



**UNIVERSITÀ  
DEGLI STUDI  
DI UDINE**  
hic sunt futura

TECNOLOGIE WEB E LABORATORIO (a.a. 2022-2023)

# INTERNET - PARTE 1

**Daniele Salvati**

Dipartimento di Scienze Matematiche, Informatiche e Fisiche  
Università degli Studi di Udine

# Informazioni slide

- Il materiale contenuto in queste slide è riservato esclusivamente agli studenti del corso di **Tecnologie Web e Laboratorio** del Corso di Studio in **Internet of Things, Big Data, Machine Learning** dell'Università degli Studi di Udine.
- Non è consentita la diffusione del materiale contenuto in queste slide, ma solo l'utilizzo inerente la preparazione dell'esame del suddetto corso.

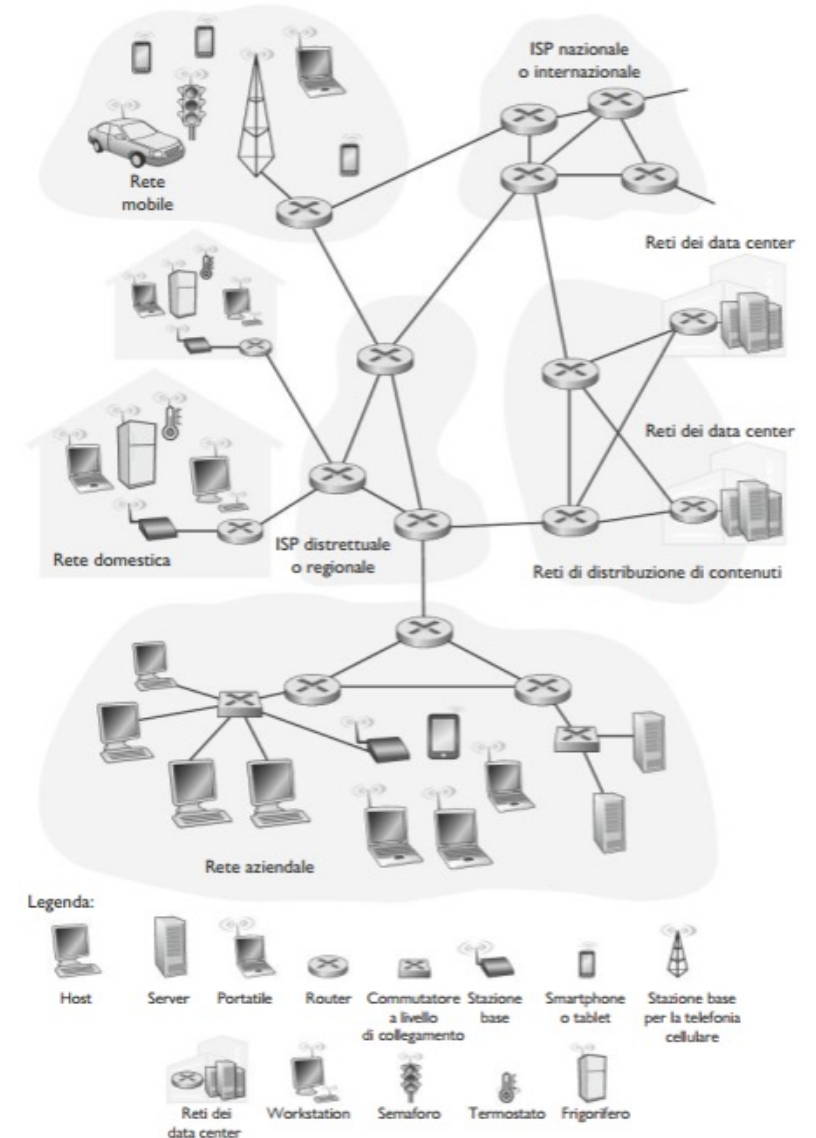
Cosa è Internet?

# Internet

- **Internet** è un sistema di **reti di calcolatori** (una rete di reti).
- Si intende quindi la interconnessione di singole reti di calcolatori gestite da organizzazioni pubbliche, aziendali, di ricerca o private.
- Internet fornisce l'infrastruttura e l'ambiente Hardware e Software in cui sviluppare applicazioni e servizi.
- Il **World Wide Web** è uno dei principali servizi di Internet.

# Internet (2)

- I dispositivi di elaborazione (calcolatori) sono fondamentalmente PC tradizionali, workstation Linux o server.
- Sempre più spesso però ci si connette a Internet tramite smartphone e tablet.
- Inoltre “cose” (**things**) finora non connesse a Internet, quali TV, console di gioco, termostati, sistemi di sorveglianza, elettrodomestici, orologi, occhiali, automobili, sistemi di controllo del traffico e molte altre vengono via via connesse.
- Questi dispositivi (calcolatori) sono detti **host** (ospite) o **end system** (sistemi periferici)



# Internet (3)

- I sistemi periferici sono connessi tra loro tramite una rete di collegamenti (***communication link***) e commutatori di pacchetti (***packet switch***).
- Quando un sistema periferico vuole inviare dati a un altro sistema periferico, suddivide i dati in sotto-parti e aggiunge un'intestazione a ciascuna di esse: l'insieme delle informazioni è chiamato **pacchetto** (*packet*)
- Un commutatore di pacchetto prende un pacchetto che arriva da uno dei collegamenti in ingresso e lo ritrasmette su uno di quelli in uscita.

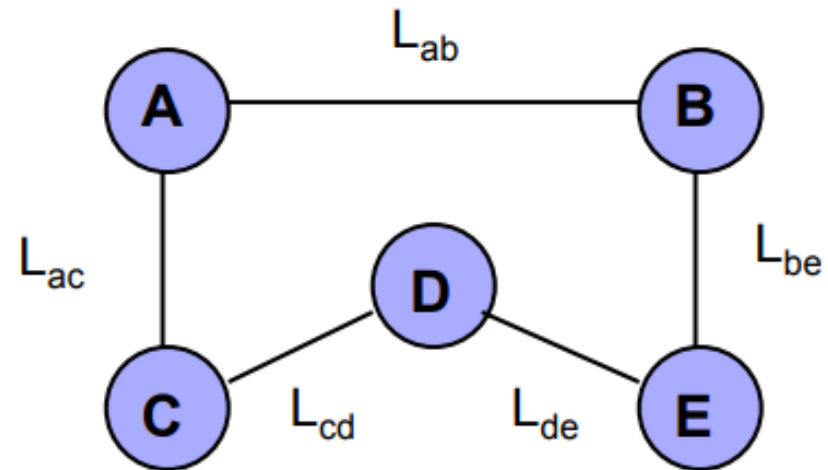
# Reti di calcolatori

- **Definizione:** Una rete di calcolatori è un insieme di nodi (i sistemi), e di linee di comunicazione che li connettono.
- Una rete può essere rappresentata da un grafo

$$G = \langle \{N_i\}; \{L_{ij}\} \rangle$$

dove  $\{N_i\}$  indica l'insieme dei nodi e  $\{L_{ij}\}$  indica l'insieme delle linee di comunicazione (i collegamenti fisici) esistenti tra coppie di nodi

## Reti di calcolatori (2)



$$Gr = \langle \{A, B, C, D, E\}; \{L_{ab}, L_{ac}, L_{cd}, L_{de}, L_{be}\} \rangle$$

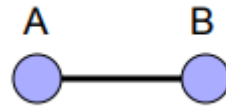


# Nodi

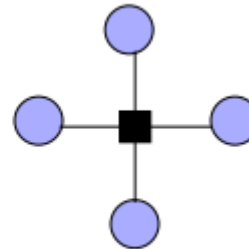
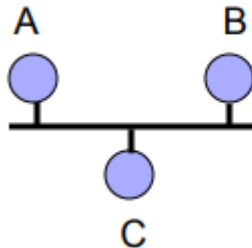
- **Sistemi periferici (*host*):** PC, server, portatile, telefono cellulare, stampanti, smart objects, sensori, ecc. (ospitano applicazioni).
- **Commutatori di pacchetti**
  - **Router** (comunemente usato nel nucleo della rete)
  - **Commutatori a livello di collegamento (*link-layer switch*)** (comunemente usato nelle reti di accesso)
- **Internet Service Provider (ISP):** fornisce accesso a Internet ai sistemi periferici.

# Linee di comunicazione

- Collegamenti **punto a punto**: collega due nodi (solo due) senza passare per un nodo intermedio.

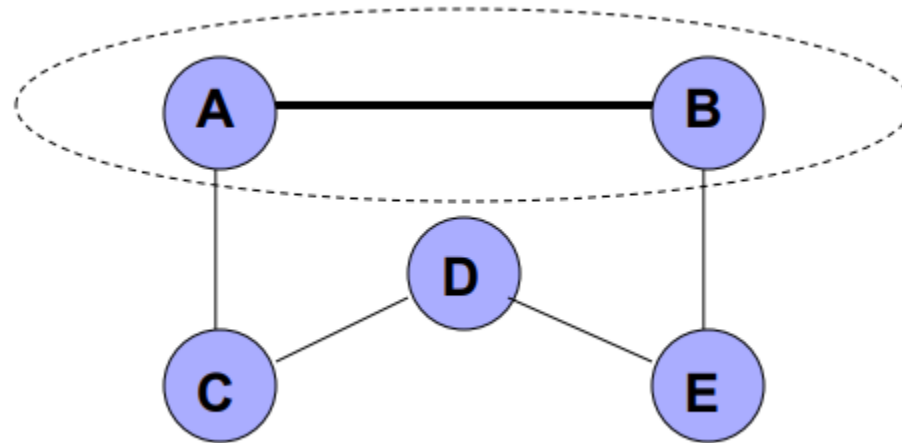


- Collegamenti **multipoint**: esiste un solo collegamento condiviso da più nodi.



# Linee di comunicazione (2)

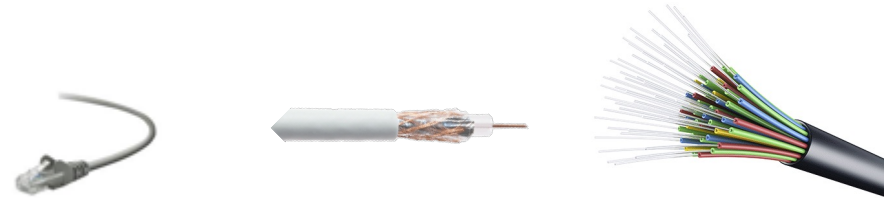
- Un **collegamento fisico** tra due nodi (es. A-B) è costituito da un link fisico permanente o temporaneo che unisce i due nodi.



# Linee di comunicazione (3)

- **Mezzi vincolati** (*guided media*): i segnali sono trasmessi in un mezzo fisico:

- doppino di rame intrecciato
- cavi coassiali
- fibra ottica



- **Mezzi non vincolati** (*unguided media*): i segnali si propagano nell'atmosfera e nello spazio esterno:

- canali radio terrestri
- canali radio satellitari

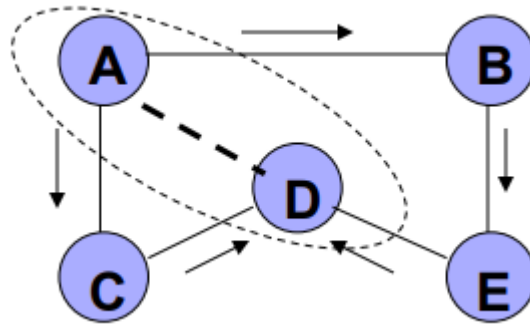


# Velocità di trasmissione

- Linee di comunicazione diverse possono trasmettere dati a velocità differenti, e tale **velocità [o tasso] di trasmissione** (*transmission rate*) è misurata in **bit/secondo (bps)**.
- Esempi:
  - doppino di rame intrecciato: 3 Mbps - 50 Mbps
  - cavi coassiali: 10 - 500+ Mbps
  - canali radio satellitari: 12 Mbps - 100 Mbps
  - wireless: 5 Mbps - 50 Mbps
  - fibra ottica: 250 Mbps - 2+ Gbps

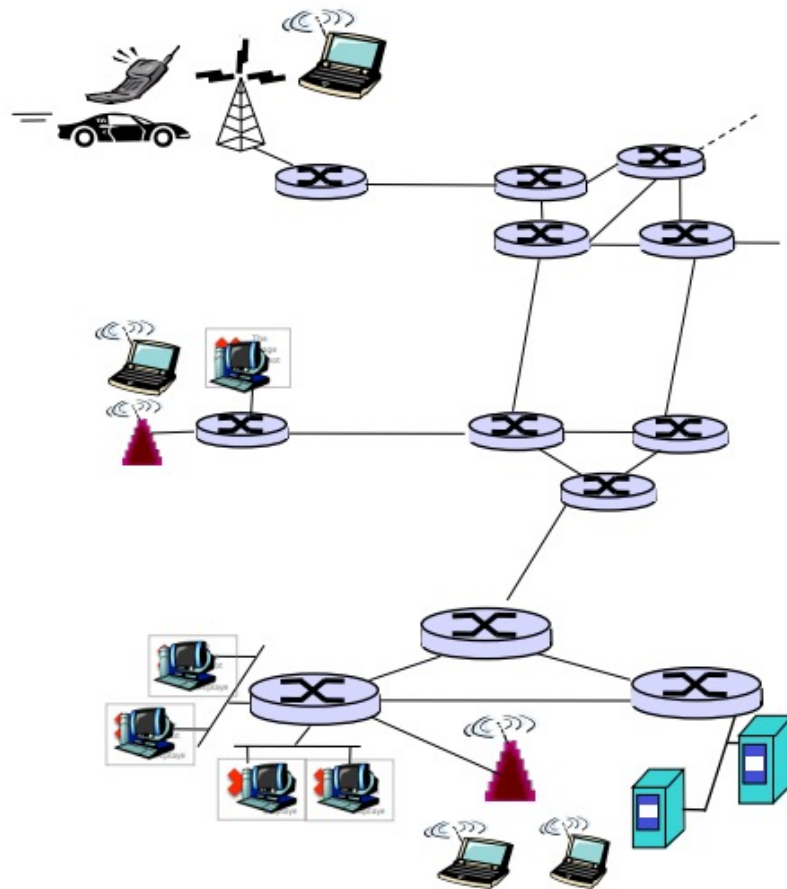
# Collegamento logico

- L'esistenza di un **collegamento logico** tra due nodi (es. **A-D**) implica che i due nodi possano comunicare tra di loro indipendentemente dal fatto che esista una connessione fisica diretta tra di essi.

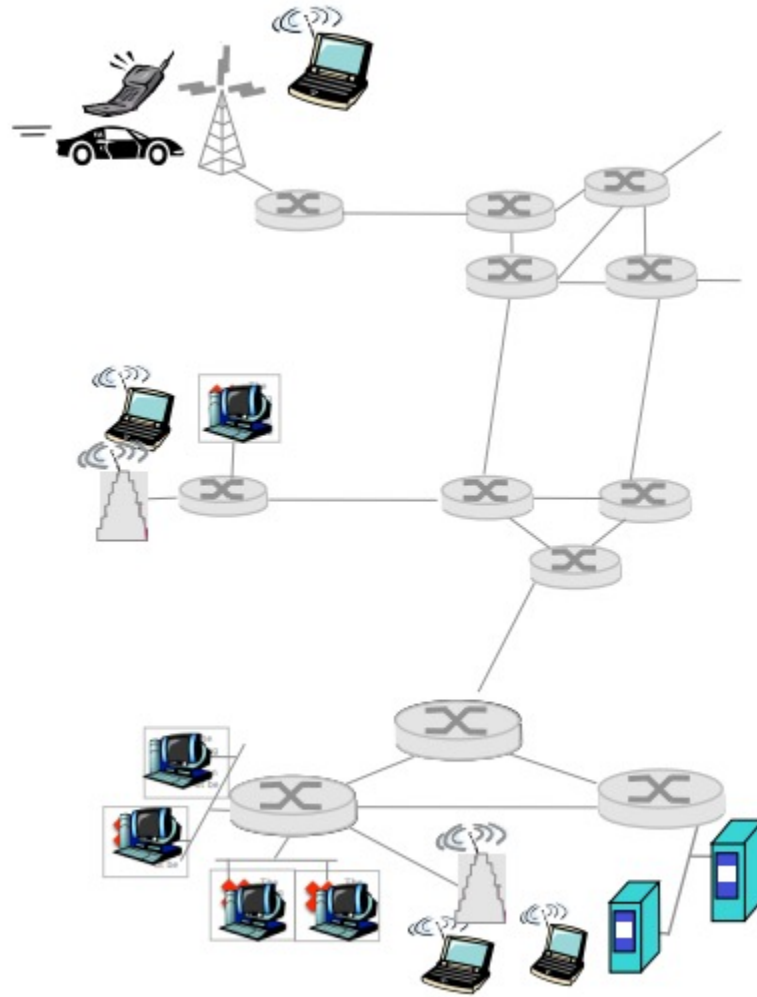


- Il collegamento logico presuppone l'esistenza di un percorso tra i due nodi (es. **A-C-D** o **A-B-E-D**).

# Una rete

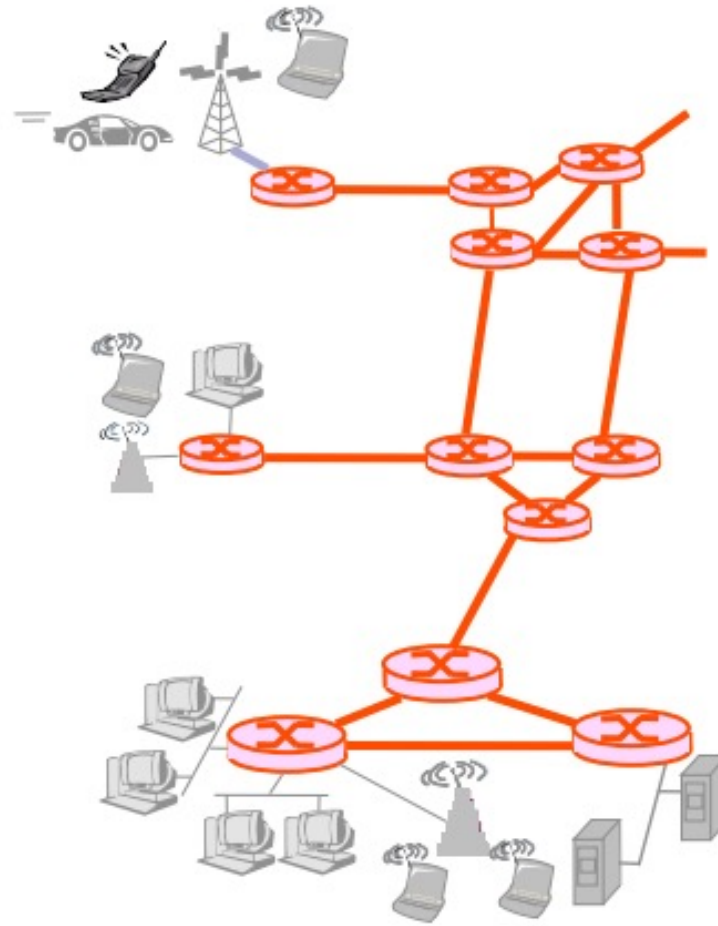


# Gli host

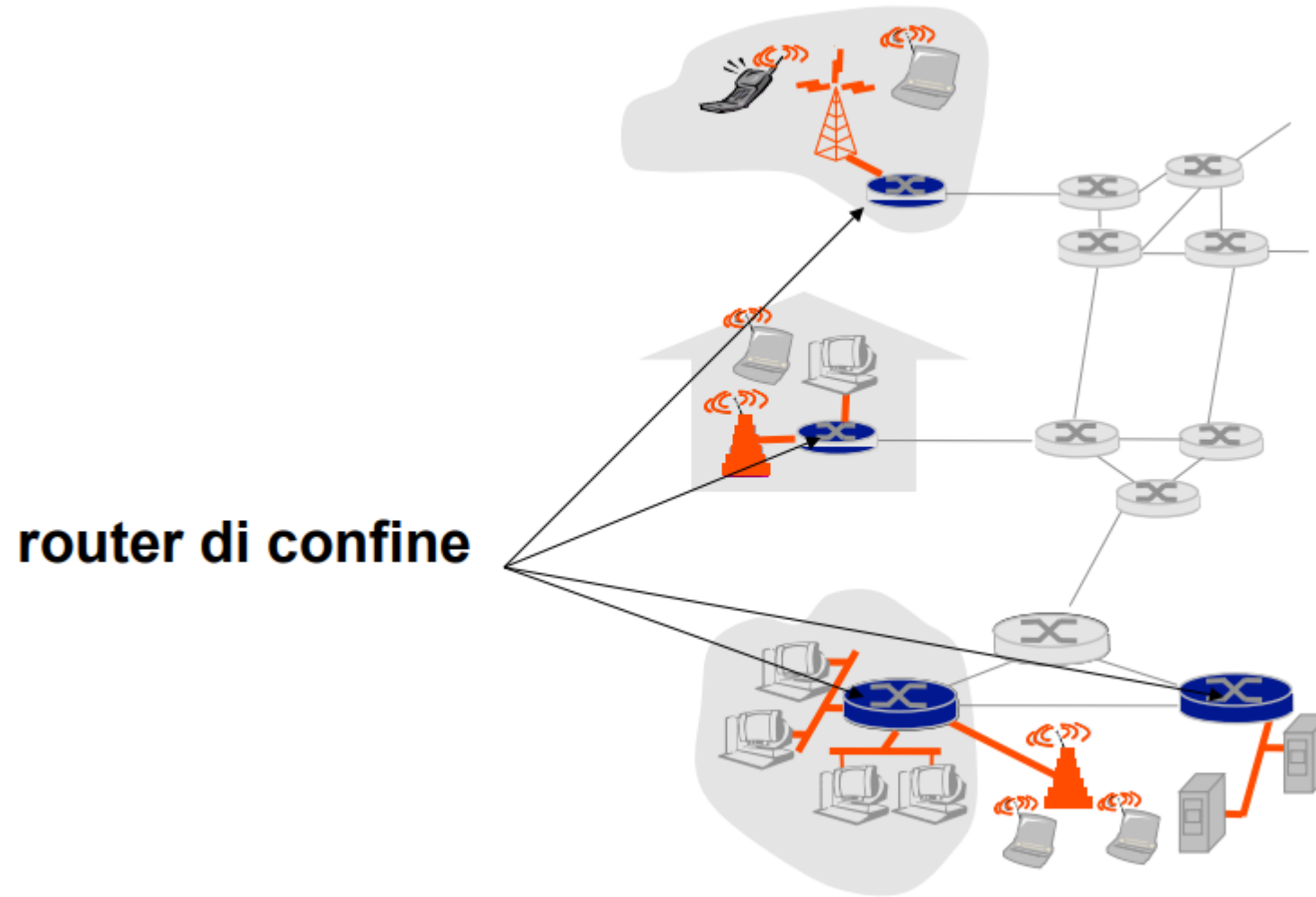




# La sottorete di commutazione



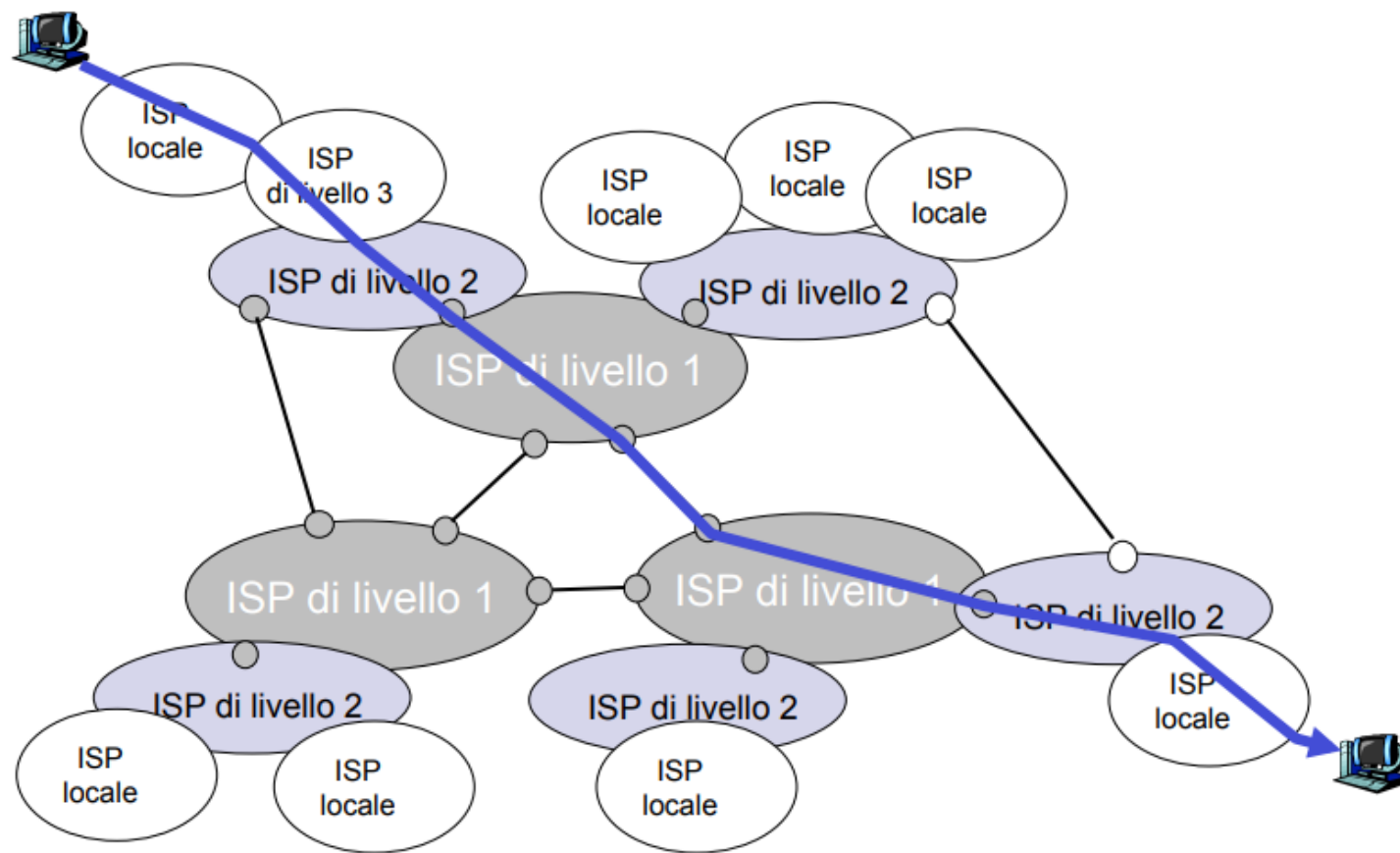
# Internet Service Provider (ISP)



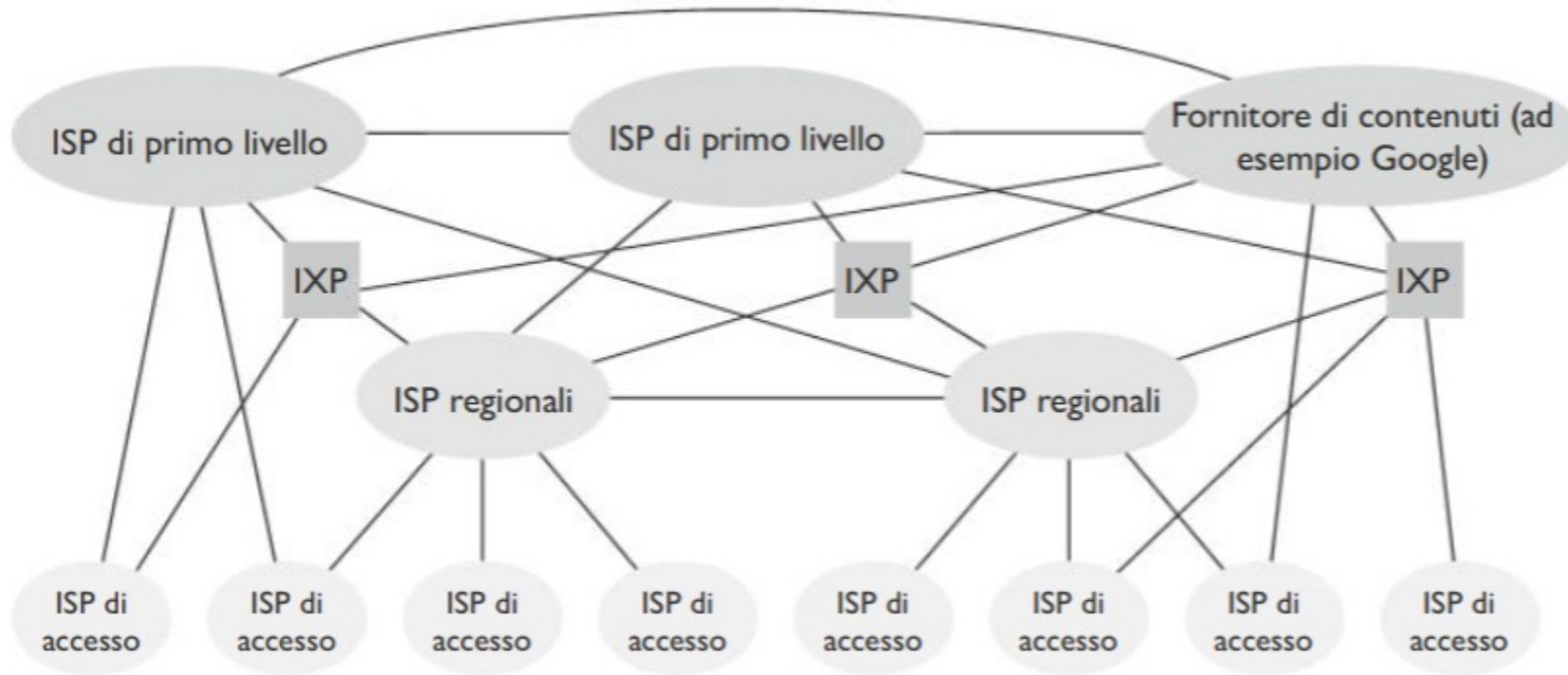
# Una rete di reti

- L'obiettivo generale è quello di interconnettere gli ISP di accesso in modo che i sistemi periferici possano scambiarsi pacchetti.
- Esiste una gerarchia di ISP (fornitori di servizi Internet):
  - **ISP di Livello 1**: reti della dorsale di Internet, copertura internazionale
  - **ISP di Livello 2**: copertura nazionale o regionale
  - **ISP di Livello 3 e ISP locali** (ISP di accesso)
- I punti in cui un ISP si collega ad un altro (sia orizzontalmente tra IPS dello stesso livello sia verticalmente tra ISP di livello diverso) sono detti punti di presenza (POP). Sono un gruppo di uno o più router.

# Una rete di reti (2)



# Una rete di reti (3)



**IXP** (Internet exchange point): un punto d'incontro dove più ISP possono fare *peering* tra di loro.

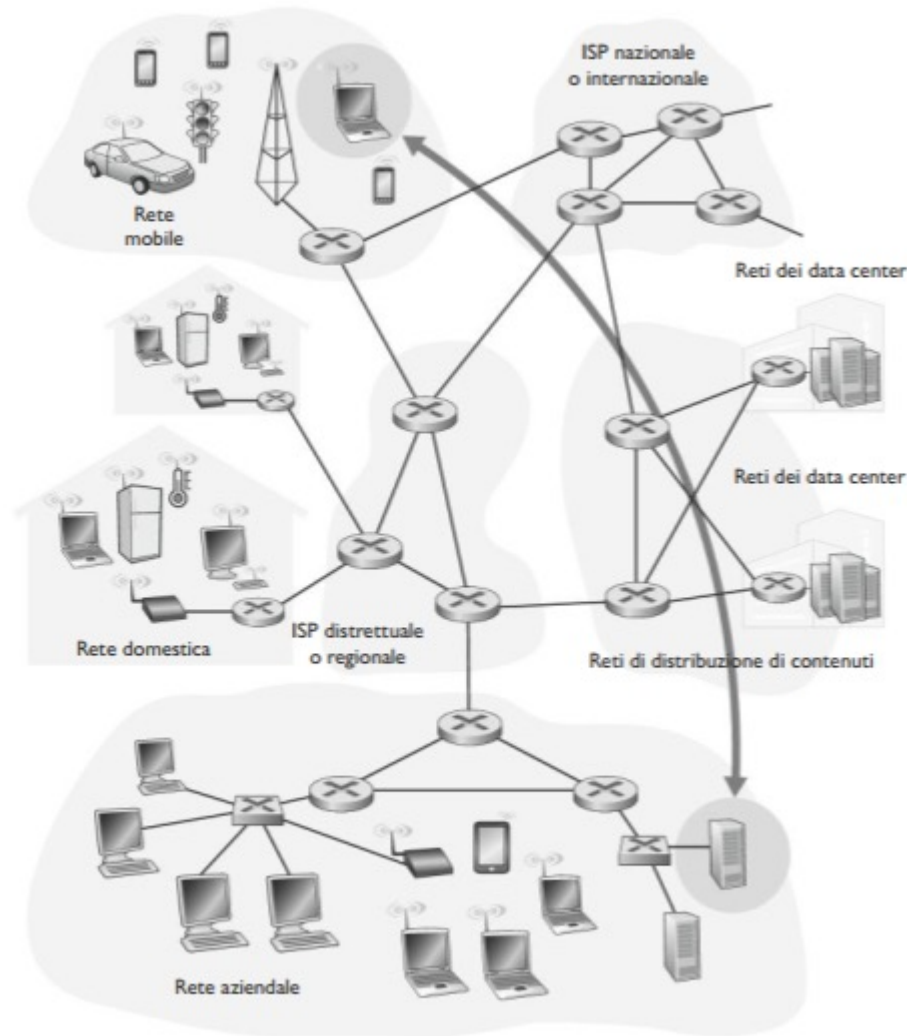
# Applicazioni distribuite

- Internet può essere definito anche da un altro punto di visto, ossia come un'**infrastruttura che fornisce servizi alle applicazione** (es. posta elettronica, navigazione sul Web, messaggistica istantanea, sistemi di navigazione con informazioni sul traffico in tempo reale, streaming di musica e video, on-line social media, ecc.)
- Queste applicazioni sono dette **applicazioni distribuite** (*distributed applications*), in quanto coinvolgono più sistemi periferici che si scambiano reciprocamente dati.

# Applicazioni distribuite (2)

- I sistemi periferici collegati a Internet forniscono una **interfaccia socket** (*socket interface*), che specifica come un programma eseguito su un sistema periferico possa chiedere a Internet di recapitare dati a un programma eseguito su un altro sistema periferico.
- L'interfaccia socket è un insieme di regole che il programma mittente deve seguire in modo che i dati siano recapitati al programma di destinazione.

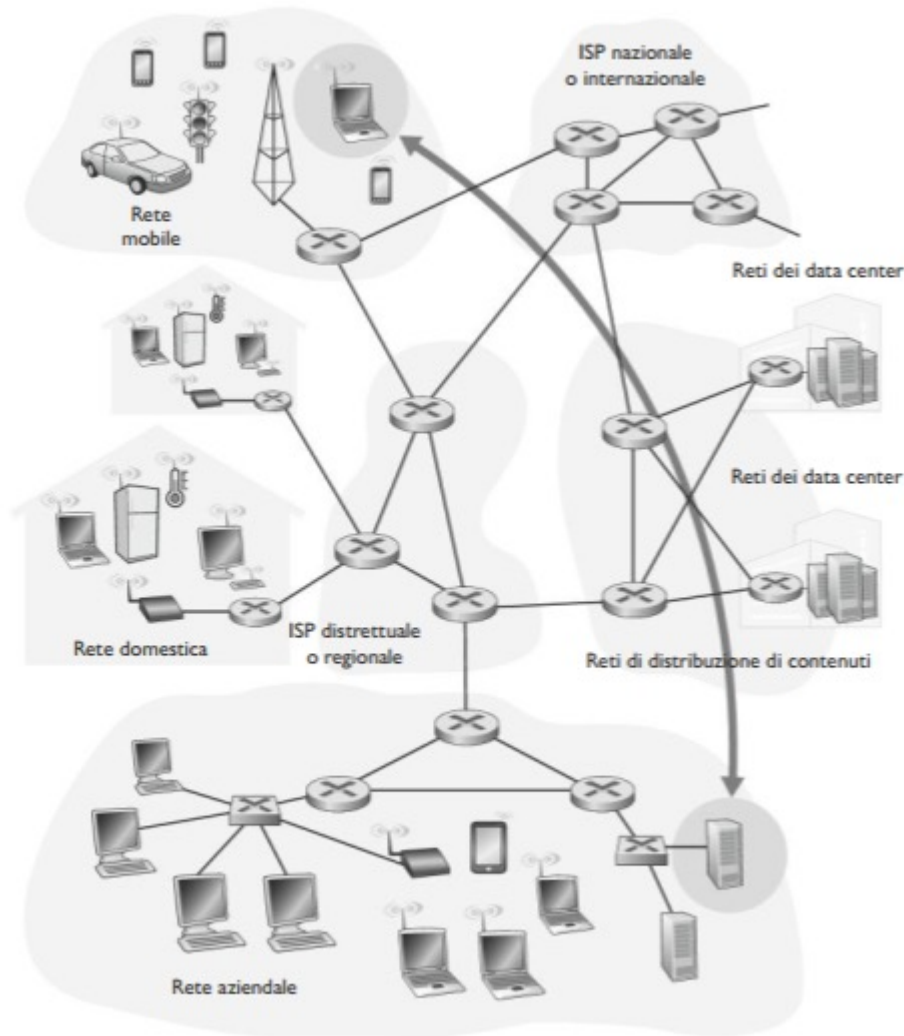
# Interazione tra sistemi periferici



In che modi possiamo trasferire un pacchetto da un sistema periferico ad un altro?



# Interazione tra sistemi periferici (2)



Definire un **insieme di regole** che definiscono le modalità di interazione tra sistemi



**Protocollo di comunicazione**

# Protocolli di rete

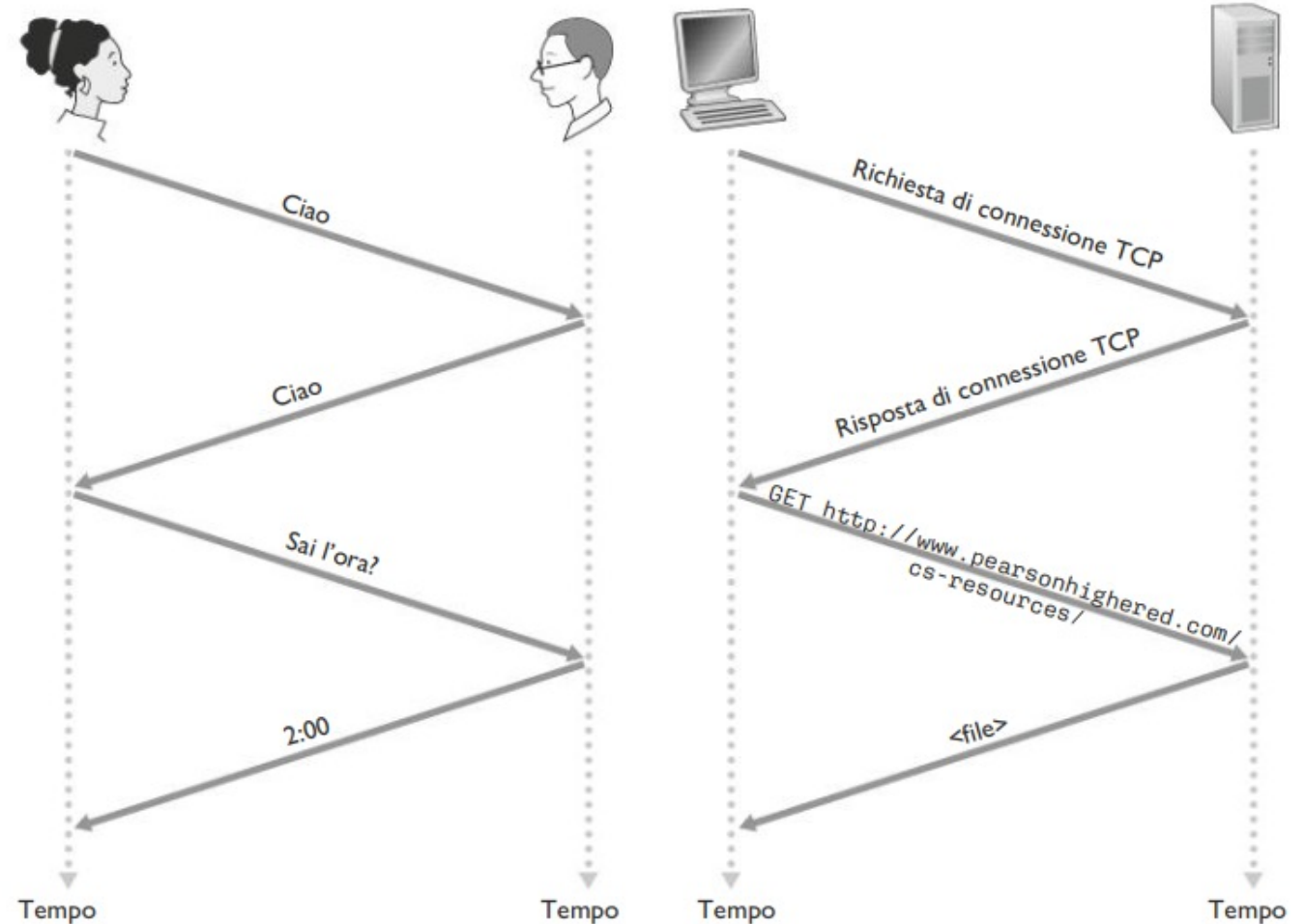
*Un **protocollo** (protocol) definisce il formato e l'ordine dei messaggi scambiati tra due o più entità in comunicazione, così come le azioni intraprese in fase di trasmissione e/o di ricezione di un messaggio o di un altro evento.*

# Protocolli di rete (2)

- Internet, e le reti di calcolatori in generale, fanno un uso estensivo di protocolli.
- Si impiegano protocolli differenti per realizzare compiti diversi.
- Padroneggiare il campo delle reti di calcolatori equivale a comprendere l'essenza, la finalità e le modalità operative dei protocolli di rete.

# Protocollo

- Che cos'è un protocollo?  
Che cosa fa?
- Lo scambio si appoggia a due (o più) entità che comunicano utilizzando lo stesso protocollo al fine di assolvere un certo compito



Protocollo umano

Protocollo di rete

# Il nucleo della rete

- Per spostare i dati in una rete di collegamenti e commutatori esistono due approcci fondamentali:
  - la commutazione di circuito
  - la commutazione di pacchetto

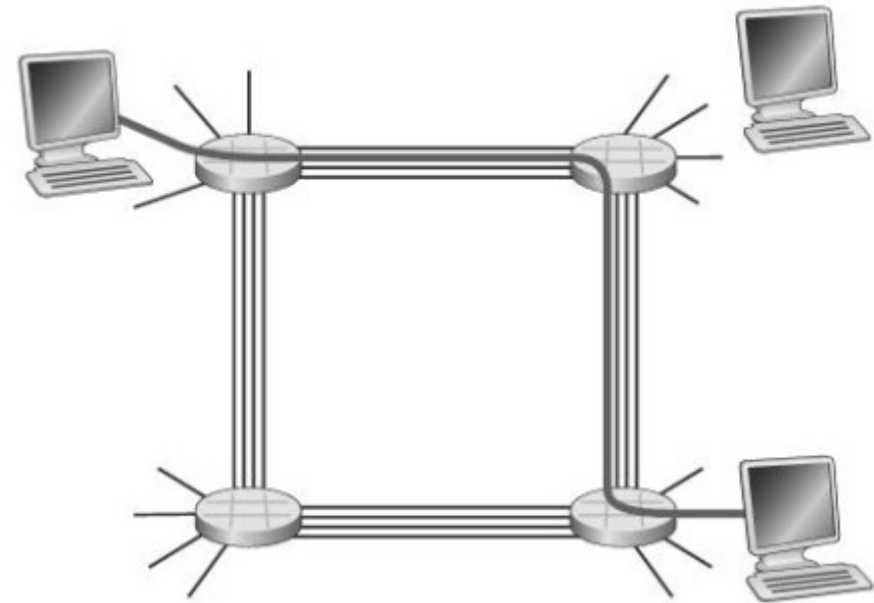
# La commutazione di circuito

- Nelle **reti a commutazione di circuito** le risorse richieste lungo un percorso (buffer e velocità di trasmissione sui collegamenti) per consentire la comunicazione tra sistemi periferici sono riservate per l'intera durata della sessione di comunicazione.
- I commutatori sul percorso tra mittente e destinatario mantengono lo stato della connessione per tutta la durata della comunicazione.
- Le reti telefoniche sono esempi di reti a commutazione di circuito.
- **Vantaggi:** il mittente può trasferire i dati a una velocità costante garantita (tutti i dati seguono lo stesso percorso).

# La commutazione di circuito (2)

- Esempio:

- Commutatori con 4 connessioni
- Connessione punto a punto
- Ogni collegamento ha un quarto della capacità trasmissiva totale del collegamento



# Multiplexing - commutazione di circuito

- Un circuito all'interno di un collegamento a commutazione di circuito è implementato con due tecniche principali:
  - Multiplexing a divisione di frequenza (FDM, *frequency-division multiplexing*)
  - Multiplexing a divisione di tempo (TDM, *time-division multiplexing*).

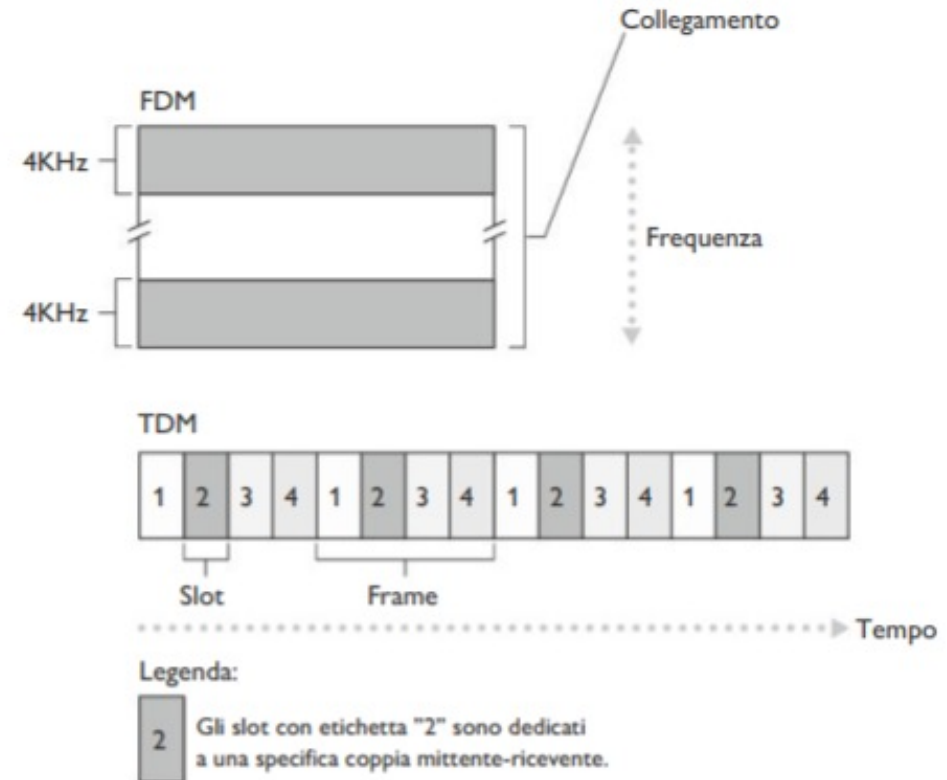


# Multiplexing - commutazione di circuito (2)

- **FDM:** lo spettro di frequenza di un collegamento viene suddiviso tra le connessioni stabilite tramite il collegamento. Nello specifico, il collegamento dedica una banda di frequenza a ciascuna connessione per la durata della connessione stessa.
- **TDM:** il tempo viene suddiviso in frame (intervalli) di durata fissa, a loro volta ripartiti in un numero fisso di slot (porzioni) temporali. Quando la rete stabilisce una connessione attraverso un collegamento, le dedica uno slot di tempo in ogni frame.

# Multiplexing - commutazione di circuito (3)

- In FDM ogni circuito occupa con continuità una frazione dell'ampiezza di banda.
- In TDM, ogni circuito occupa l'intera ampiezza di banda per brevi intervalli di tempo (detti slot).



FDM e TDM per uno specifico collegamento di rete che supporta fino a quattro circuiti.

# Commutazione di pacchetto

- Le applicazioni distribuite scambiano messaggi (es. testo, immagine, audio, video), che sono suddivisi in parti più piccole (**pacchetti**).
- I pacchetti viaggiano attraverso collegamenti e commutatori di pacchetto.
- I pacchetti vengono trasmessi su ciascun collegamento a una velocità pari alla velocità totale di trasmissione del collegamento stesso.

Tempo trasmissione del pacchetto

$$T = L/R$$

L: dimensione pacchetto (bit)

R: velocità di trasmissione del collegamento (bps, bit/s)

# Commutazione di pacchetto (2)

- Ogni pacchetto utilizza le risorse **su richiesta** e viene trattato dalla rete di commutazione indipendentemente dagli altri pacchetti che formano il messaggio originale.
- Questo significa che ogni pacchetto può seguire una **strada diversa** dagli altri per raggiungere il destinatario.
- Ogni pacchetto può quindi raggiungere la destinazione con un ritardo diverso dagli altri pacchetti, in un ordine non corretto, oppure andare perso.

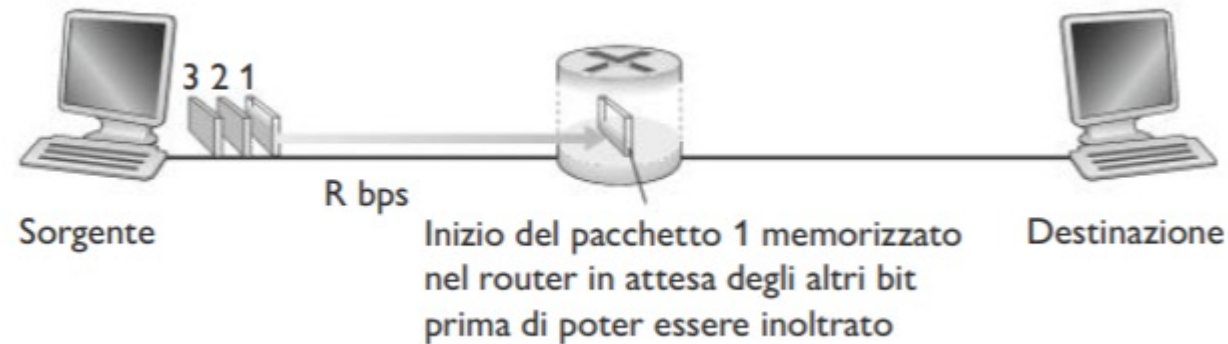
# Commutazione di pacchetto (3)

- **Vantaggi:**

- Migliore condivisione della larghezza di banda rispetto alla commutazione di circuito.
- Implementazione semplice, efficiente e meno costosa della commutazione di circuito.
- La commutazione di pacchetto e quella di circuito sono entrambe presenti negli odierni sistemi di telecomunicazioni, ma la tendenza è certamente in direzione della commutazione di pacchetto.

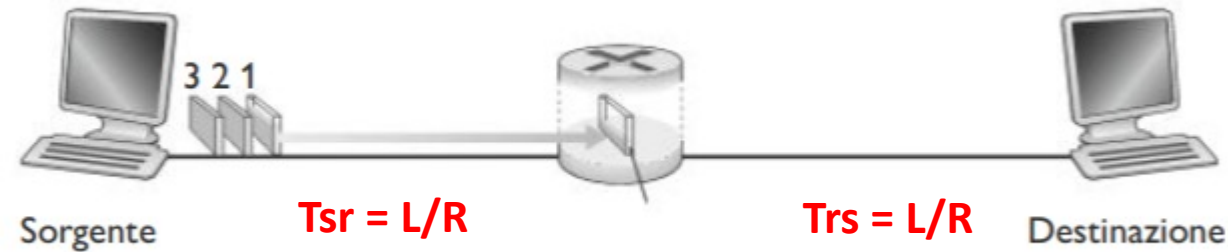
# Trasmissione store-and-forward

- Il commutatore deve ricevere l'intero pacchetto prima di poter cominciare a trasmettere sul collegamento in uscita il primo bit.



# Trasmissione store-and-forward

- Esempio:



Tempo trasmissione del pacchetto dalla sorgente alla destinazione

$$T = T_{sr} + T_{rs} = L/R + L/R = 2L/R$$

# Commutazione di pacchetto: Ritardi

- Ogni commutatore di pacchetto connette più collegamenti. Per ciascuno di questi, il commutatore mantiene un **buffer di output** per conservare i pacchetti che sta per inviare su quel collegamento.
- **Ritardo di accodamento:** un pacchetto in arrivo, che debba essere inviato attraverso un collegamento occupato dalla trasmissione di un altro, deve attendere nella coda di output.
- **Perdita di pacchetto:** se il buffer di output è pieno e arriva un nuovo pacchetto, questo o un pacchetto nel buffer viene eliminato.



# Commutazione di pacchetto: Ritardi (2)

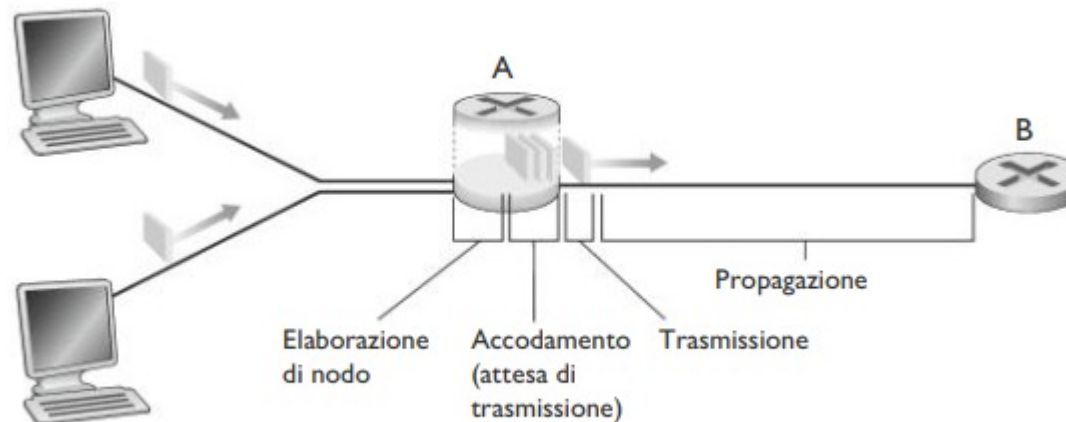
- **Ritardo di elaborazione:** tempo richiesto dal commutatore per esaminare l'intestazione del pacchetto e per determinare dove dirigerlo.
- **Ritardo di trasmissione:** tempo richiesto per trasmettere tutti i bit del pacchetto sul collegamento (*link*) ( $L/R$ ).
- **Ritardo di propagazione:** tempo di trasferimento del bit sulla linea di comunicazione ( $d/v$ ,  $d$  distanza fra router,  $v$  velocità di propagazione del segnale).
  - **Mezzi non vincolati (etere)**  $v = c \cong 3 \times 10^8 \text{ m/s}$ . ← velocità propagazione della luce nel vuoto
  - **Mezzi vincolati**  $v = 2 \times 10^8 \text{ m/s}$ .

# Commutazione di pacchetto: Ritardi (3)

- Il **ritardo di trasmissione** è la quantità di tempo impiegata dal router per trasmettere in uscita il pacchetto, ed è funzione della lunghezza del pacchetto e della velocità di trasmissione del collegamento, ma non ha niente a che fare con la distanza tra i due router.
- Il **ritardo di propagazione**, invece, è il tempo richiesto per la propagazione di un bit da un router a quello successivo, ed è funzione della distanza tra i due router, ma non ha niente a che fare con la lunghezza del pacchetto o con la velocità di trasmissione propria del collegamento.

# Commutazione di pacchetto: Ritardi (4)

- I ritardi possono essere in genere dell'ordine dei microsecondi o dei millisecondi.



dipende dal traffico di rete

dipende dal mezzo fisico

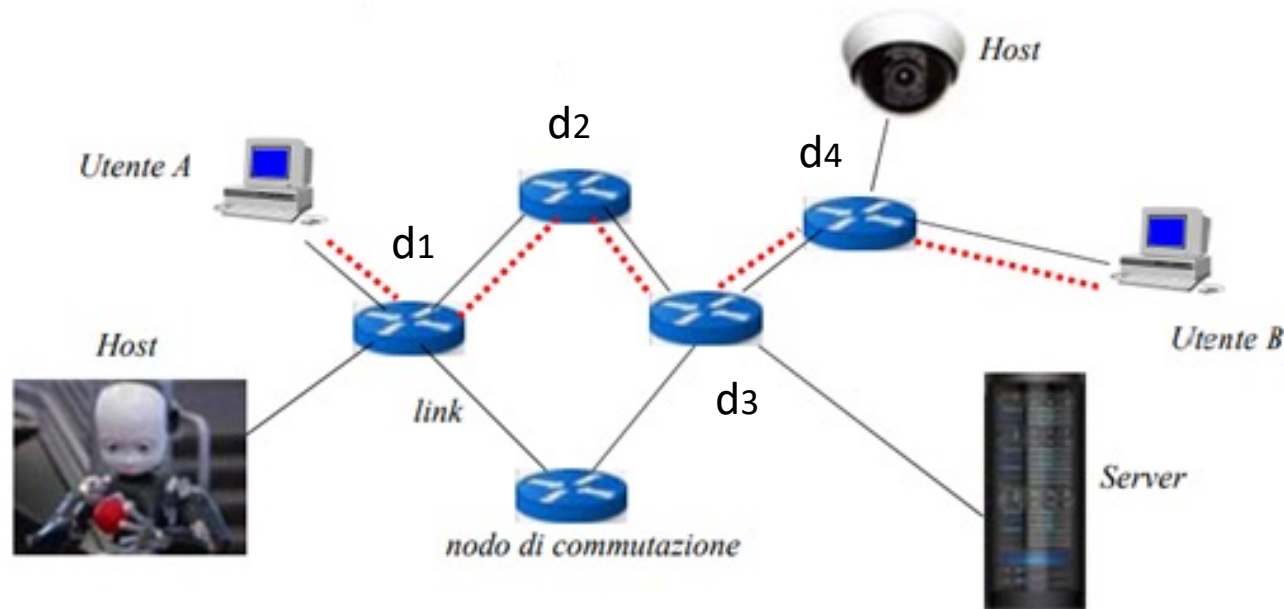
$$d_{\text{nodo}} = d_{\text{elab}} + d_{\text{acc}} + d_{\text{trasm}} + d_{\text{prop}}$$

dipende dalla capacità del processore del router

dipende dalla velocità del link

# Commutazione di pacchetto: Ritardi (5)

- **Ritardo end-to-end:** ritardo totale dalla sorgente alla destinazione

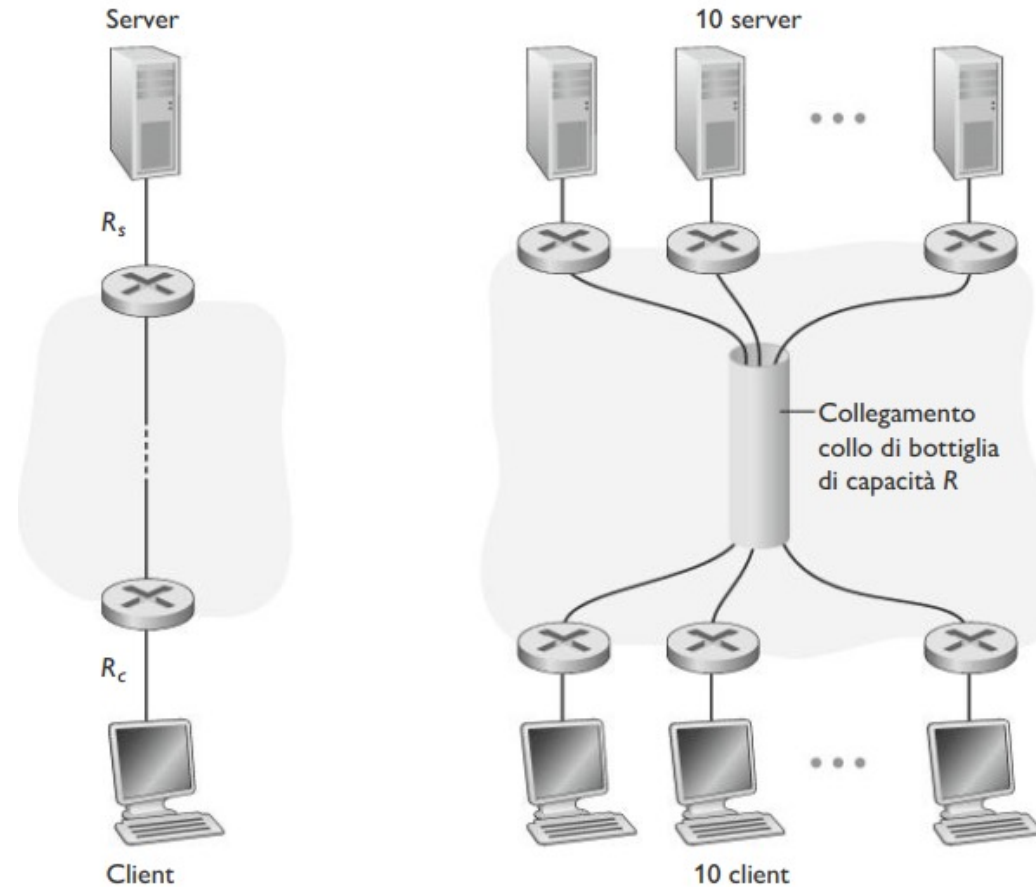


$$d_{A-B} = d_1 + d_2 + d_3 + d_4$$

# Throughput

- Il **throughput istantaneo** è la velocità (in bps) alla quale un host sta ricevendo una risorsa.
- Per alcune applicazioni, come la telefonia su Internet, è auspicabile avere un ritardo basso e un throughput istantaneo sopra una certa soglia in modo continuativo (per esempio sopra i 24 kbps per alcune applicazioni di telefonia e oltre i 256 kbps per alcune applicazioni video in tempo reale).

# Throughput (2)



Il throughput dipende non solo dalla velocità di trasmissione dei collegamenti lungo il percorso, ma anche dal traffico sulla rete.

# Progettazione di Internet

- I problemi da affrontare:
  - **Affidabilità della trasmissione**
    - Guasti hardware
    - Danneggiamento dei dati (es. rumore sui link)
    - Perdite di dati
    - Duplicazioni pacchetti
    - Arrivi ordine pacchetti invertiti
  - **Gestione della ritrasmissione** (es. nel caso di perdita dei dati)
  - **Gestione del flusso** (non inviare dati con un tasso che non può essere gestito dal destinatario!)
  - **Instradamento**, scelta del cammino migliore del pacchetto (conoscenza della topologia della rete e del traffico)

# Progettazione di Internet (2)

- I problemi da affrontare:
  - **Congestione** della rete (es. come evitare zone congestionate?)
  - Gestione **reti eterogenee**
  - **Frammentazione** e riassemblaggio dei messaggi
  - Apertura e chiusura delle **connessioni**, gestione di connessioni multiple
  - **Qualità del servizio** (es. ritardo, larghezza di banda, ecc.)
  - Organizzazione del dialogo e della sincronizzazione tra programmi applicativi
  - Gestione della sicurezza

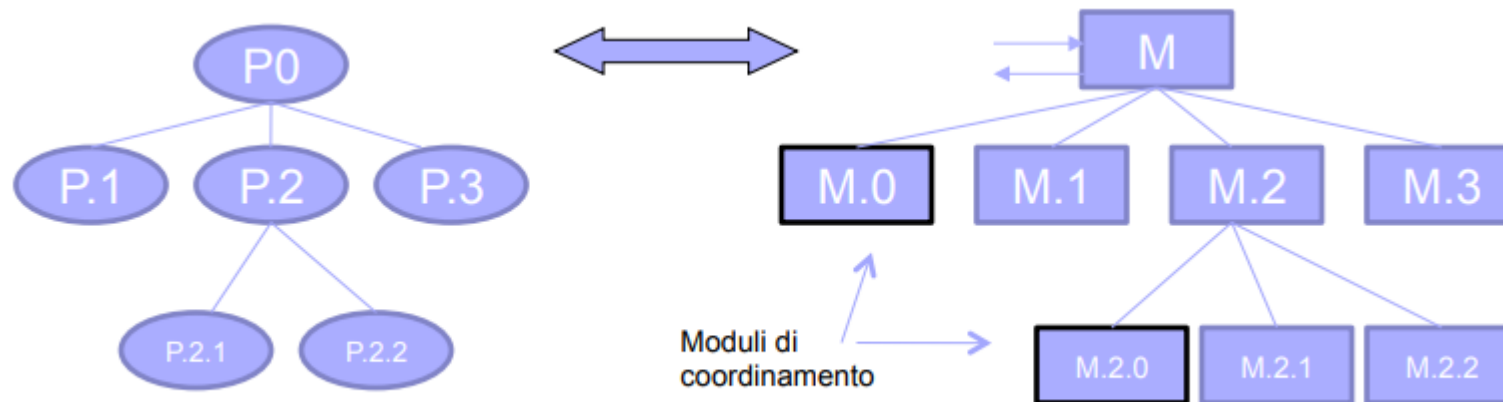


# Decomposizione modulare

- Come fare per risolvere i problemi principali di una progettazione della rete?
- Il problema della comunicazione è un problema complesso
- Va affrontato mediante decomposizione del problema e modularizzazione delle funzioni

# Decomposizione modulare (2)

- Decomposizione del problema iniziale (P0) - la comunicazione in rete - in sottoproblemi (si ottiene un albero di decomposizione).
- La decomposizione del problema iniziale in sottoproblemi induce un'analoga decomposizione del sistema software iniziale M (detto modulo iniziale) in sottomoduli (**processo di modularizzazione**).



# Architettura a livelli

- Un'**architettura a livelli** consente di discutere una parte specifica e ben definita di un sistema articolato e complesso.
- Questa stessa semplificazione ha un valore considerevole grazie all'introduzione della modularità, che rende molto più facile cambiare l'implementazione del servizio fornito da un determinato livello.
- La stratificazione dei protocolli presenta vantaggi concettuali e strutturali.

# Pila dei protocolli

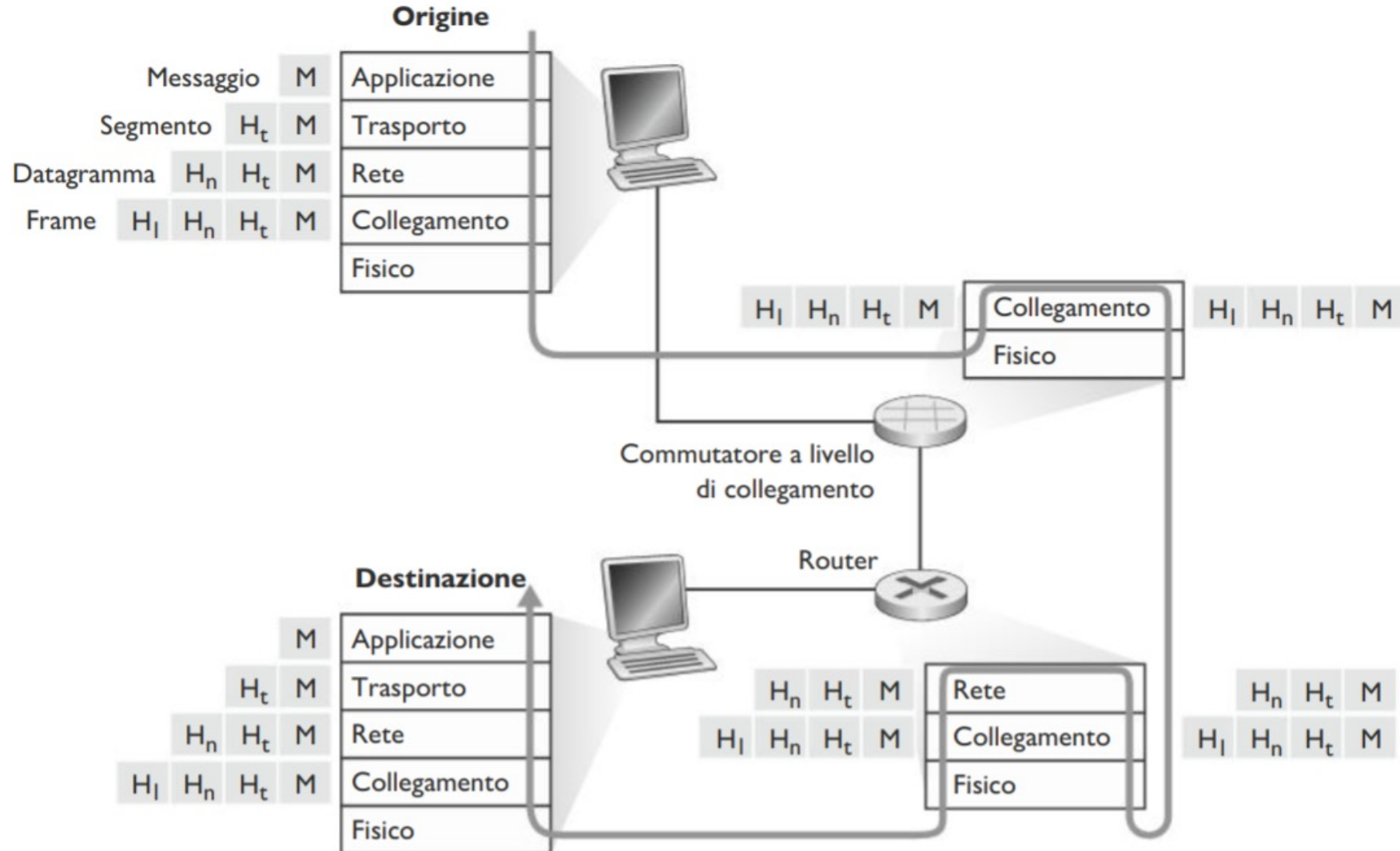
- La **pila di protocolli** (*protocol stack*) di Internet consiste di cinque livelli: fisico, collegamento, rete, trasporto e applicazione.



# Pila dei protocolli (2)

- Il **livello di applicazione** (*application layer*) è la sede delle applicazioni di rete e dei relativi protocolli.
- Il **livello di trasporto** (*transport layer*) trasferisce i messaggi (segmenti) del livello di applicazione tra punti periferici gestiti dalle applicazioni.
- Il **livello di rete** (*network layer*) si occupa di trasferire i pacchetti (datagramma) a livello di rete da un host a un altro.
- Il **livello di collegamento dati** (*data link layer*) si occupa del trasferimento dei pacchetti (frame) da un nodo (host o router) a quello successivo sul percorso del pacchetto.
- Il **livello fisico** (*physical layer*) trasferisce i singoli bit del frame da un nodo a quello successivo.

# Pila dei protocolli (3)



# Pila dei protocolli (4)

- **Incapsulamento:** ad ogni informazione ai vari livelli il protocollo concatena informazioni aggiuntive, le cosiddette informazioni di intestazione a livello di trasporto, ***header*** (H).
- Aa ciascun livello, il pacchetto ha due tipi di campi:
  - Intestazione
  - Payload (il carico utile trasportato)
- Il payload è tipicamente un pacchetto proveniente dal livello superiore.

# Tipi di architetture

- **Proprietaria:** è una architettura basata su scelte indipendenti e arbitrarie del costruttore ed è generalmente incompatibile con architetture diverse. Il produttore non rende pubbliche le specifiche per cui nessun altro può produrre apparati compatibili. Esempio: vecchie reti IBM SNA, Digital DECNET Phase IV, Appletalk.
- **Standard de facto:** è una architettura basata su specifiche di pubblico dominio diventata di larghissima diffusione. Esempio: **Internet Protocol Suite** (o TCP/IP)
- **Standard de iure:** è un'architettura basata su specifiche pubbliche approvate da organismi di standardizzazione internazionali. Esempio: IEEE 802 per LAN.



# Modello ISO/OSI

- Il **modello OSI** (*Open Systems Interconnection*) è uno **standard de iure** stabilito nel 1984 dall'*International Organization for Standardization* (**ISO**),
- Il modello prevede un'architettura basata su sette livelli:
  - **Livello di applicazione**
  - **Livello di presentazione** (gestione della sintassi e della semantica delle informazioni trasmesse)
  - **Livello di sessione** (coordina il dialogo tra utenti)
  - **Livello di trasporto**
  - **Livello di rete**
  - **Livello di collegamento dati**
  - **Livello fisico**

# Modello TCP/IP

- Il **Modello TCP/IP** è uno **standard de facto** basato su quattro livelli:
  - Livello di applicazione
  - Livello di trasporto
  - Livello di rete
  - Livello di accesso alla rete
- Essendo importante la distinzione fra livello di collegamento e livello fisico, è possibile utilizzare un modello ibrido che distingue i due strati e porta il numero di livelli a cinque.