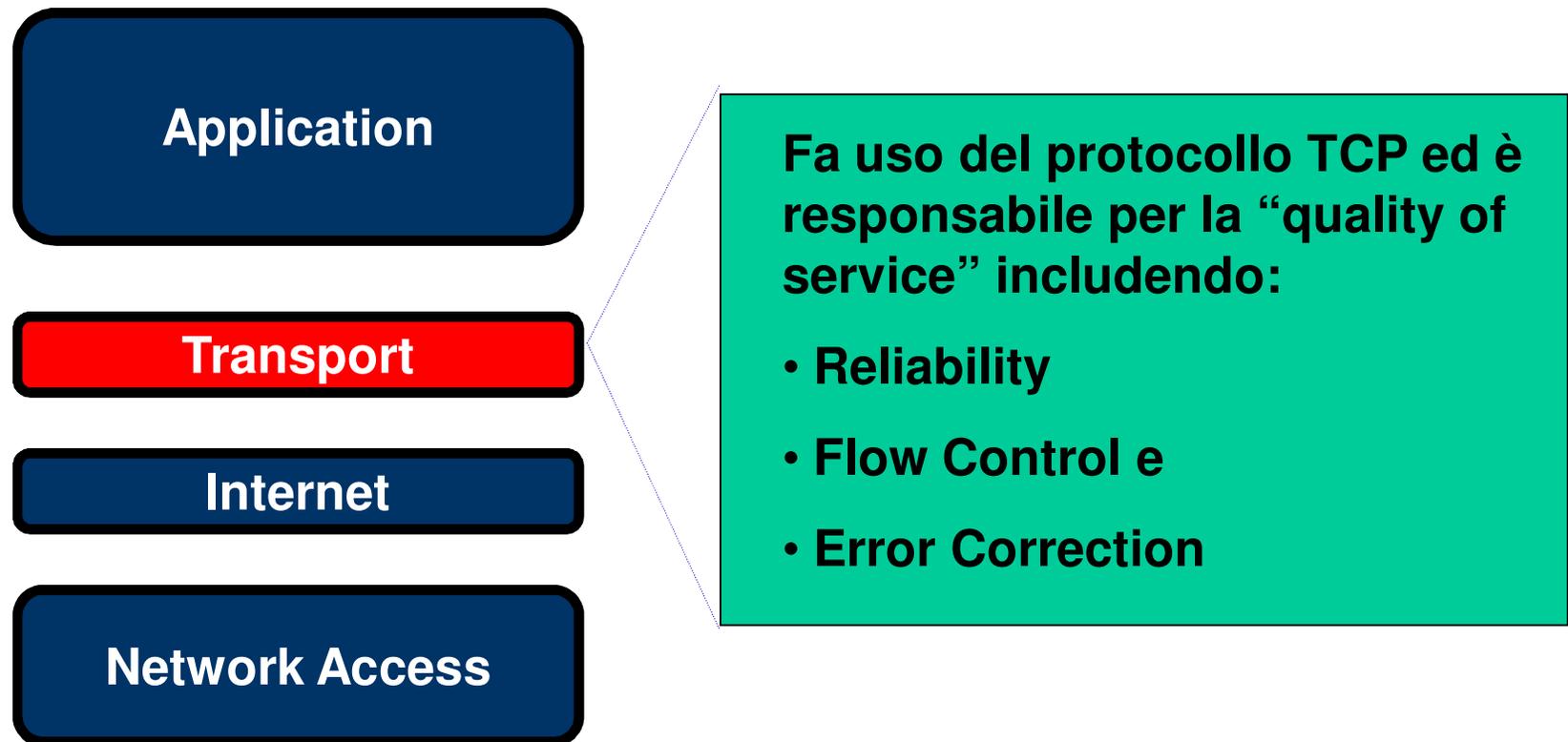
**Internet**

**Network Access**

1. **FTP- File Transfer Protocol**
2. **HTTP- Hypertext Transfer Protocol**
3. **SMTP- Simple Mail Transfer Protocol**
4. **DNS- Domain Name Service**
5. **TFTP- Trivial File Transfer Protocol**



# Esplorando i 4 Layers del modello TCP/IP



# Protocolli di livello trasporto

Application

Transport

Internet

Network Access

1. **TCP** (connection-oriented, reliable, guaranteed delivery.)
2. **UDP** (connectionless, unreliable, meno overhead, la reliability deve essere garantita a livello Application)
3. **SPX**



# Esplorando i 4 Layers del modello TCP/IP

Application

Transport

Internet

Network Access

Usa il protocollo **IP** ed è responsabile di:

- determinazione del Path
- Packet switching.



# Esplorando i 4 Layers del modello TCP/IP

**Application**

**Transport**

**Internet**

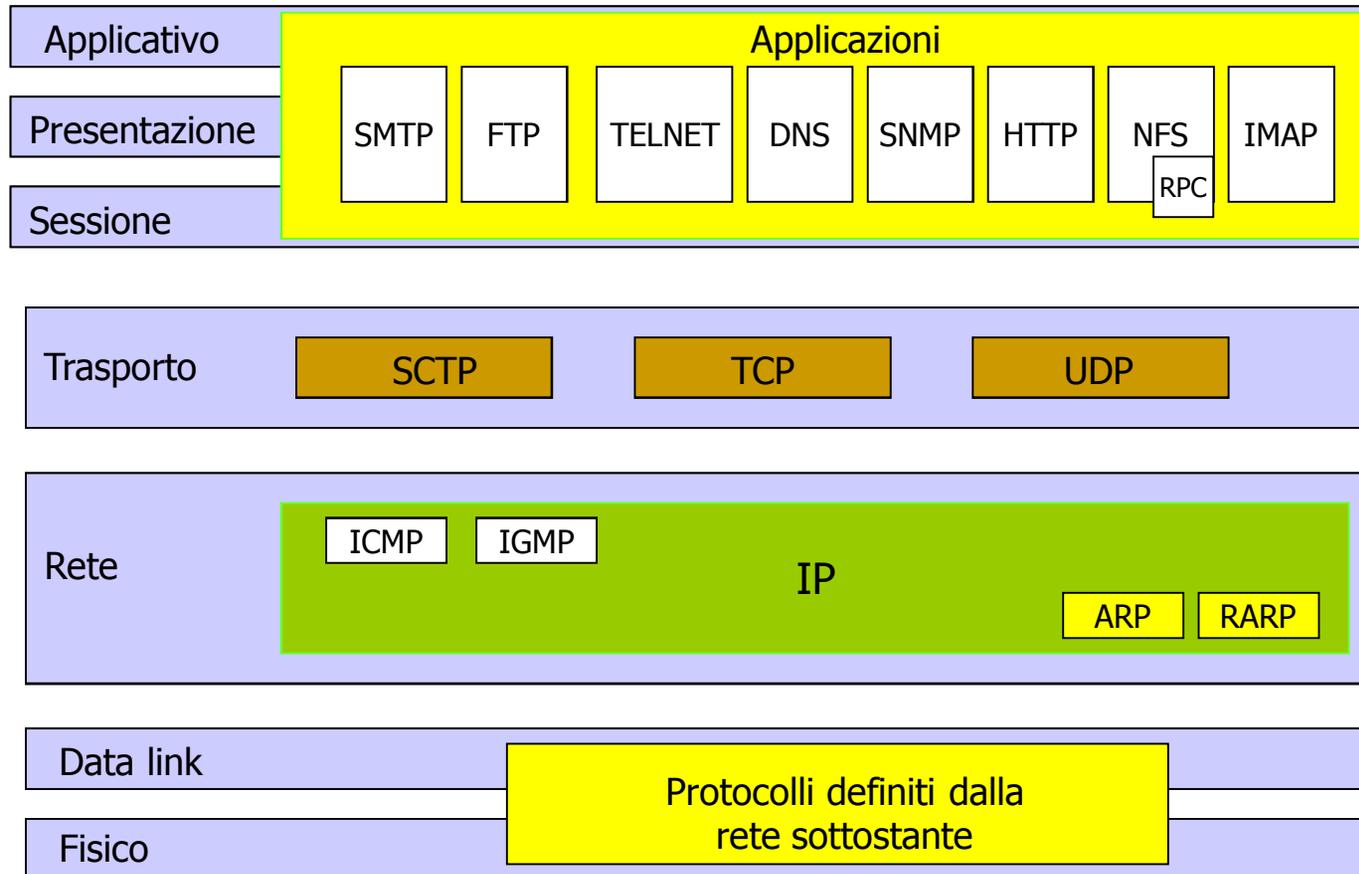
**Network Access**

**Raggruppa tutte le funzioni OSI di Data Link e Physical layers incluso:**

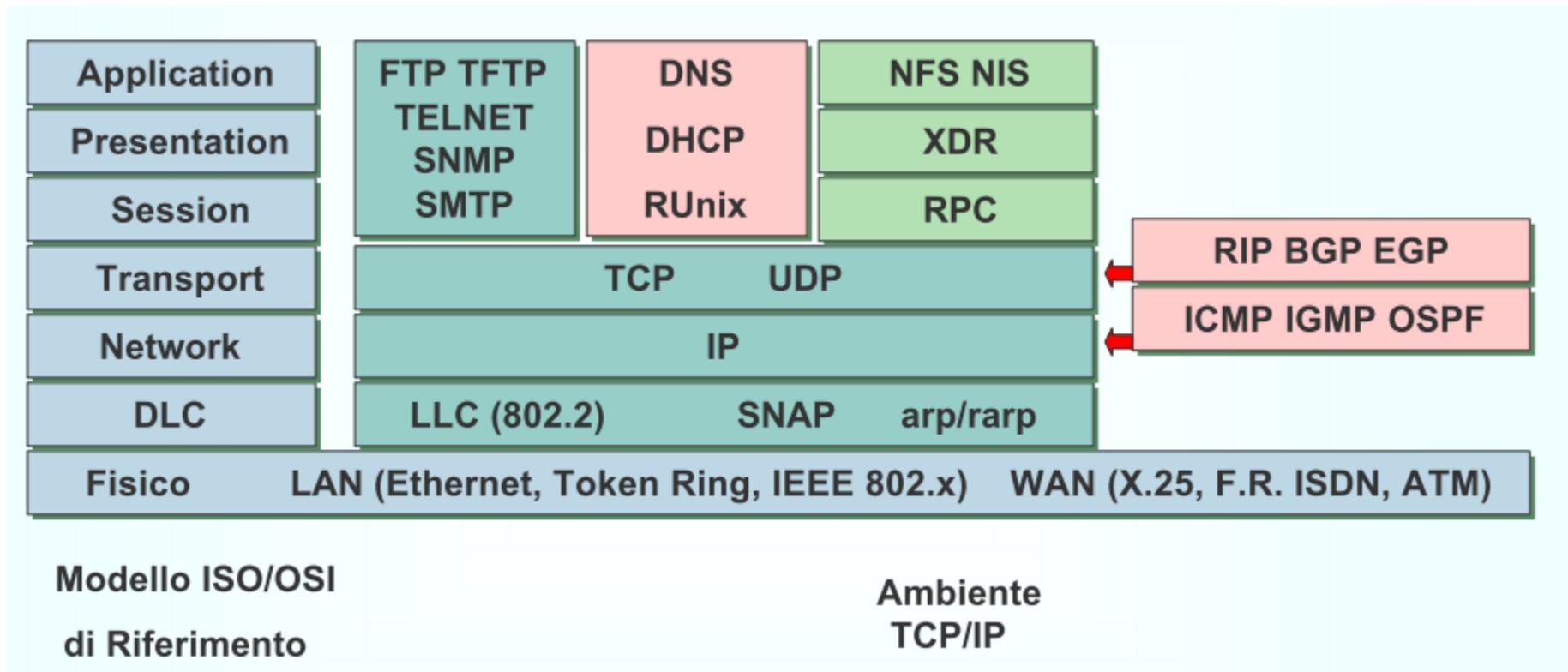
- **Processi necessari ad IP per assicurare che il pacchetto raggiunga la sua destinazione.**
- **Tutte le tecnologie LAN & WAN come 100BaseTX o Frame Relay.**



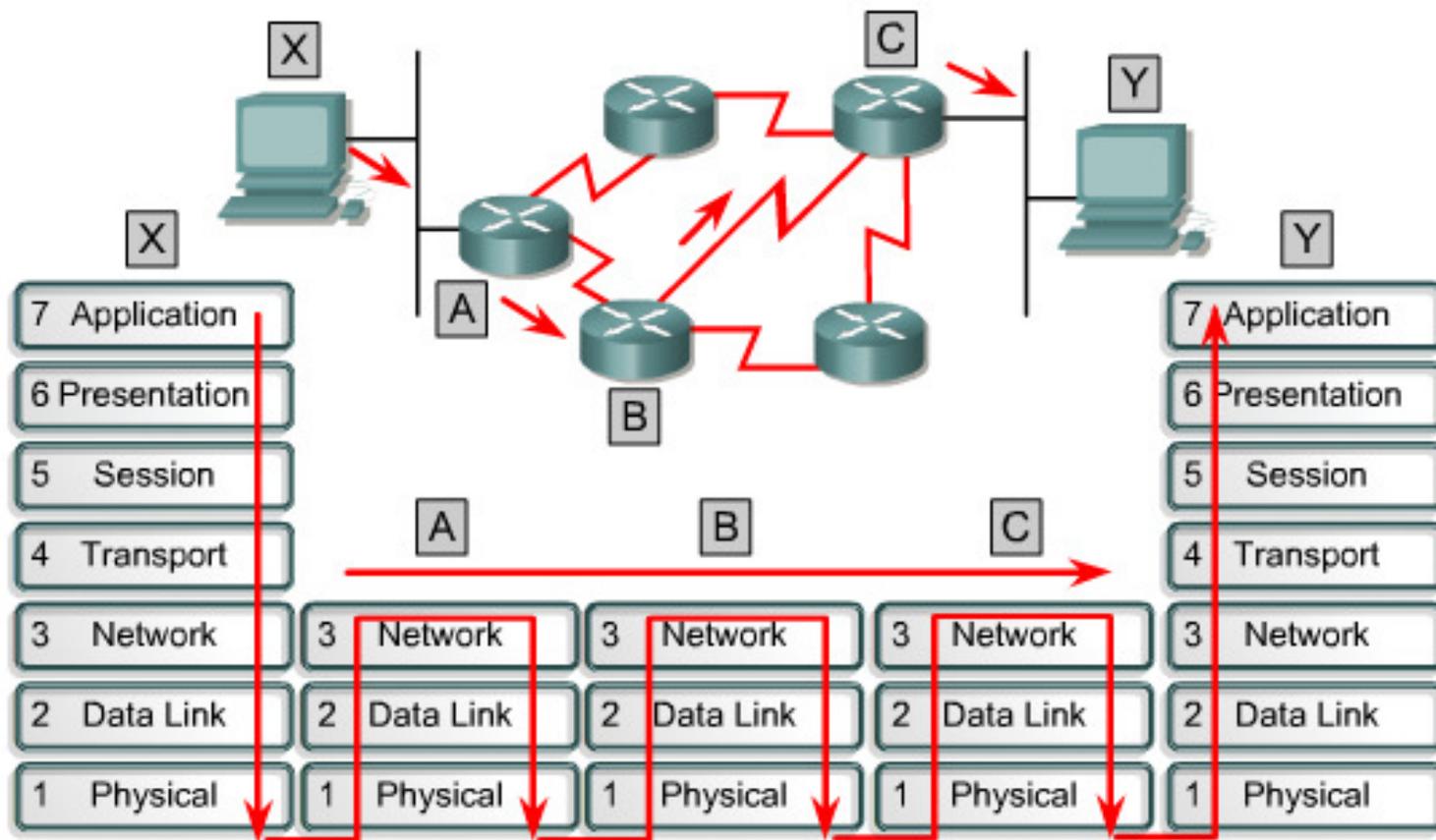
# La suite di protocolli TCP/IP



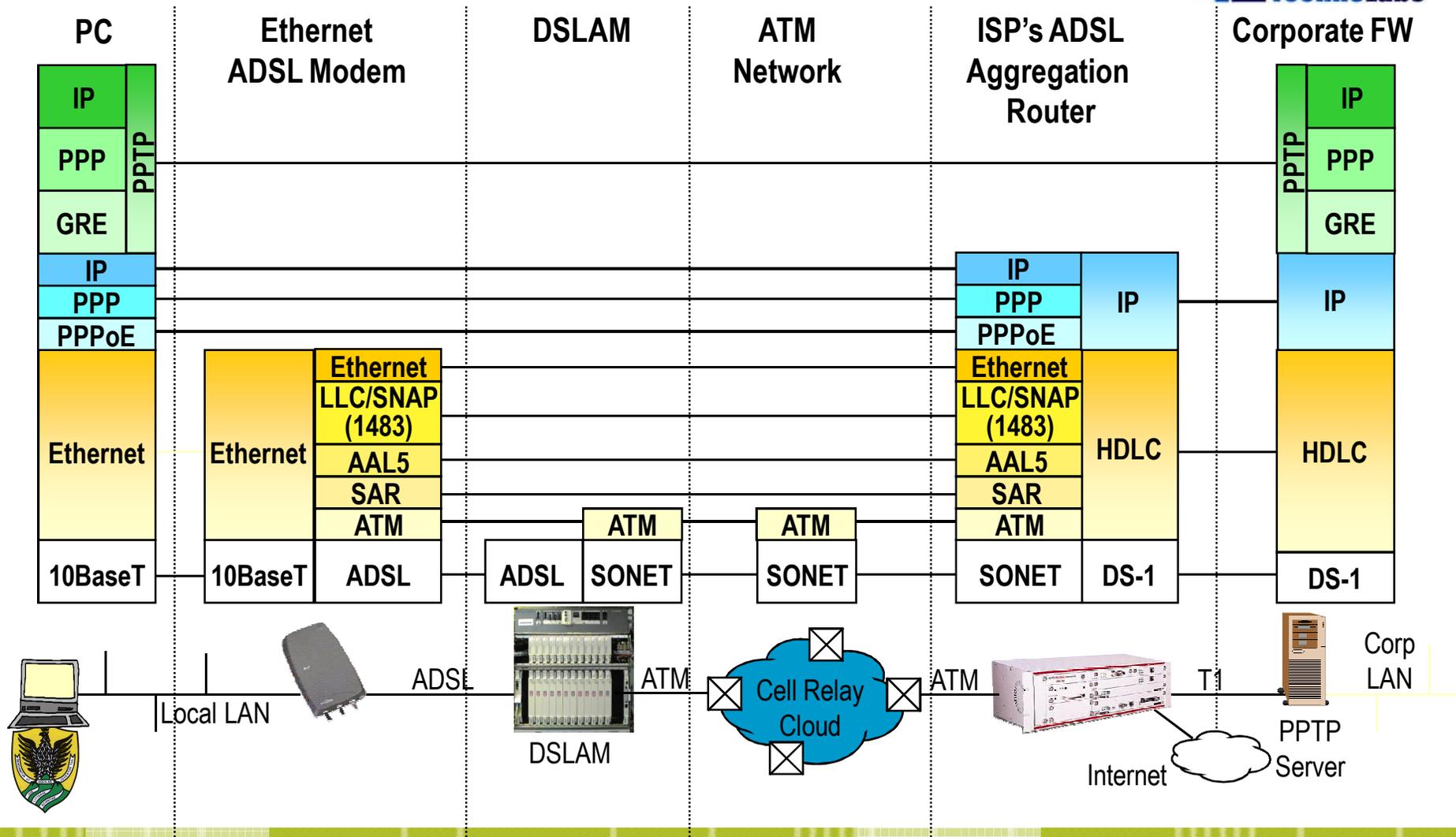
# ISO/OSI vs TCP/IP



# Data Flow



# Data Flow End-to-End: Example



# Indirizzamento

- ☞ Per definizione una rete è fatta da più stazioni e il suo scopo è di permettere lo scambio di informazioni tra esse. Per far ciò abbiamo bisogno di un metodo di indirizzamento che sia univoco.
- ☞ In realtà un indirizzo deve essere univoco all'interno del livello architetturale nel quale ha validità (locally-unique)
  - ◆ A livello DATA-Link l'indirizzo deve identificare univocamente la stazione in ambito LAN e non è necessario distinguere workstation su link disgiunti poiché WS su differenti LAN non possono comunicare a livello DATA-Link
  - ◆ A livello Network l'indirizzamento deve essere capace di identificare la stazione globalmente



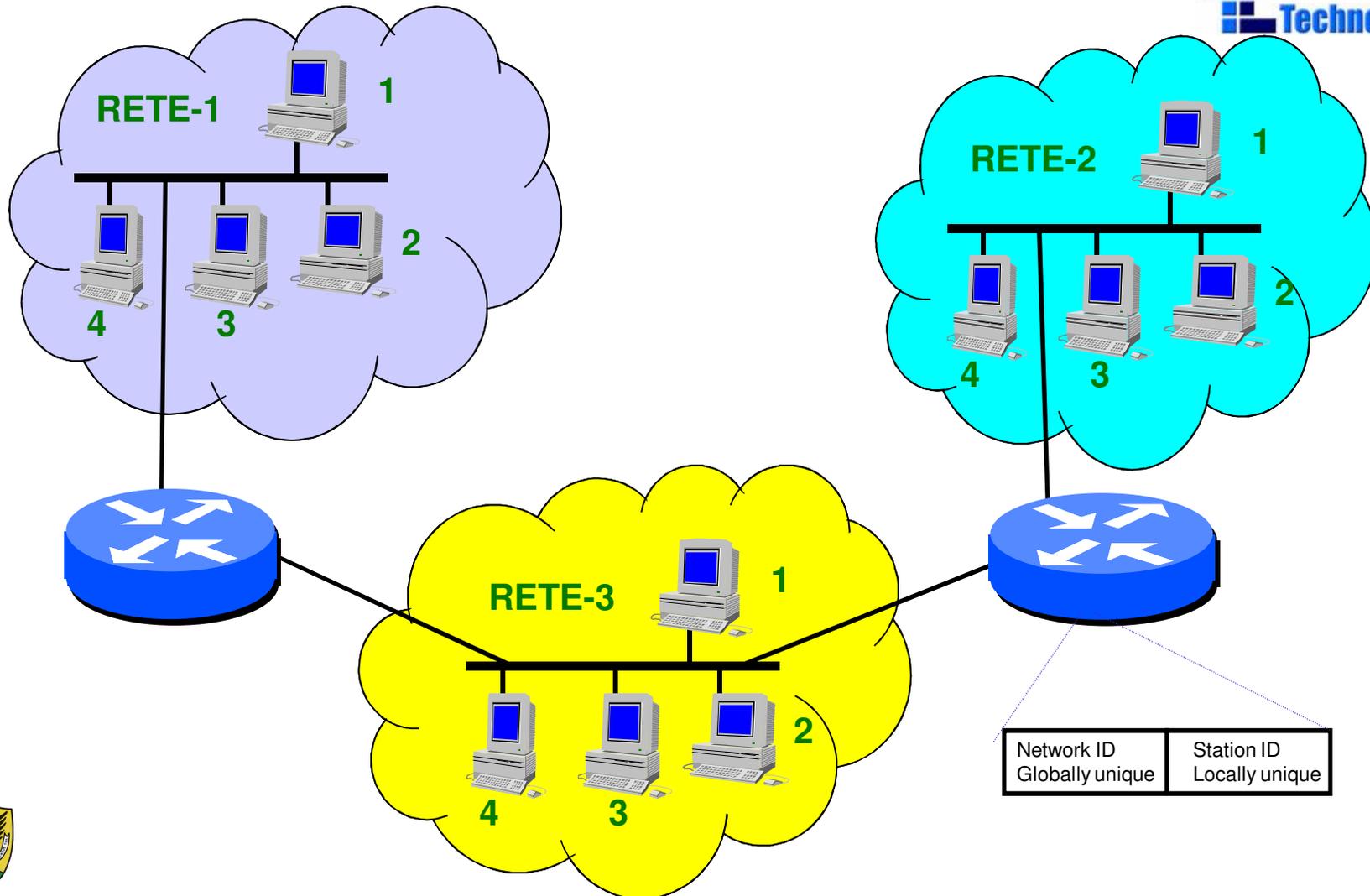
# Indirizzamento



- ☞ Ogni endpoint della rete di comunicazione deve avere un indirizzo.
- ☞ Considerate 2 processi che comunicano su internet o su una intranet:
  - ◆ La rete deve essere specificata
  - ◆ L' host (end-system) deve essere specificato
  - ◆ Il processo deve essere specificato.
- ☞ In una analogia postale per recapitare una lettera va specificato
  - ◆ Nazione → Città → Via → Numero Civico



# Indirizzamento



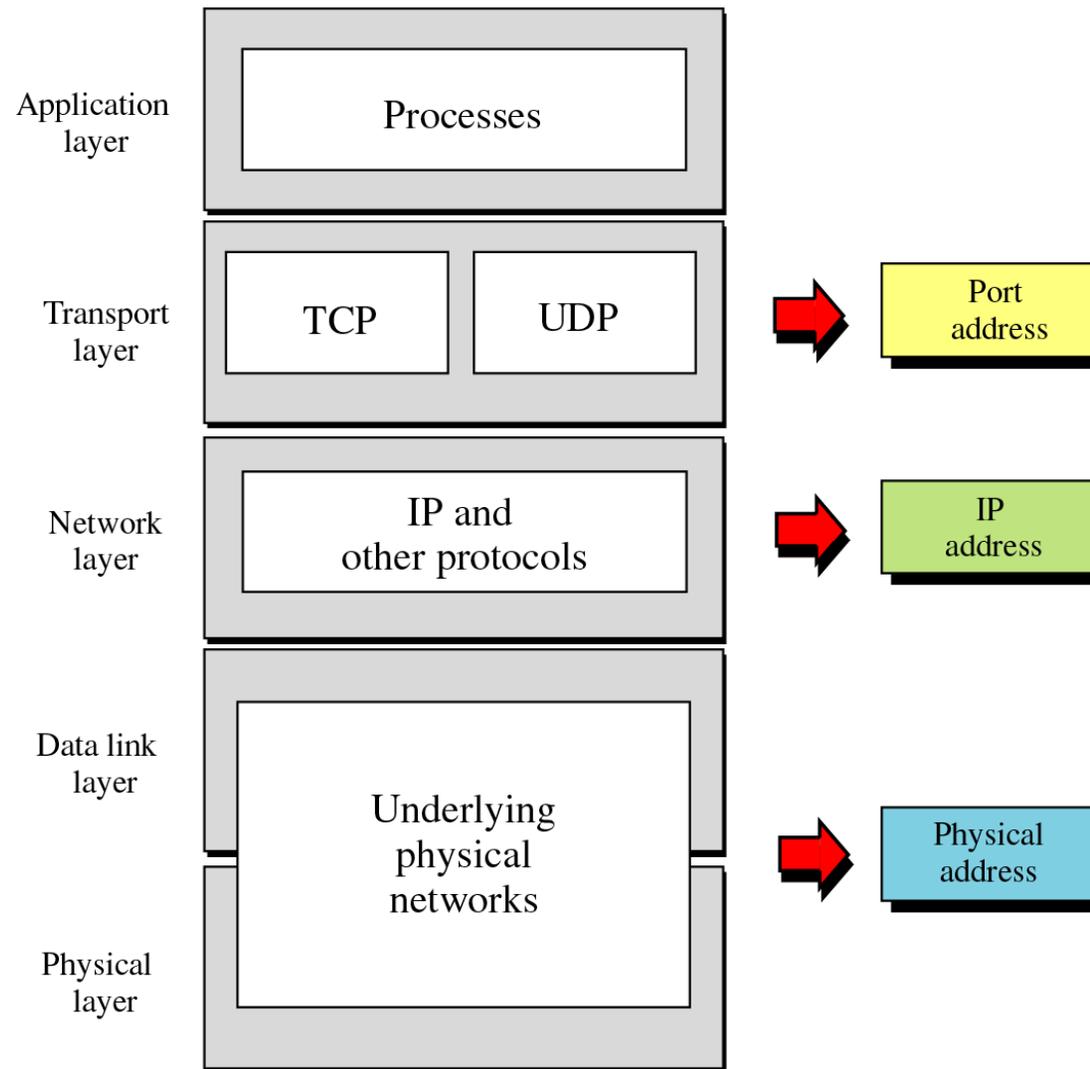
# Indirizzamento e Layers



- Physical Layer: non c'è bisogno di indirizzamento
- Data Link Layer – l'indirizzamento deve garantire la possibilità di selezionare qualunque host sulla LAN.
- Network Layer – l'indirizzamento deve fornire tutte le informazioni necessarie per il routing.
- Transport Layer – l'indirizzo deve identificare il processo destinazione.



# Indirizzamento e Layers (cont.)



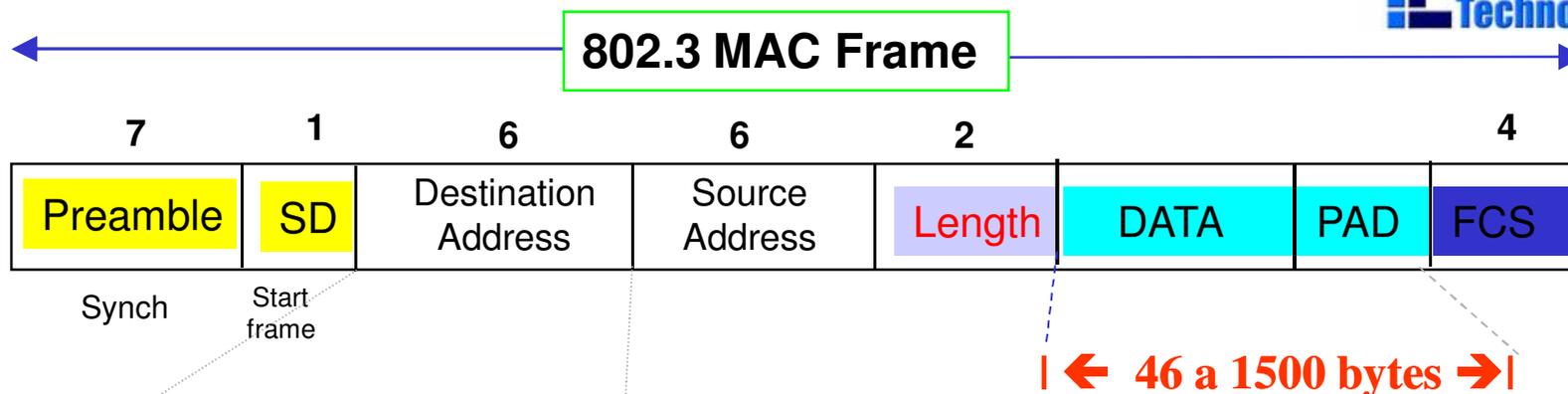
# Broadcasts



- Diverse reti supportano la nozione di spedire un messaggio da un host a tutti gli altri hosts appartenenti a quella rete.
- Per ottenere ciò è stato previsto un indirizzo speciale chiamato “broadcast address”.
- Diversi servizi popolari sono basati sul concetto di broadcasting (YP/NIS, rup, rusers)



# Esempio: indirizzamento a livello data-link IEEE 802.3



0	x	Unicast address
---	---	-----------------

1	x	Multicast address
---	---	-------------------

x	0	Local address
---	---	---------------

x	1	Global address
---	---	----------------

- Destination address può essere:
  - UNICAST (single address)
  - MULTICAST (group address)
  - BROADCAST (FF-FF-FF-FF-FF-FF)
- Source address è sempre di tipo UNICAST
- Gli indirizzi sono definiti su base locale o globale
- Complessivamente sono possibili  $2^{46}$  indirizzi globali



# Gli indirizzi MAC

- ☞ Sono standardizzati dalla IEEE
  - ◆ sono lunghi 6 byte, cioè 48 bit
  - ◆ si scrivono come 6 coppie di cifre esadecimali
- ☞ Ad esempio:

0000100000000000010101100111100000011110011010

0	8	0	0	2	b	3	c	0	7	9	a
---	---	---	---	---	---	---	---	---	---	---	---

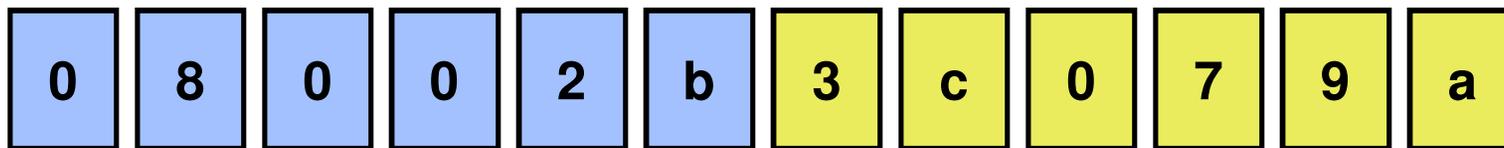
08-00-2b-3c-07-9a



# Indirizzi MAC



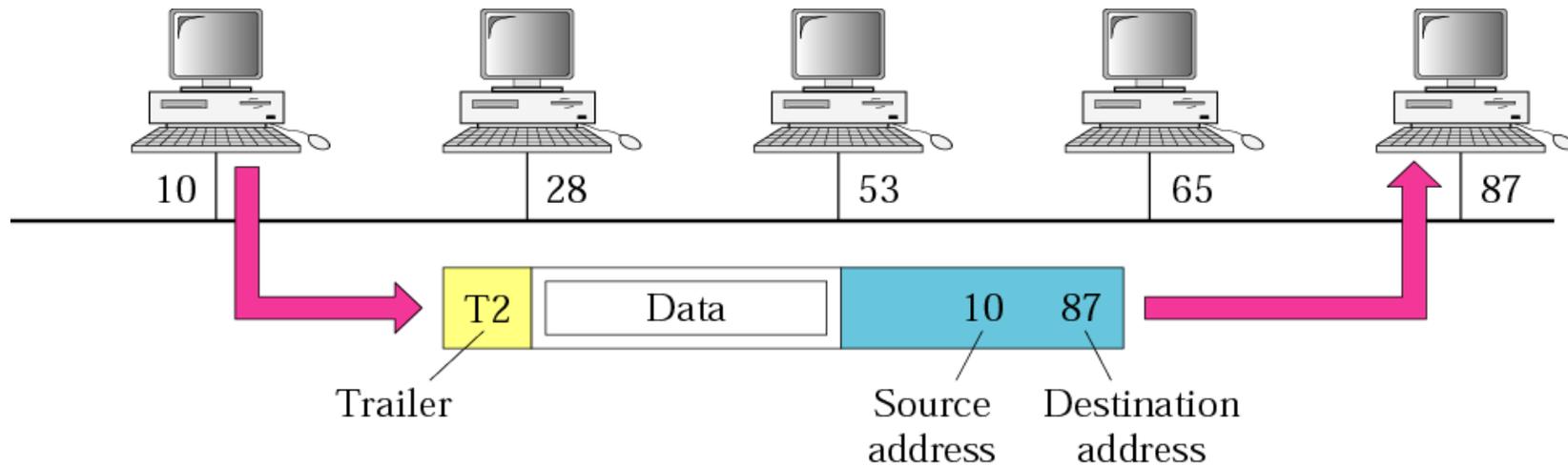
- ☞ Si compongono di due parti grandi 3 bytes ciascuna:
  - ◆ I tre bytes più significativi indicano il lotto di indirizzi acquistato dal costruttore della scheda, detto anche *vendor code* o *OUI (Organization Unique Identifier)*.
    - ◆ *Esempio:* 00-60-2F è assegnato a CISCO e 08-00-2b a DEC
  - ◆ I tre bytes meno significativi sono una numerazione progressiva decisa dal costruttore



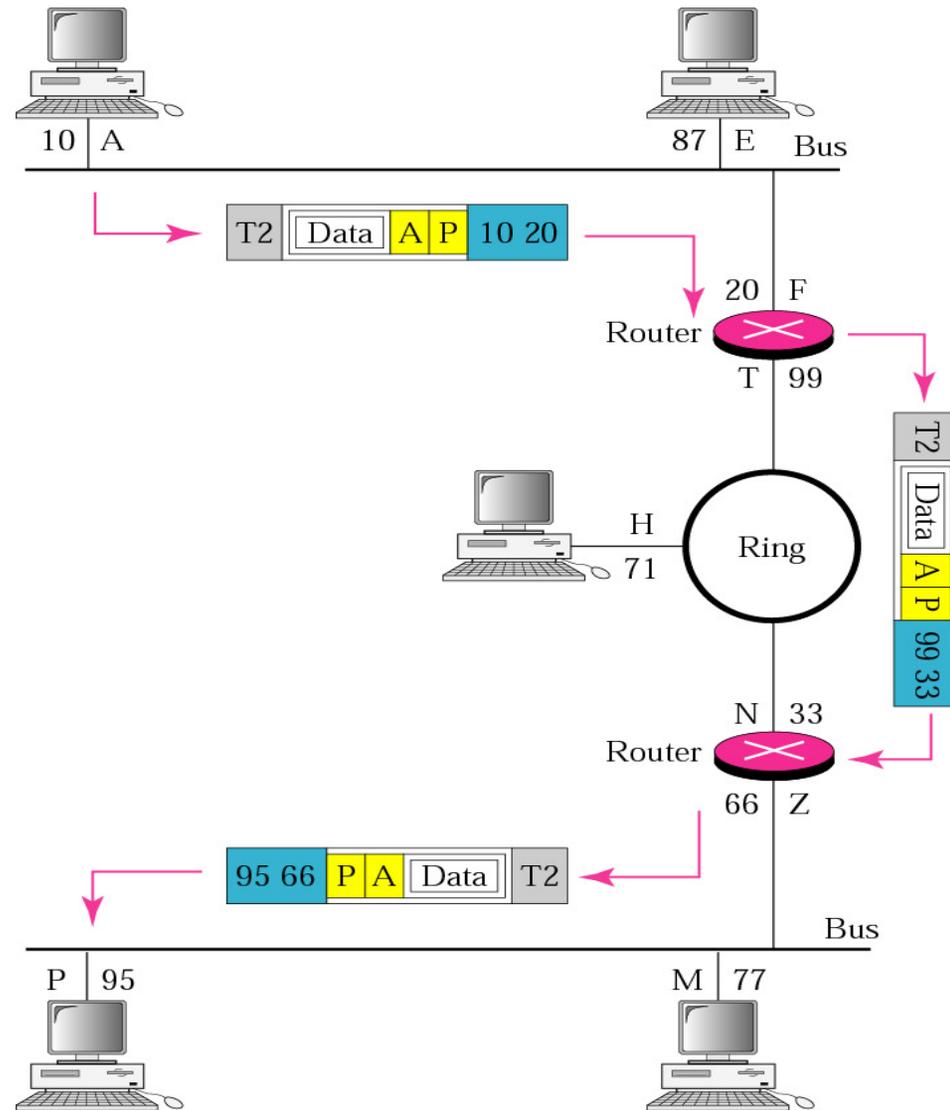
OUI assegnato dall'IEEE

Assegnato dal costruttore

# Indirizzamento fisico



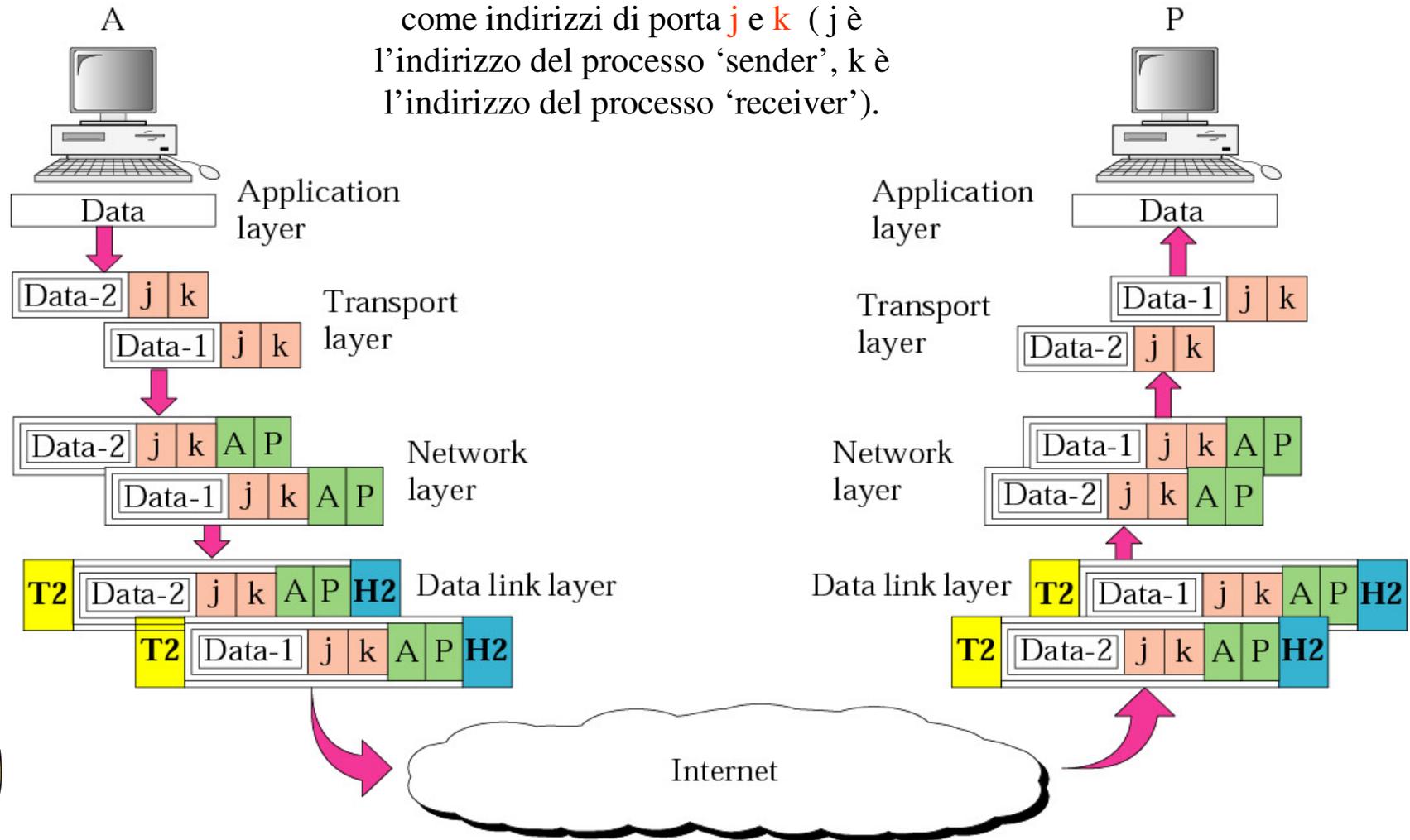
# Indirizzamento IP



# Port address



Dati generati dai livelli superiori hanno come indirizzi di porta **j** e **k** (**j** è l'indirizzo del processo 'sender', **k** è l'indirizzo del processo 'receiver').



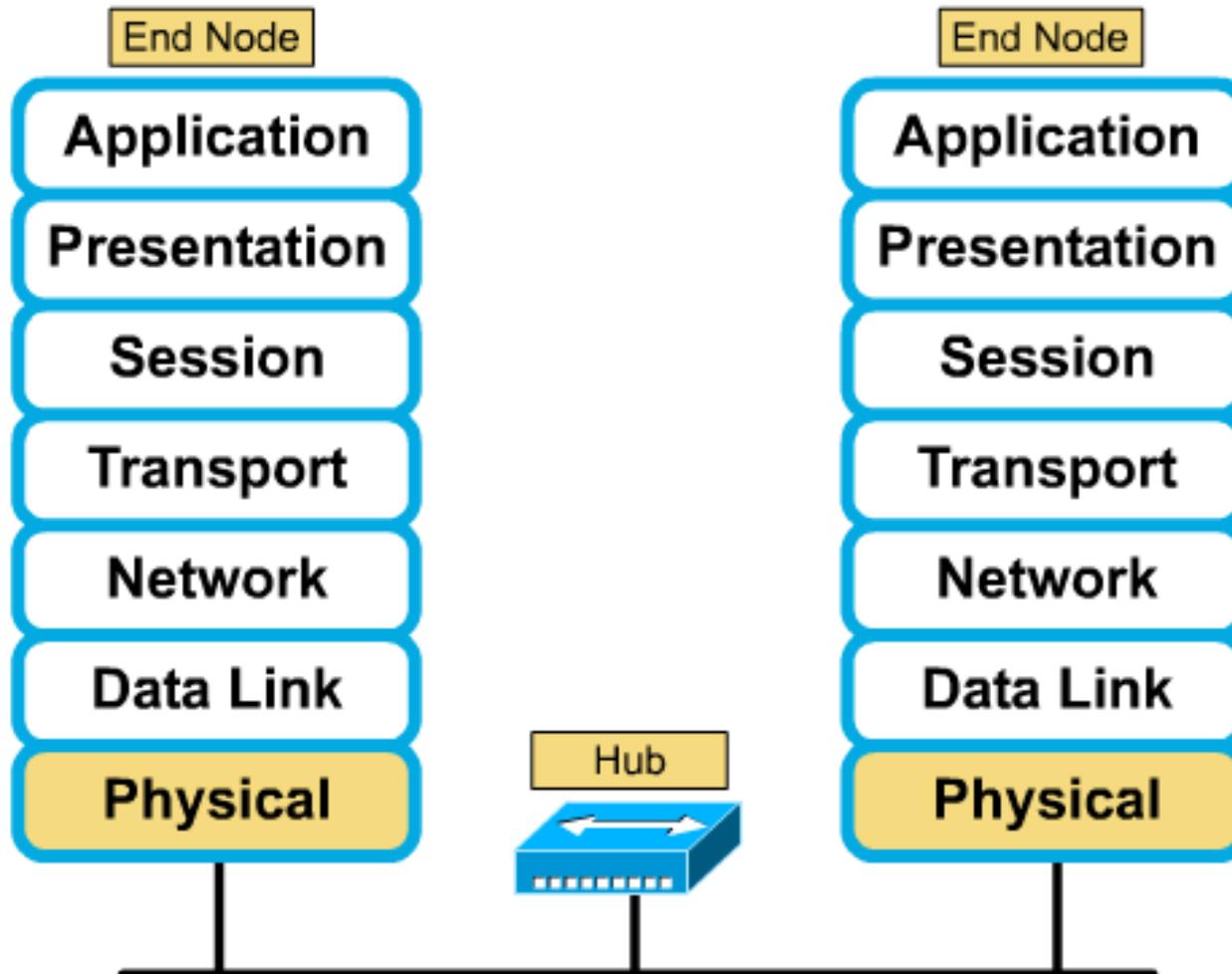
# Networking Devices

☞ Equipment that connects directly to a network segment is referred to as a device. These devices are broken up into two classifications. The first classification is end-user devices. The second classification is network devices.

Network Devices	
Repeater 	Bridge 
Small Hub (10BASE-T) 	Workgroup Switch 
100BASE-T Hub 	Router 
Hub 	Network Cloud 



# Hub: Livello 1

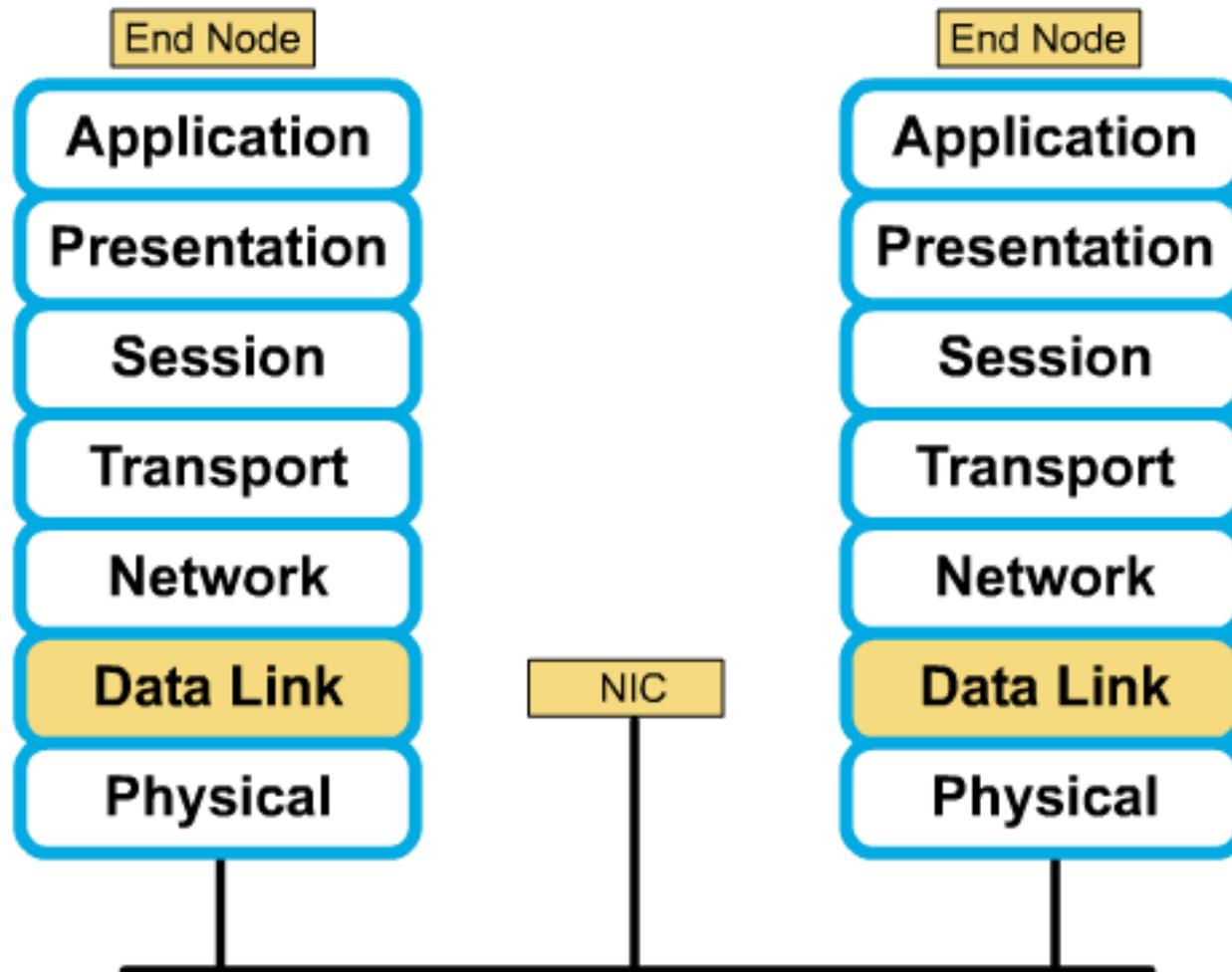


# HUB o Repeater

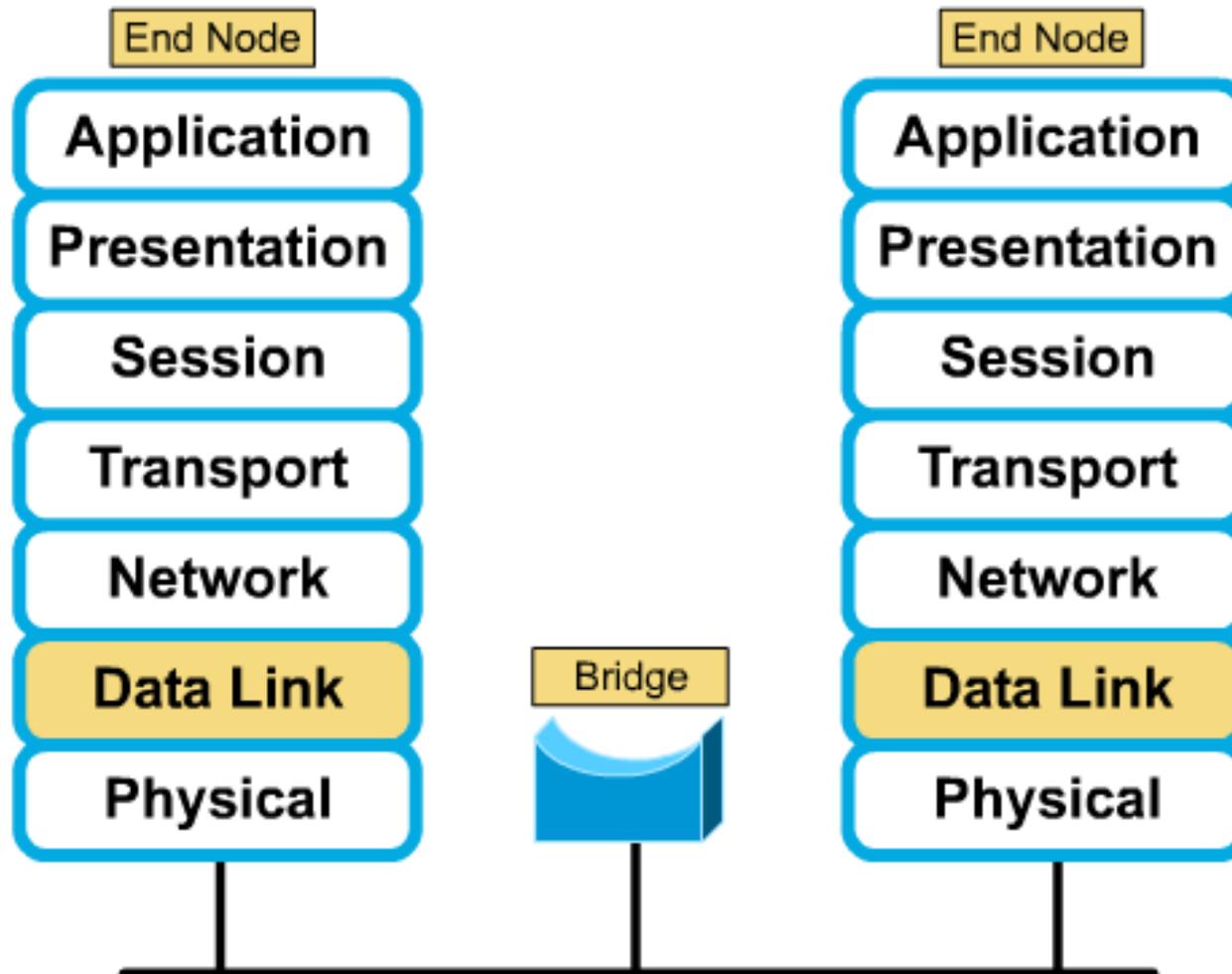
- Copia I bits da un segmento di rete ad un altro
- Non guarda a nessun bits
- Permette l'estensione di una rete oltre i limiti fisici



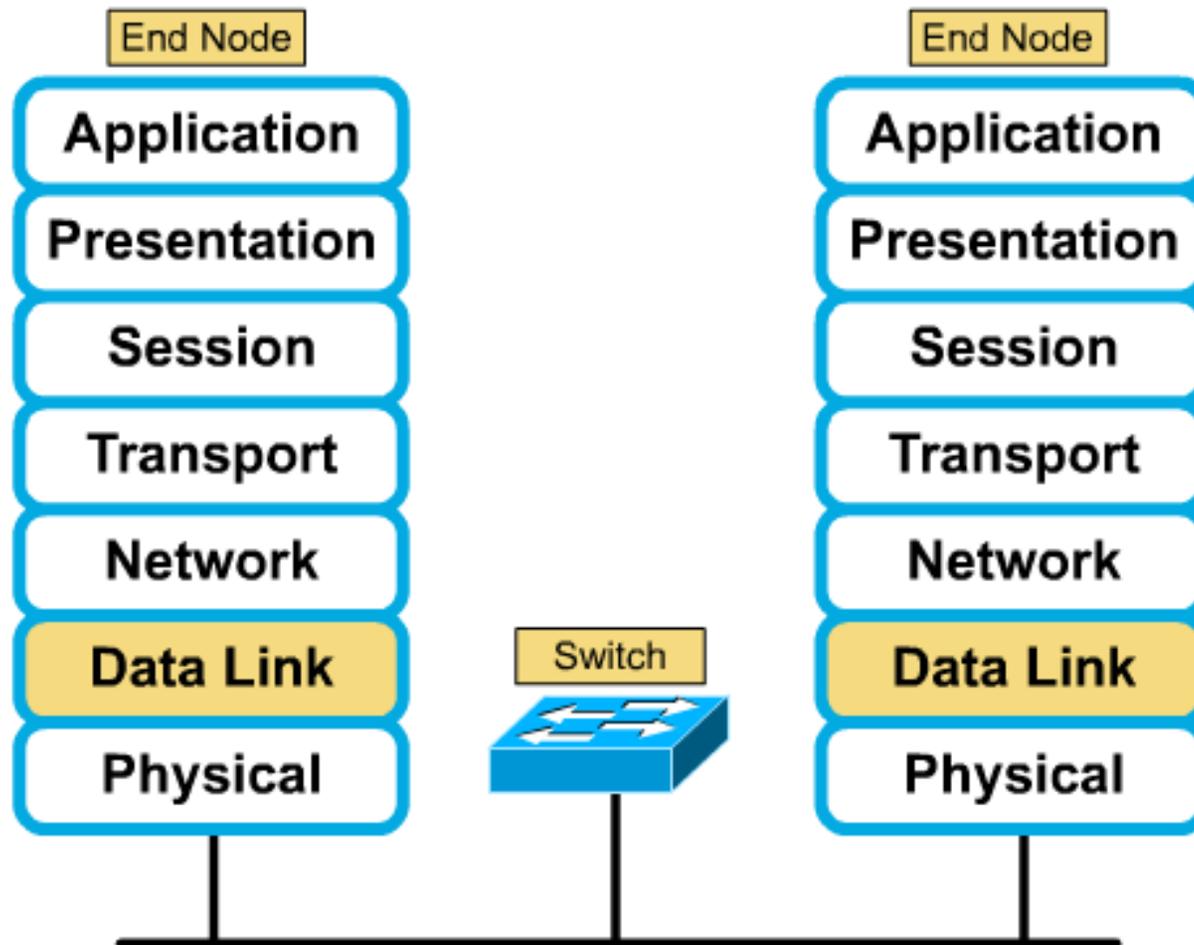
# NICs in the OSI Model



# Bridges: Layer 2



# Switches: Layer 2



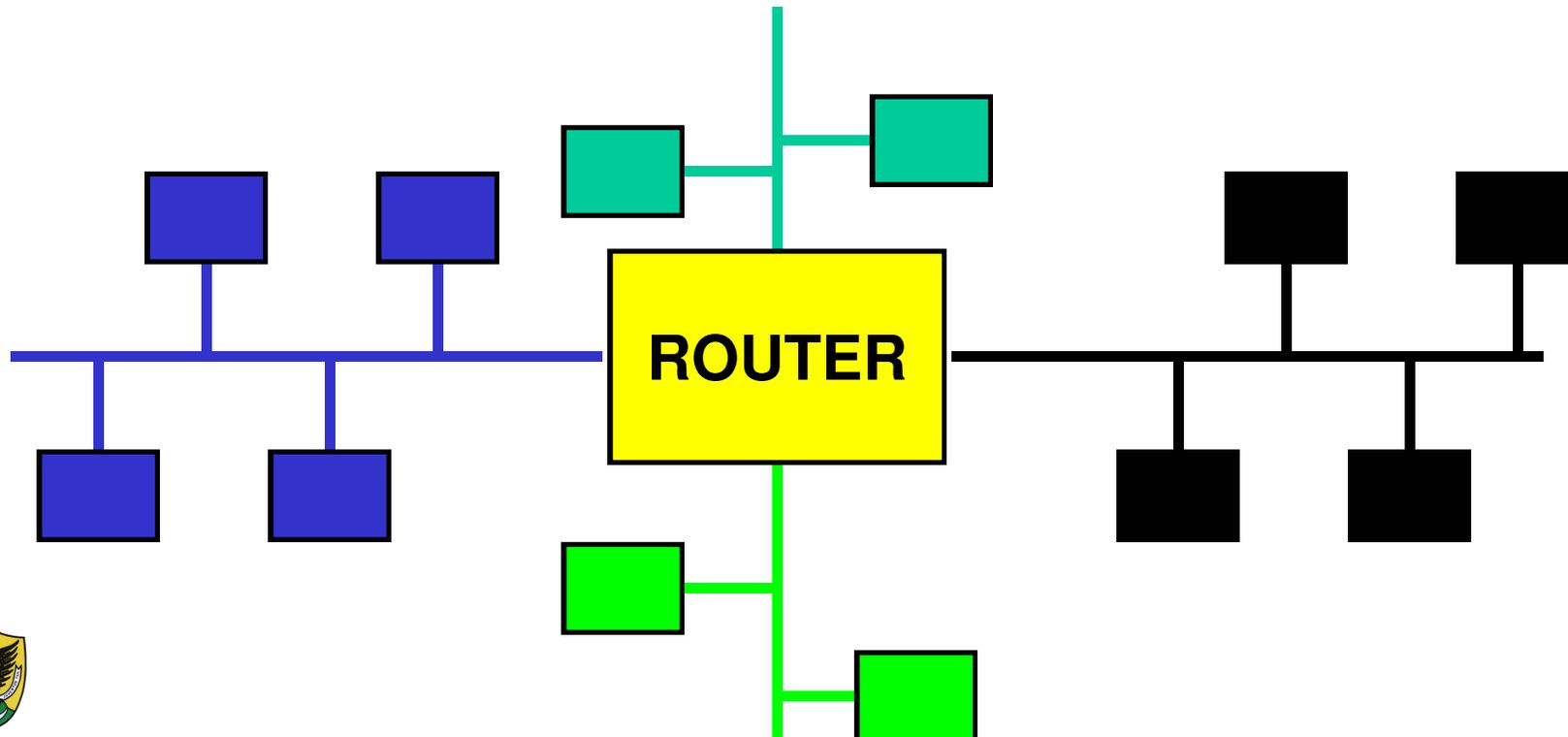
# Bridge

- Copia frames da un segmento di rete ad un altro
- Opera in modo selettivo - non copia tutte le frames (deve ovviamente ispezionare i data-link headers).
- Estende la rete oltre i limiti fisici.



# Router

- Copia packets da una rete ad un'altra.
- Prendono decisioni su quale strada un packet dovrebbe prendere (ispeziona i network headers).



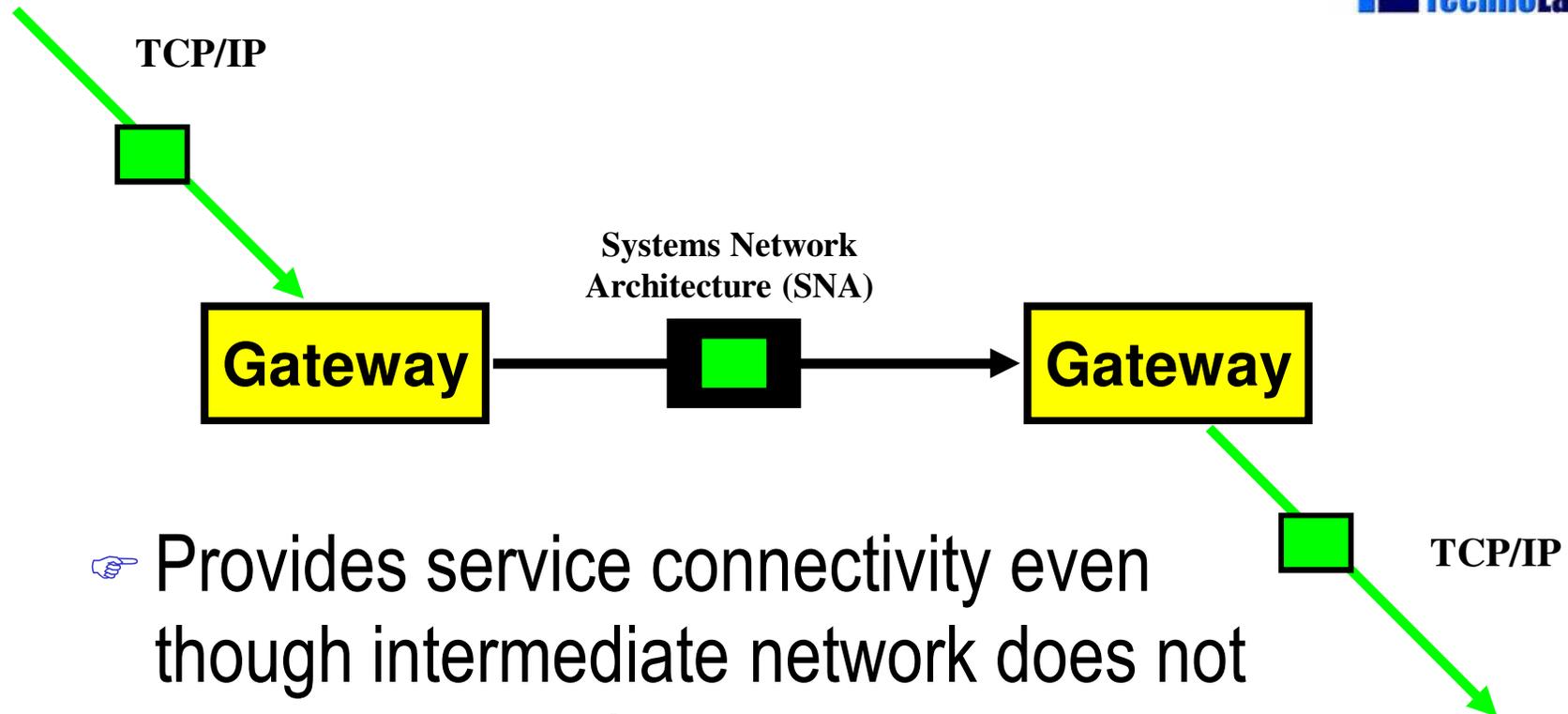
# Gateway



- Opera come un router
- Data conversions tra network layer.
- Conversioni:
  - ◆ encapsulation - use an intermediate network
  - ◆ translation - connect different application protocols
  - ◆ encryption - could be done by a gateway



# Encapsulation Example



☞ Provides service connectivity even though intermediate network does not support protocols.



# Translation



- ☞ Traduzione dal protocollo 'verde' al protocollo 'marrone'



# Primo Modulo: Sommario



## Copertura del 1° modulo!

- ☞ Internet overview
- ☞ Che cos'è un protocollo ?
- ☞ Network edge, core, access
  - ◆ packet-switching versus circuit-switching
- ☞ Internet/ISP struttura
- ☞ Prestazioni: perdite, ritardo
- ☞ Modello a layers
- ☞ Un pò di storia



## Ora voi avete:

- ☞ Contesto, overview, “feeling” di cosa sia il networking
- ☞ Andremo più in profondità nei prossimi moduli !