

Reti di Elaboratori

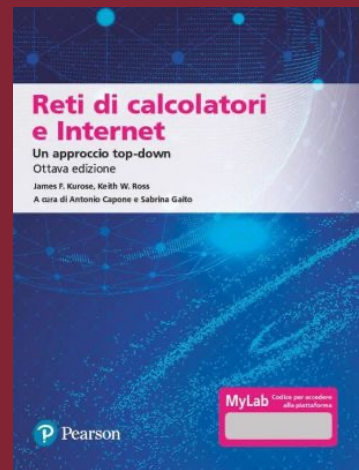
Introduzione allo stack protocollare



SAPIENZA
UNIVERSITÀ DI ROMA

Alessandro Checco

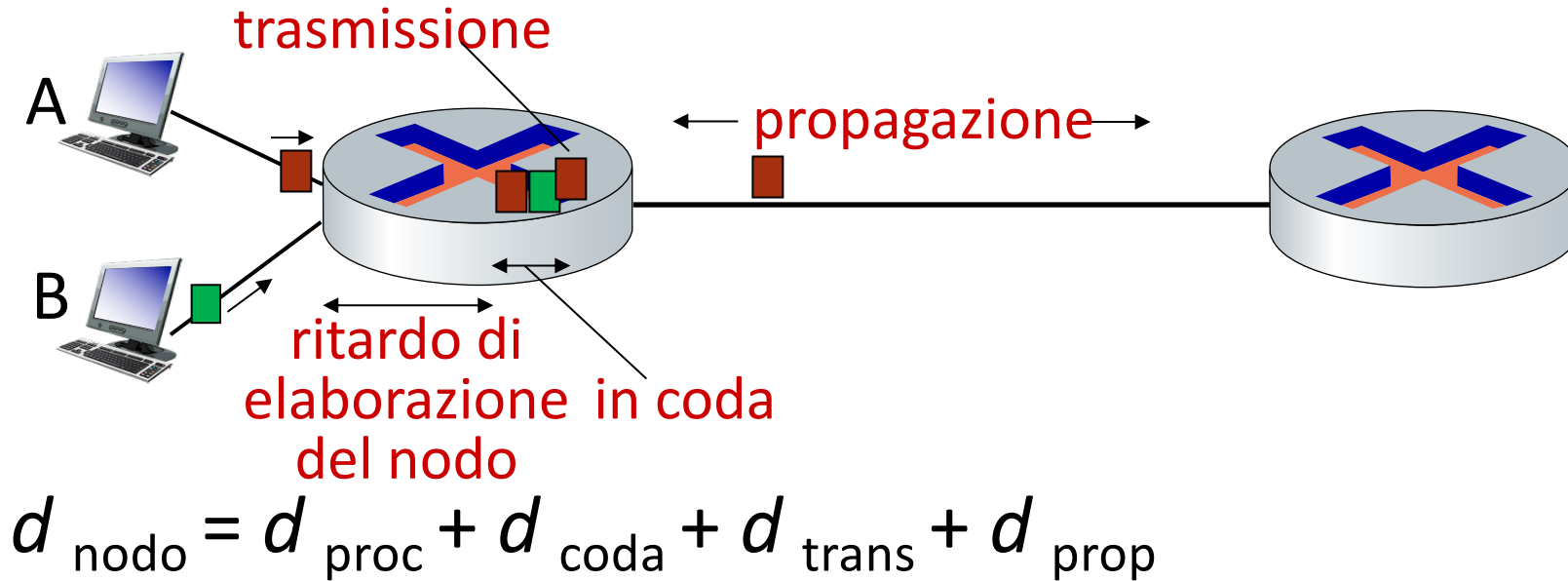
alessandro.checco@uniroma1.it



Capitolo 1

Riepilogo

Ritardo (latenza) del pacchetto: quattro fonti



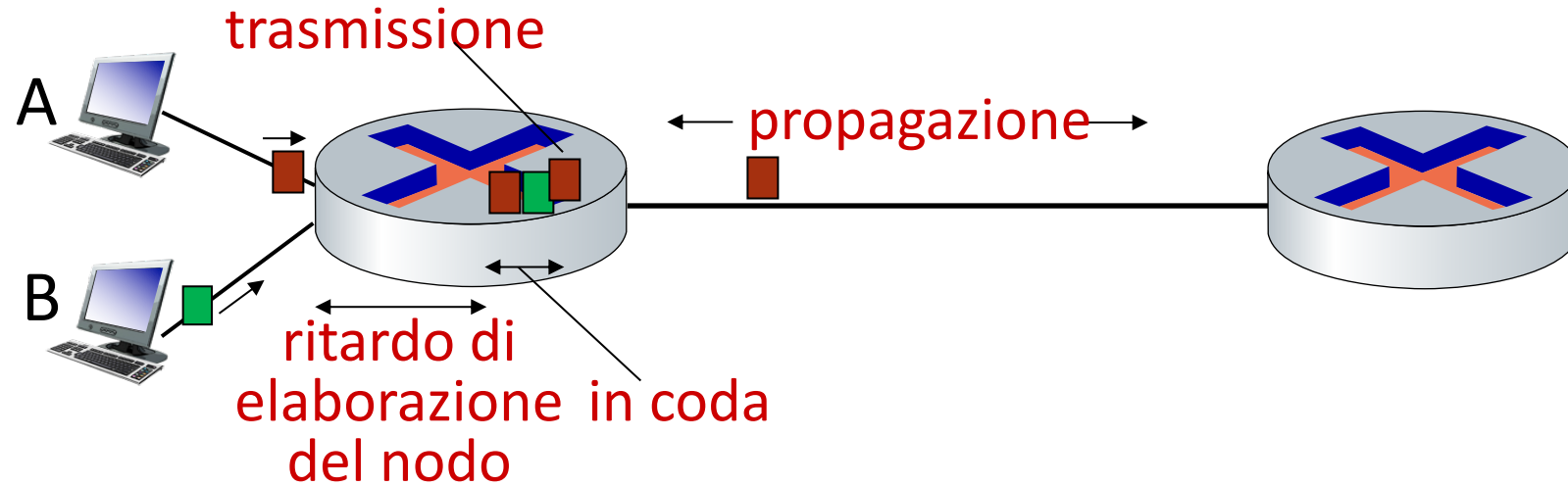
d_{proc} : elaborazione del nodo (processing delay)

- controllo errori sul bit
- determinazione del link di uscita
- tipicamente $< \mu\text{s}$

d_{queue} : ritardo di accodamento (queuing delay)

- attesa di trasmissione (in coda nel buffer)
- dipende dal livello di congestione del router
- dipende dal rate di arrivo e può variare da pacchetto a pacchetto

Ritardo (latenza) del pacchetto: quattro fonti



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

d_{trans} : ritardo di trasmissione:

- L : lunghezza pacchetto (bit)
- R : velocità di trasmissione del link (bps)

■ $d_{\text{trans}} = L/R$

d_{prop} : ritardo di propagazione:

- k : lunghezza del collegamento fisico
- v : velocità di propagazione ($\sim 2 \times 10^8$ m/s)

■ $d_{\text{prop}} = k/v$

d_{trans} e d_{prop}
molto diversi!

Capitolo 1: sommario

- *Cos'è Internet ?*
- *Cos'è un protocollo?*
- Periferia della rete: host, rete di accesso, supporti fisici
- Nucleo di rete: commutazione pacchetto/circuito, struttura internet
- Prestazioni: bitrate, bandwidth, throughput, loss, latency
- Sicurezza
- **Livelli di protocollo, modelli di servizio**
- Storia

Livelli di protocollo e modelli di servizio

*Le reti sono complesse,
con tante parti:*

- host
- router
- link di vari supporti fisici
- applicazioni
- protocolli
- hardware, software

Domanda:

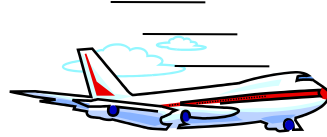
c'è qualche speranza di
avere una struttura
organizzativa della
rete?

.... o almeno *descrittiva*?

Protocollo

- Un protocollo **definisce le regole** che il mittente e il destinatario, così come tutti i sistemi intermedi coinvolti, devono rispettare per essere in grado di comunicare.
- In situazioni particolarmente semplici potrebbe essere sufficiente un solo protocollo, in situazioni più complesse potrebbe essere opportuno suddividere i compiti fra più **livelli (layer)**, nel qual caso è richiesto un protocollo per ciascun livello: si parla dunque di **layering di protocolli**.

Esempio: organizzazione di viaggi aerei



biglietto (acquisto)

bagaglio (check-in)

gate (imbarco)

decollo della pista

rotta aerea

biglietto (reclamo)

ritiro bagagli

gate (sbarco)

atterraggio in pista

rotta aerea

rotta aerea

viaggio aereo: una serie di passaggi, che coinvolgono molti servizi

Esempio: organizzazione di viaggi aerei

biglietto (acquisto)	<i>servizio di biglietteria</i>	biglietto (reclamo)
bagaglio (check-in)	<i>servizio bagagli</i>	ritiro bagagli (ritardo)
gate (imbarco)	<i>servizio gate</i>	gate (sbarco)
decollo della pista	<i>servizio di pista</i>	atterraggio in pista
rotta aerea	<i>servizio traffico aereo</i>	rotta aerea

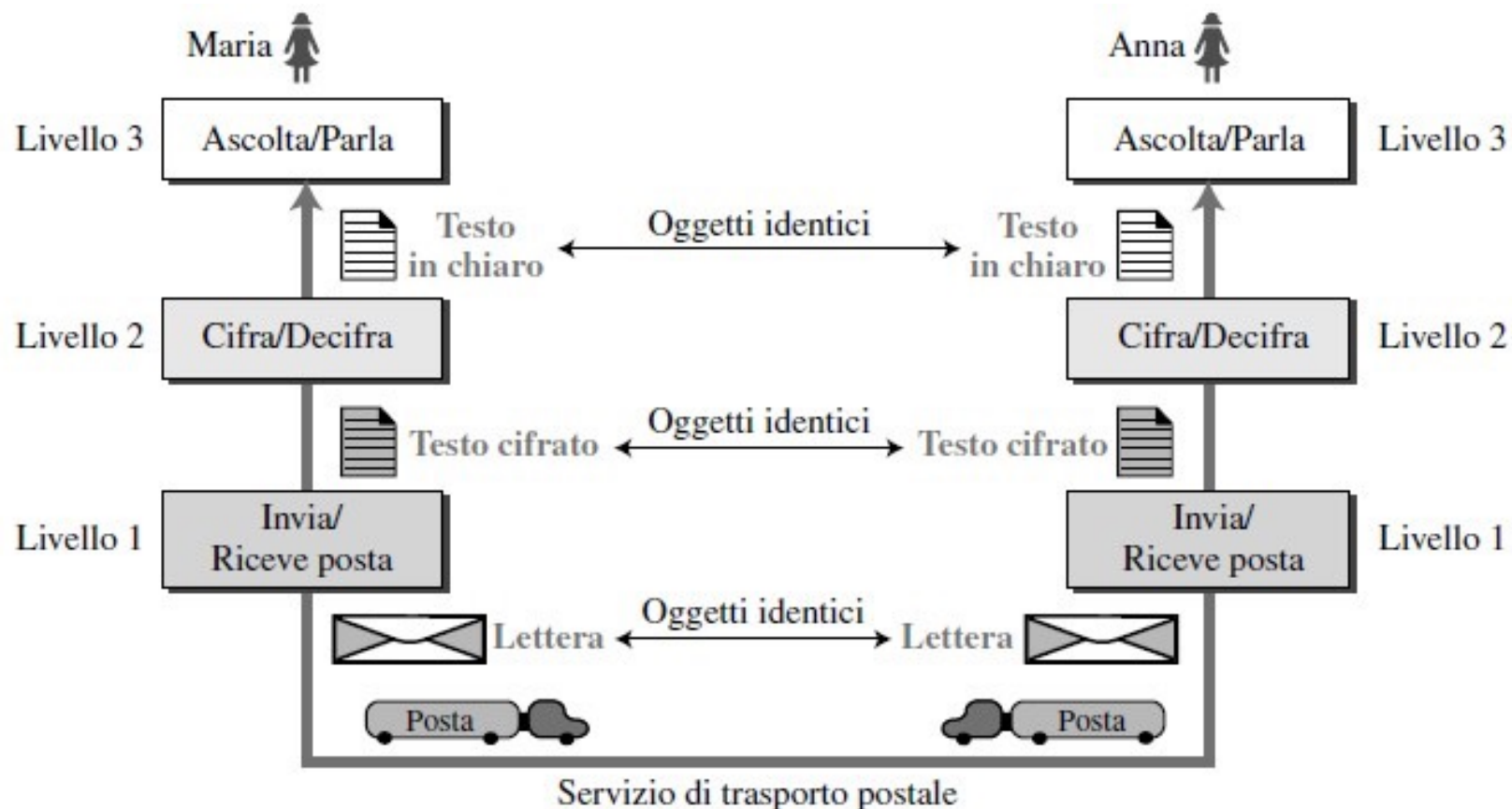
strati: ogni livello implementa un servizio

- tramite le proprie azioni interne al livello
- usando i servizi forniti dal livello sottostante

D: descrivere a parole il servizio fornito a ogni livello superiore

Esempio: Organizzazione a tre livelli

- Si ipotizza che le due amiche abbiano tre macchine ciascuna per portare a termine i compiti di ciascun livello
- Supponiamo che Maria invii la prima lettera.
- Maria comunica con la macchina al terzo livello come se fosse Anna e la potesse ascoltare



Strutturazione a livelli

La strutturazione dei protocolli in livelli consente di suddividere un compito complesso in compiti più semplici

Si potrebbe usare una sola macchina ma cosa accadrebbe se le due amiche decidessero di cambiare tecnica di crittografia?

- Nel caso delle 3 macchine verrebbe sostituita solo quella intermedia
→ *Modularizzazione* (indipendenza dei livelli)
- Un modulo (livello) può essere considerato come un black box con **opportuni** ingressi e uscite, senza preoccuparsi delle modalità con cui i dati in ingresso vengano trasformati nei dati di uscita
- Se due macchine forniscono lo stesso output dato il medesimo input allora possono essere considerate equivalenti → Le macchine possono essere acquistate da fornitori diversi
- Separazione tra servizi e implementazione: un livello usa servizi dal livello inferiore e offre servizi al livello superiore – indipendentemente da come sia implementato

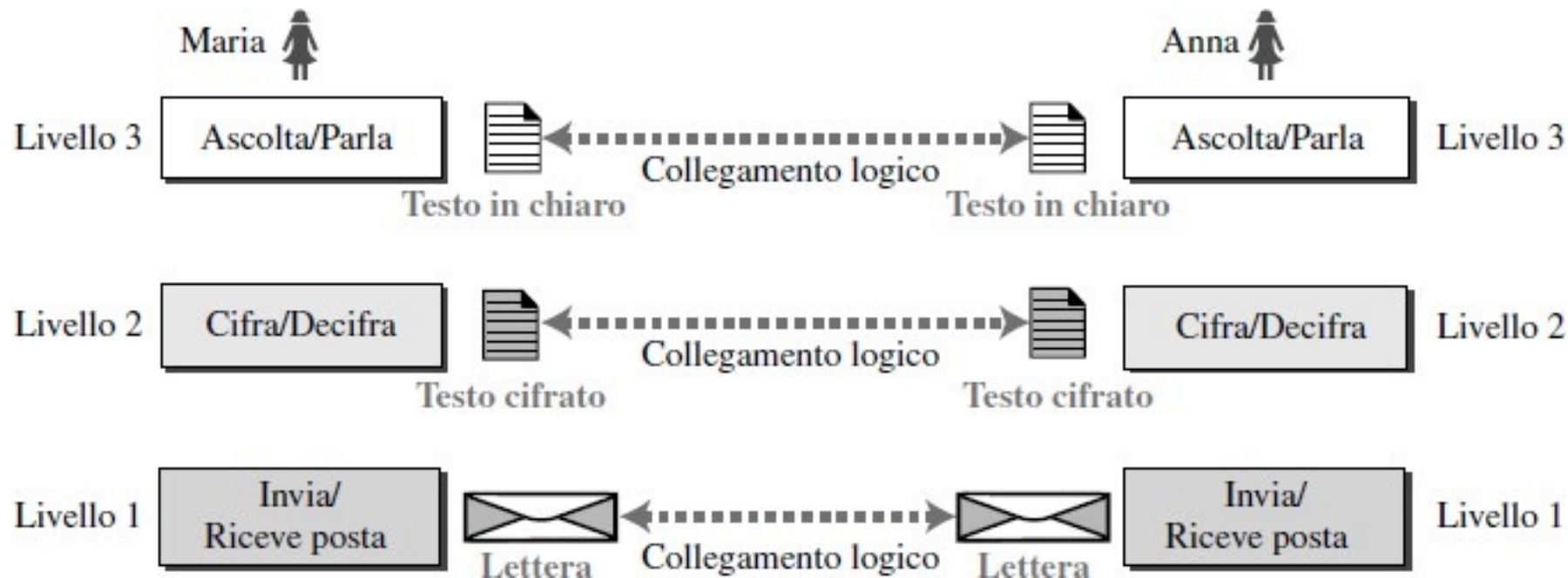
Principi della strutturazione a livelli

Quando è richiesta una comunicazione *bidirezionale*, ciascun livello deve essere capace di effettuare i due compiti opposti, uno per ciascuna direzione (es., crittografare, decrittografare)

Gli oggetti in *input/output* sotto ciascun livello di entrambi i lati devono essere *identici* (es. sotto il livello 2 c'è una lettera cifrata)

Collegamento logico fra i livelli

- Collegamento logico: i livelli logicamente sono direttamente collegati, ovvero il protocollo implementato a ciascun livello specifica una comunicazione diretta fra i pari livelli delle due parti



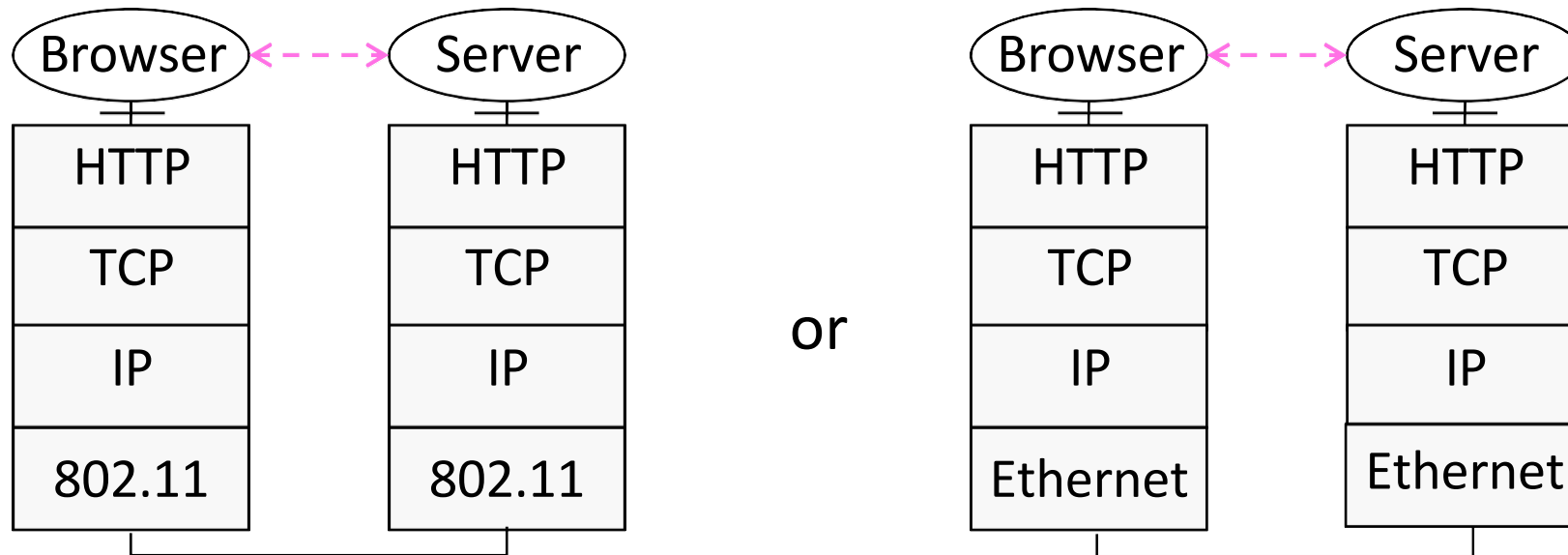
Perché stratificare?

gestione di sistemi complessi:

- la struttura esplicita permette l'identificazione delle relazioni tra le parti di un sistema complesso
 - *modello di riferimento* stratificato per la descrizione del sistema
- la modularizzazione facilita la manutenzione e l'aggiornamento del sistema
 - cambiamento nell'implementazione del servizio di un layer: trasparente al resto del sistema
 - cambio del servizio di imbarco non influisce sul resto del sistema (ad es. su come l'aereo decolla)
- **D:** esempi di stratificazione in altri sistemi complessi?

Layering: vantaggi


Riuso per gestire eterogeneità



Layering: svantaggi

- A volte necessario scambio di informazioni tra livelli non adiacenti (esempio: per ottimizzare app funzionante su wireless) non rispettando principio della stratificazione
- Sacrificio efficienza?
 - ridondanza
 - overhead

Layer del protocollo Internet

- **applicazione**: supporto delle applicazioni di rete
 - IMAP, SMTP, HTTP
- **trasporto**: trasferimento di dati processo-processo
 - TCP, UDP
- **rete**: instradamento dei datagrammi dalla sorgente alla destinazione
 - IP, protocolli di instradamento 
- **collegamento**: trasferimento di dati tra elementi di rete vicini
 - Ethernet, 802.11 (Wi-Fi), PPP
- **fisico**: bit sul canale fisico



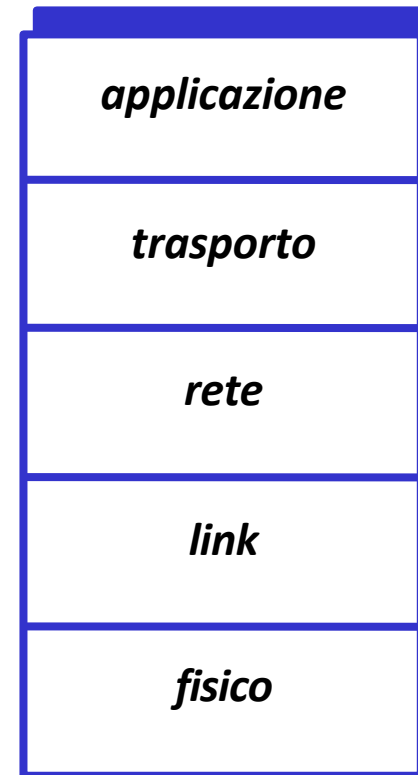
Pila di protocolli TCP/IP (stack protocollare, protocol stack)

Applicazione: è la sede delle applicazioni di rete

- HTTP, SMTP, FTP, DNS
- I pacchetti sono denominati ***messaggi***

Trasporto: trasferimento dei messaggi da/a livello di applicazione tra il modulo client e server di un'applicazione

- TCP, UDP
- I pacchetti sono denominati ***segmenti***



Pila di protocolli TCP/IP

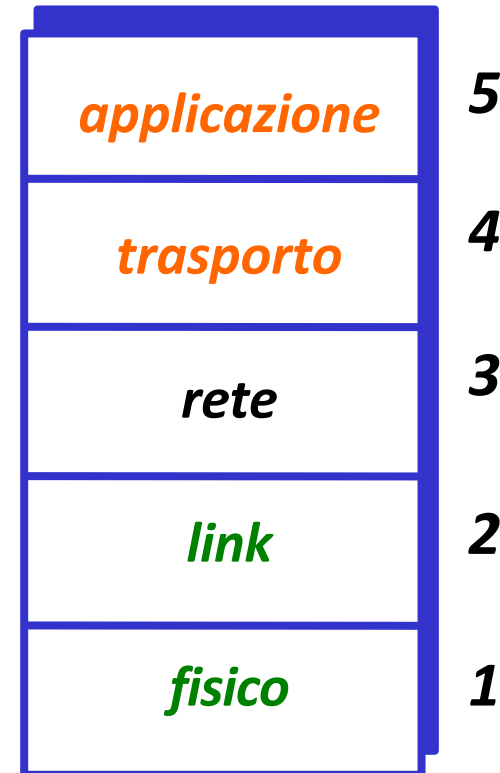
Rete: instradamento dei segmenti dall'origine alla destinazione

- IP, protocolli di instradamento
- I pacchetti sono denominati ***datagrammi***

Link (collegamento): trasmettere datagrammi da un nodo a quello successivo sul percorso

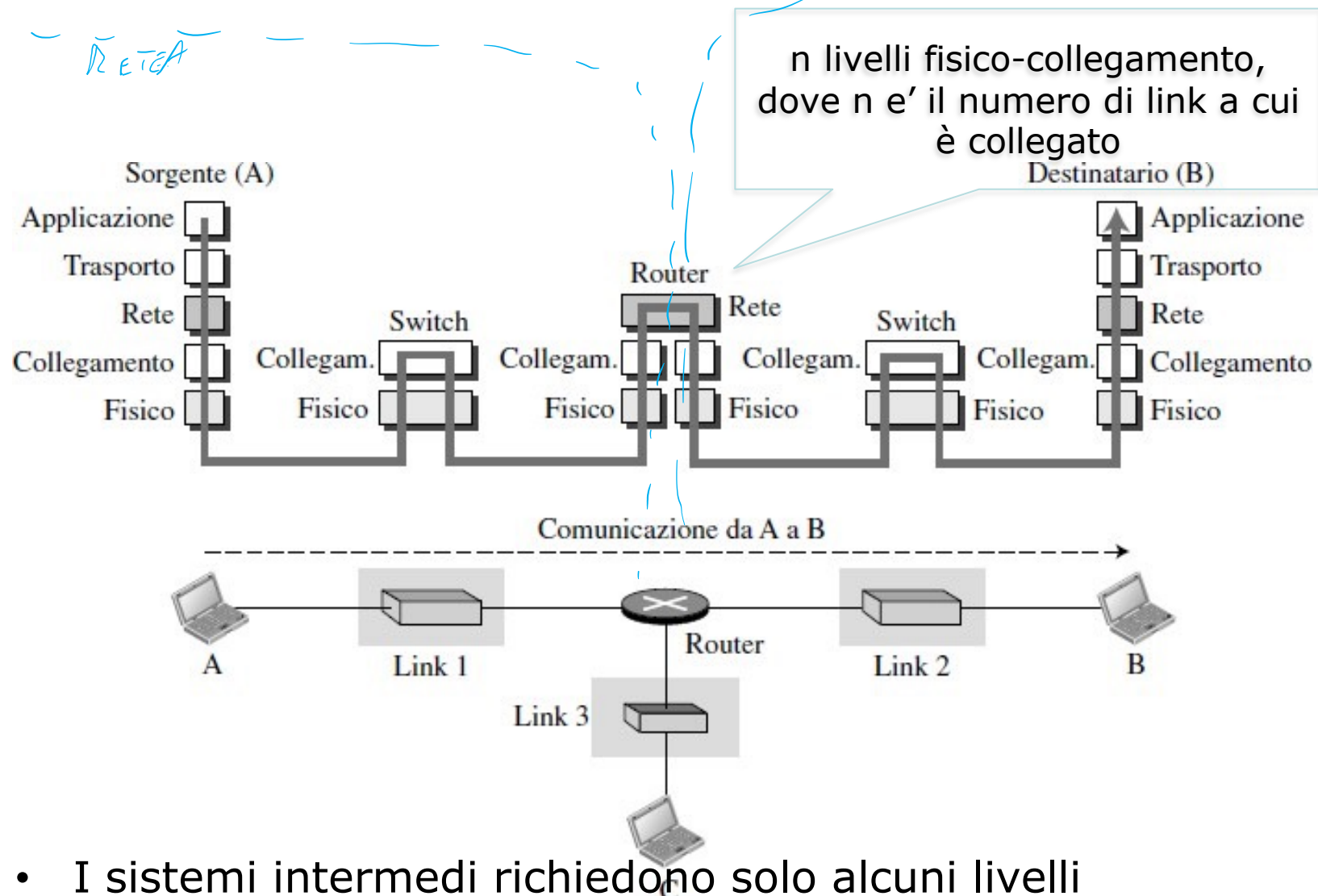
- Ethernet, Wi-Fi, PPP
- Lungo un percorso sorgente-destinazione un datagramma può essere gestito da protocolli diversi
- I pacchetti sono denominati ***frame***

fisico: trasferimento dei singoli bit

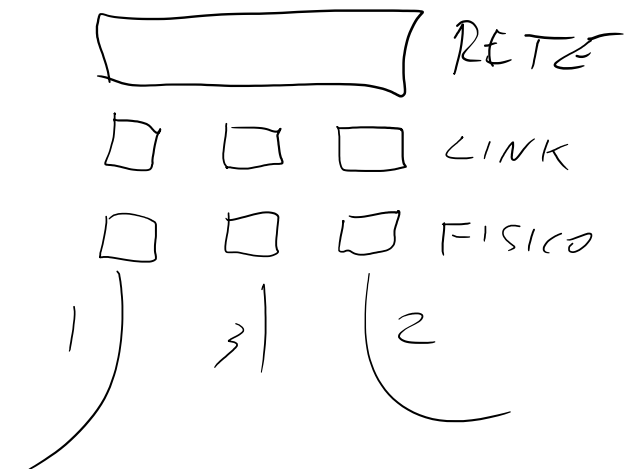


Software
Hardware

Comunicazione in una internet



- I sistemi intermedi richiedono solo alcuni livelli
- Grazie al layering tali sistemi implementano solo i livelli necessari, riducendo la complessità

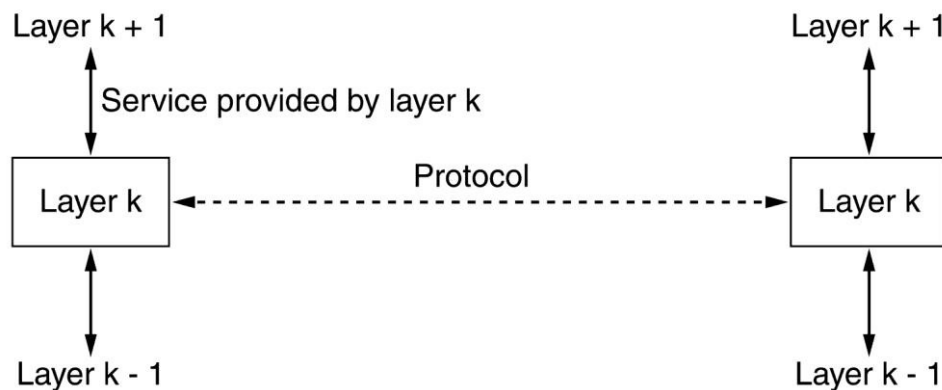


Gerarchia di protocolli

- La rete è organizzata come pila di **strati**(layer) o **livelli**, costruiti l'uno sull'altro
- Lo scopo di ogni strato è quello di **offrire determinati servizi** agli strati di livello superiore, nascondendo i dettagli di implementazione
- Lo strato N di un computer è in comunicazione (logica/virtuale) con lo strato N di un altro computer
- Le regole e le convenzioni usate in questa comunicazione sono globalmente note come i **protocolli** dello strato N
- Le entità che formano gli strati sono chiamati **pari**(peer)
- I pari comunicano usando il protocollo
- I dati non sono trasferiti direttamente dallo strato N di un computer allo strato N di un altro computer !!!

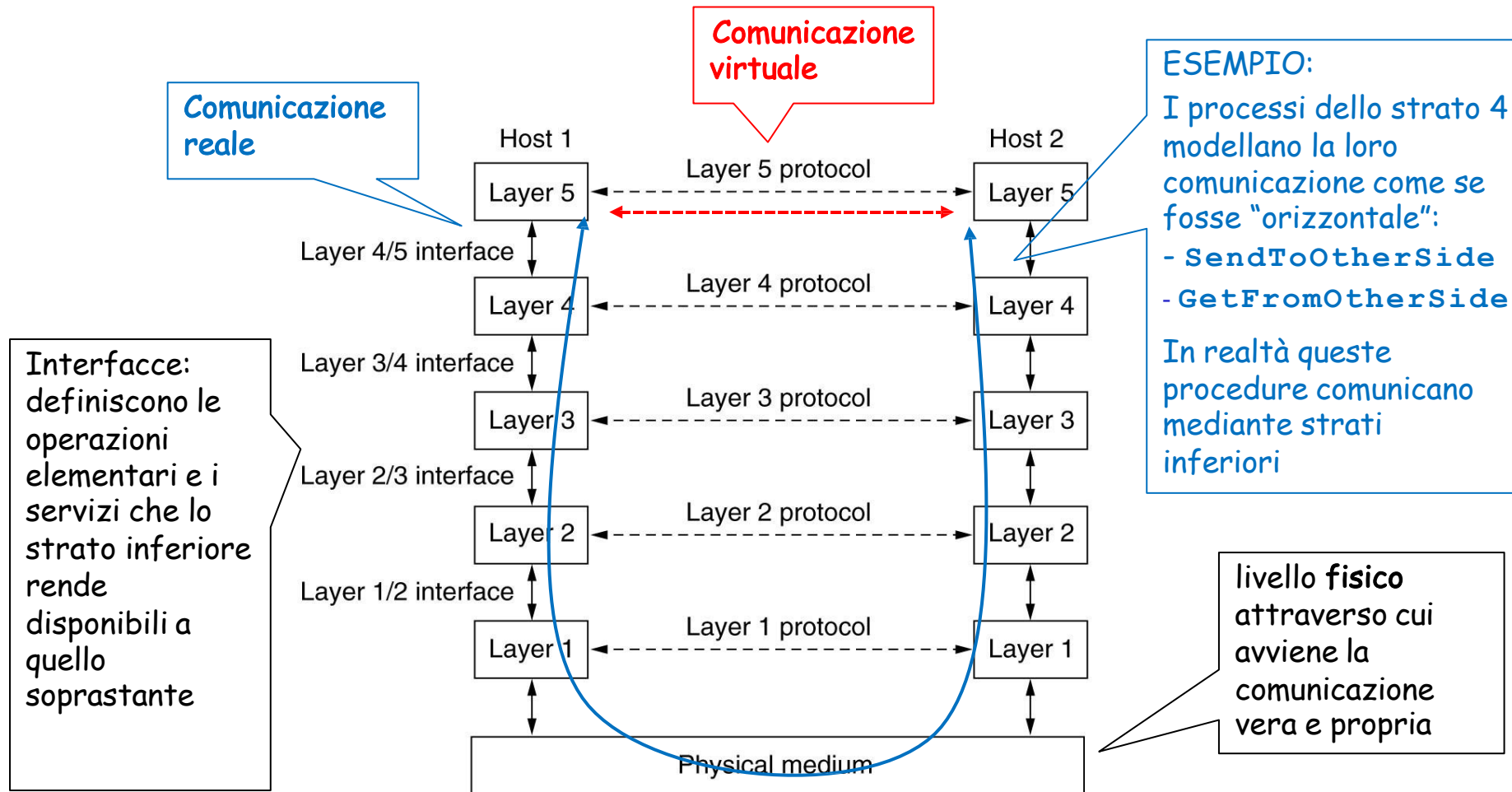
Servizi e protocolli

- Sono concetti ben distinti
- Un servizio è un insieme di primitive che uno strato offre a quello superiore
 - ❖ Definisce quali operazioni lo strato è in grado di offrire, ma non dice nulla di come queste operazioni sono implementate
 - ❖ È correlato all'interfaccia tra due strati, dove quello inferiore è il provider del servizio, mentre quello superiore è l'utente
- Un protocollo è un insieme di regole che controllano il formato e il significato dei pacchetti, o messaggi scambiati tra le entità pari all'interno di uno strato



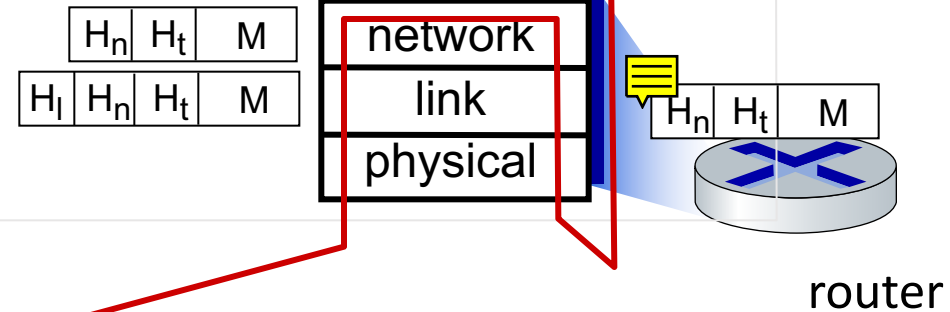
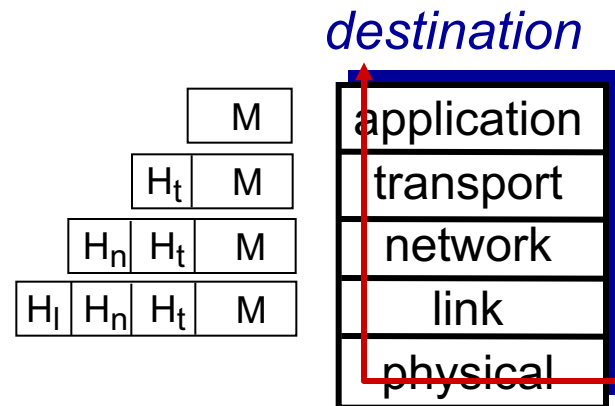
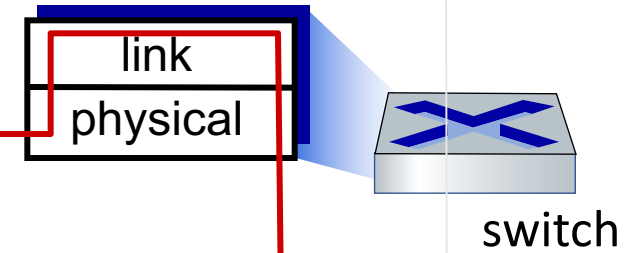
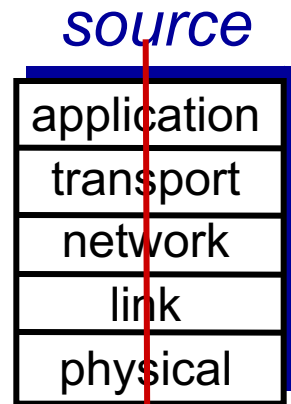
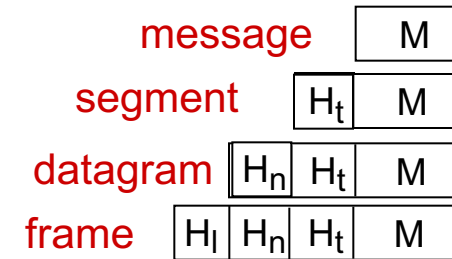
Architettura di rete

- Ogni strato passa dati e informazioni di controllo allo strato immediatamente **sottostante**, fino a raggiungere quello più basso



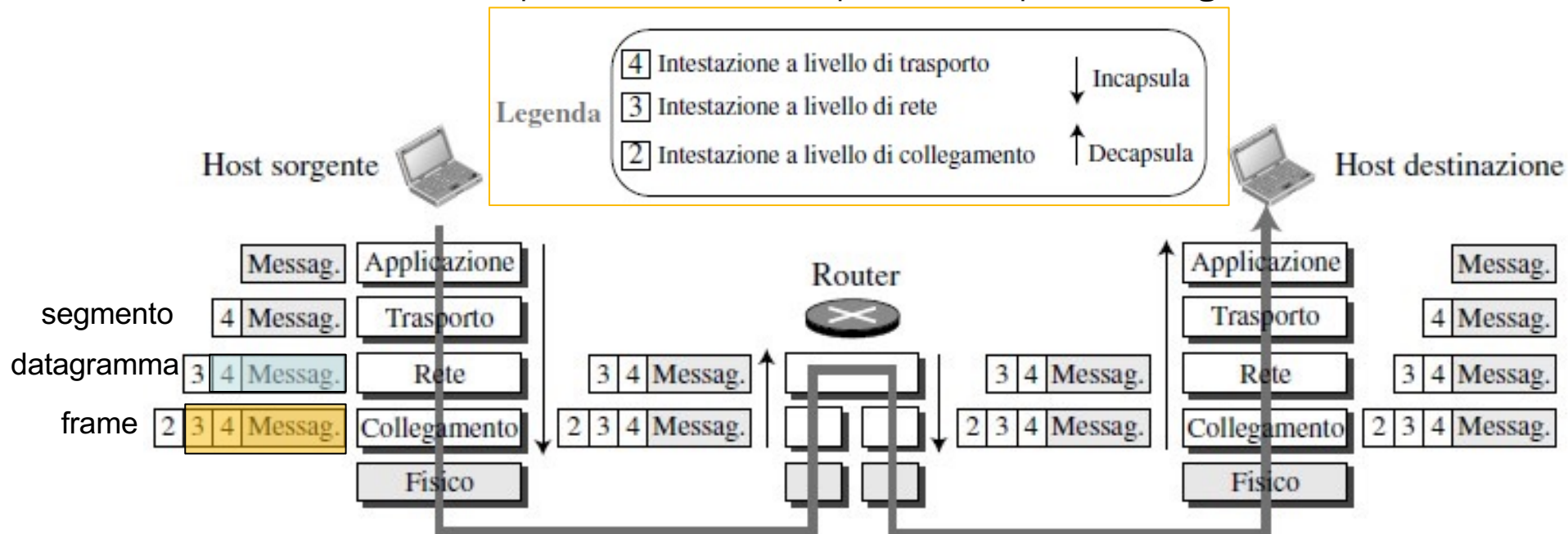
Incapsulamento

Incapsulamento/ decapsulamento



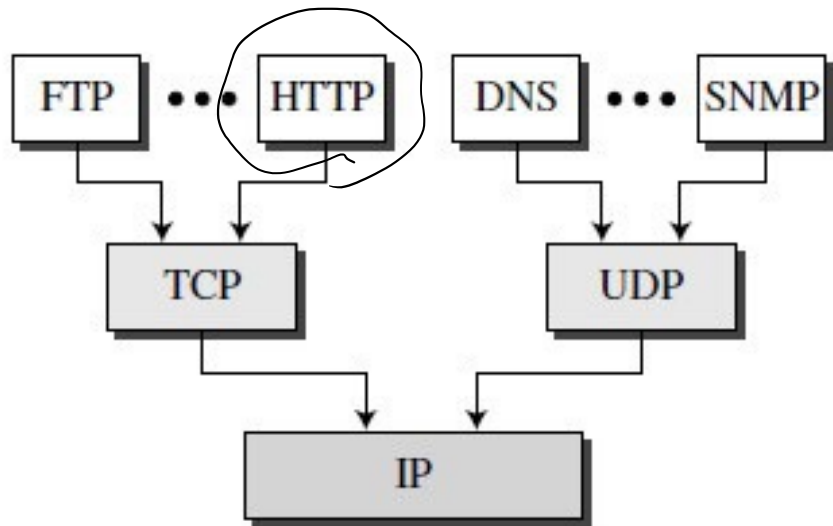
Incapsulamento e decapsulamento

- La sorgente effettua l'incapsulamento (prende il pacchetto dal livello superiore, lo considera come carico dati o payload e aggiunge un header o intestazione).
- **Messaggio** (nessuna intestazione)
- **Segmento** o **datagramma utente** = header trasporto + messaggio
- **Datagramma** = header rete + segmento
- **Frame** = header collegamento + datagramma
- Il destinatario effettua il decapsulamento
- Il router effettua sia incapsulamento che decapsulamento perché collegato a due link

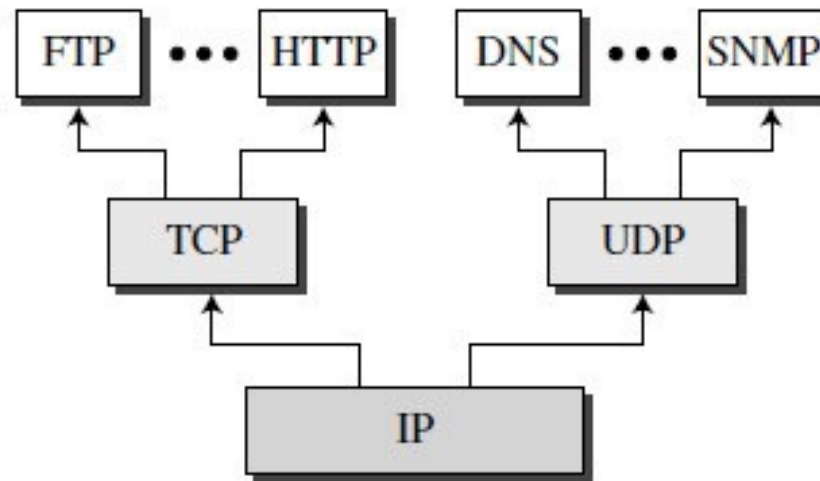


Multiplexing e demultiplexing

- Dato che lo stack protocollare TCP/IP prevede più protocolli nello stesso livello, è necessario eseguire il multiplexing alla sorgente e il demultiplexing alla destinazione
- Multiplexing: un protocollo può incapsulare (uno alla volta) i pacchetti ottenuti da più protocolli del livello superiore
- Demultiplexing: un protocollo può decapsulare e consegnare i pacchetti a più protocolli del livello superiore
- E' necessario un campo nel proprio header per identificare a quale protocollo appartengano i pacchetti incapsulati

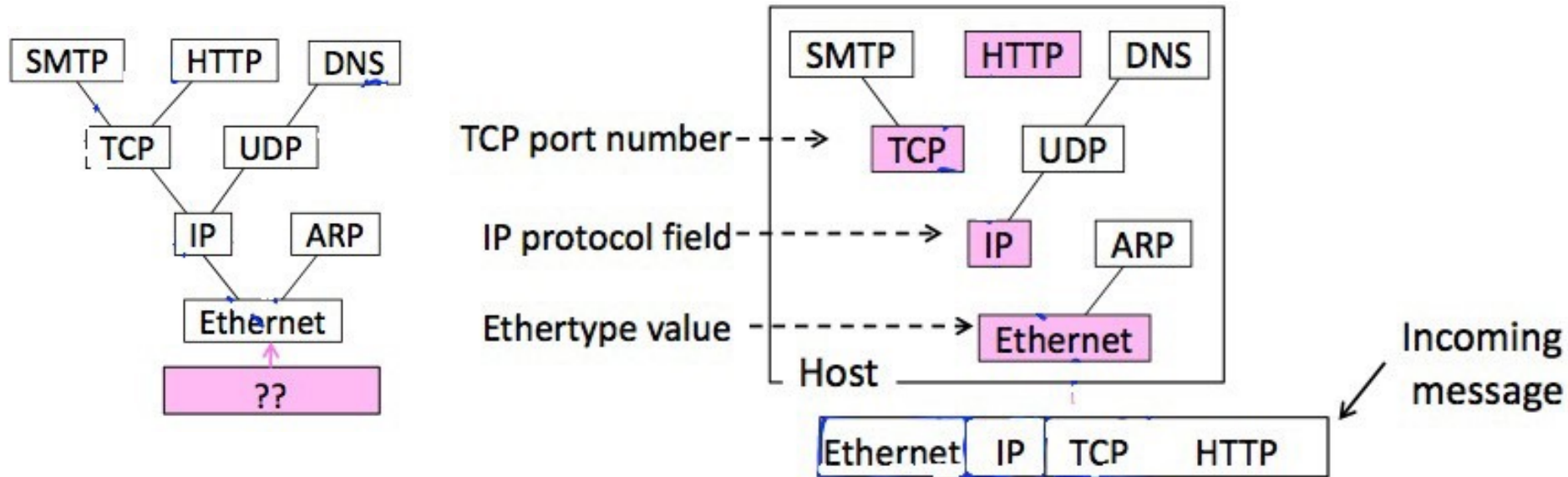


a. Multiplexing alla sorgente



b. Demultiplexing alla destinazione

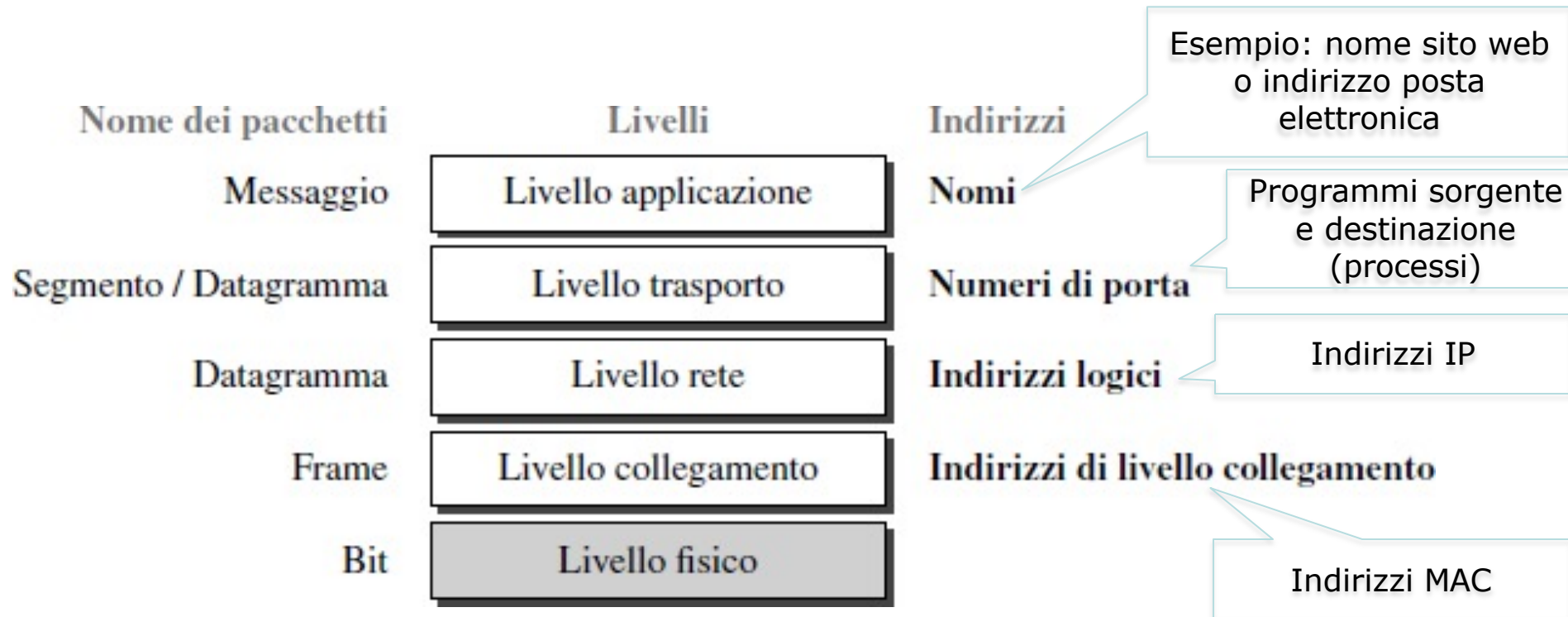
Demultiplexing



- Per poter effettuare le operazioni di multiplexing e demultiplexing, ogni pacchetto deve avere un campo all'interno dell'header per identificare a quale protocollo appartiene

Indirizzamento nel modello TCP/IP

- Poiché il modello TCP/IP prevede una comunicazione logica tra coppie di livelli è necessario avere un indirizzo sorgente e un indirizzo destinazione ad ogni livello



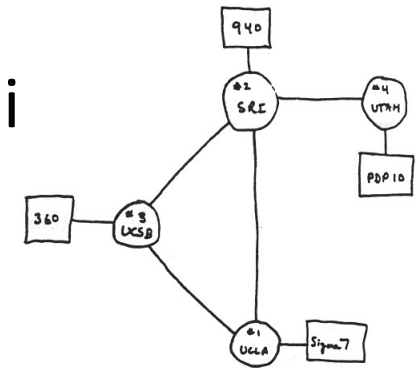
Capitolo 1: sommario

- *Cos'è Internet ?*
- *Cos'è un protocollo?*
- Periferia della rete: host, rete di accesso, supporti fisici
- Nucleo di rete: commutazione pacchetto/circuito, struttura internet
- Prestazioni: bitrate, bandwidth, throughput, loss, latency
- Sicurezza
- Livelli di protocollo, modelli di servizio
- **Storia**

Storia di Internet

1961-1972: Primi principi di commutazione di pacchetto

- **1961:** Kleinrock - la teoria delle code mostra l'efficacia della commutazione di pacchetto
- **1964:** Baran – packet switching nelle reti militari (indipend)
- **1967:** ARPAnet concepito da Advanced Research Projects Agency
- **1969:** primo nodo ARPAnet operativo
- **1972:**
 - Demo pubblica di ARPAnet
 - NCP (Network Control Protocol) primo protocollo host-host
 - primo programma di posta elettronica
 - ARPAnet ha 15 nodi



THE ARPA NETWORK

Storia di Internet

1972-1980: Internetworking, reti nuove e proprietarie

- **1970:** Rete wireless ALOHAnet Hawaii
- **1974:** Cerf e Kahn - architettura per reti di interconnessione
- **1976:** Ethernet allo Xerox PARC
- **fine anni '70 :** architetture proprietarie: DECnet , SNA, XNA
- **fine anni '70 :** commutazione di pacchetti a lunghezza fissa (precursore ATM)
- **1979:** ARPAnet ha 200 nodi

Principi di internetworking di Cerf e Kahn :

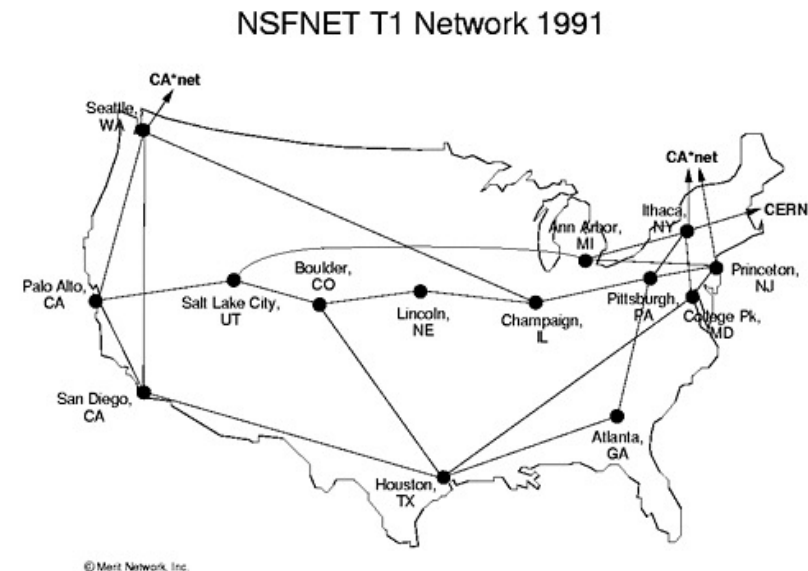
- minimalismo, autonomia - nessuna modifica interna richiesta per interconnettere le reti
- modello di servizio best-effort
- instradamento senza stato
- controllo decentrato

basi per l'architettura odierna di Internet

Storia di Internet

1980-1990: nuovi protocolli, proliferazione di reti

- **1983:** diffusione del TCP/IP
- **1982:** definizione del protocollo di posta elettronica smtp
- **1983:** DNS definito per la traduzione da nome a indirizzo IP
- **1985:** protocollo ftp
- **1988:** controllo della congestione in TCP
- nuove reti nazionali: CSnet, BITnet, NSFnet, Minitel
- 100.000 host connessi alla confederazione di reti



Storia di Internet

1990-2000: commercializzazione, Web, nuove applicazioni

- primi anni '90: ARPAnet disattivato
- 1991: NSF revoca le restrizioni sull'uso commerciale di NSFnet (smantellato, 1995)
- primi anni '90: web
 - hypertext [Bush 1945, Nelson 1960]
 - HTML, HTTP: Berners-Lee
 - 1994: Mosaic, poi Netscape
 - fine anni '90 : commercializzazione del Web
- fine anni '90 - anni 2000:
 - killer apps: messaggistica istantanea, condivisione di file P2P
 - sicurezza di rete all'avanguardia
 - stima 50 milioni di host, oltre 100 milioni di utenti
 - collegamenti backbone a Gbps

Storia di Internet

2005-presente: nuove applicazioni, Internet è "ovunque"

- ~18 miliardi di dispositivi collegati a Internet (2017)
 - ascesa degli smartphone (iPhone: 2007)
- 2008: Software Defined Networks (SDN)
- diffusione aggressiva dell'accesso a banda larga
- crescente ubiquità dell'accesso wireless ad alta velocità: 4G/5G, Wi-Fi
- nascita dei social network online: Facebook: ~2,5 miliardi di utenti
- i fornitori di servizi (Google, FB, Microsoft) creano le proprie reti
 - Bypassare l'Internet commerciale per connettersi "vicino" all'utente finale, fornendo accesso " istantaneo" ai contenuti
- le aziende gestiscono i loro servizi in "cloud" (ad esempio, Amazon Web Services, Microsoft Azure)