

PARTE 2

SUITE STANDARD

Modulo 1

Stack di protocolli standard

Comunicazione e standard

- **La comunicazione tra nodi differenti e, possibilmente, basati su piattaforme hardware e/o software eterogenee necessita di STANDARD**
- **L'informatica, da sempre, conosce due modi per arrivare ad uno standard**
 - STANDARD de iure
 - STANDARD de facto
- **Gli standard di comunicazioni tra calcolatori offrono un esempio “storico”**

Due standard in concorrenza



ISO/OSI

(de iure)



TCP/IP

(de facto)



Modulo 1a: Stack ISO/OSI

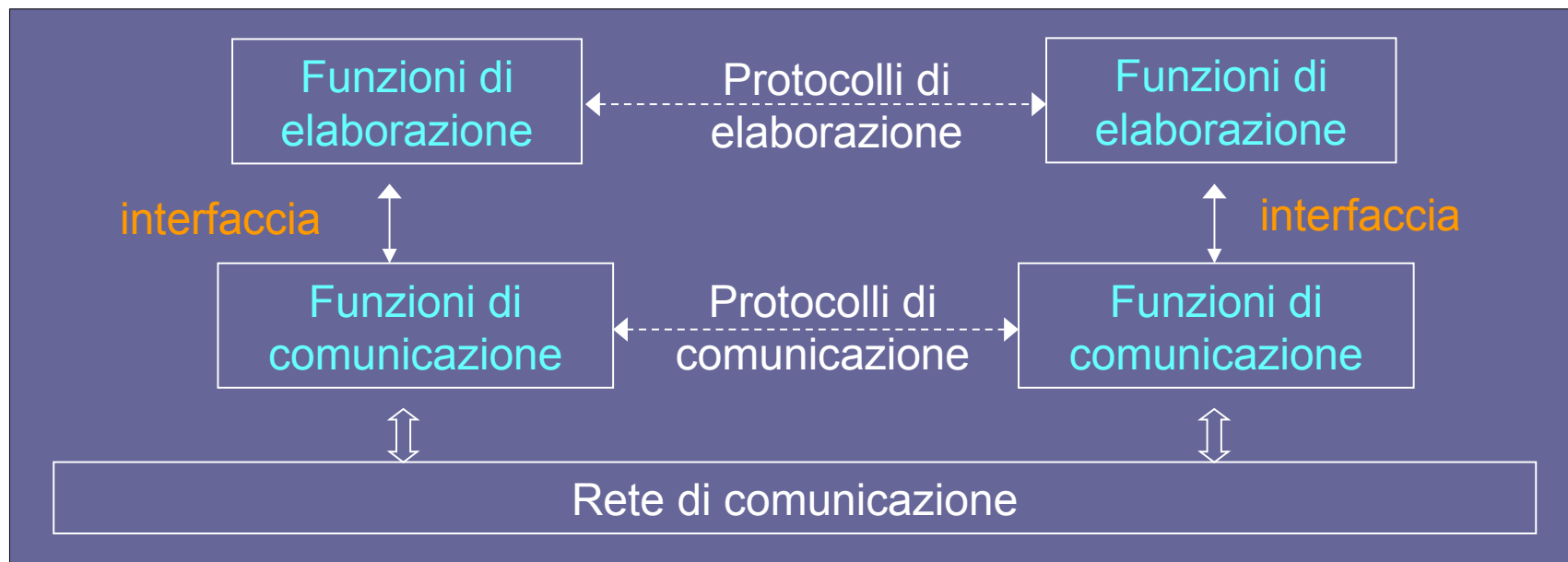
Il caso dello “standard de iure” ISO/OSI

- L'organizzazione **ISO** (International Standard Organization) ha definito le specifiche di quello che sarebbe dovuto diventare lo standard di protocolli per l'interconnessione di nodi eterogenei:
- **OSI** (Open System Interconnection)

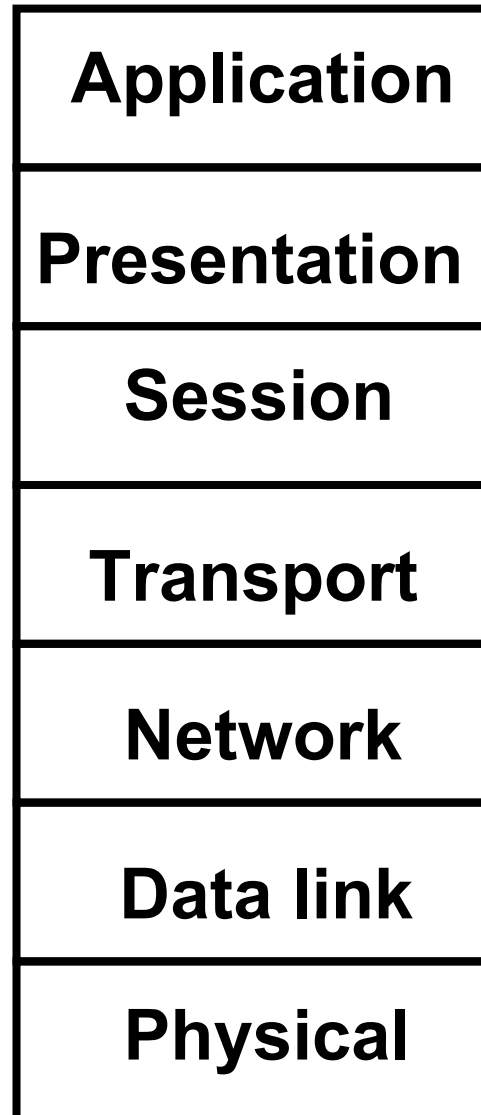
Funzionalità del modello ISO/OSI

1) Protocolli di comunicazione (network level): riguardano la comunicazione di messaggi tra nodi della rete, in modo da nascondere le caratteristiche dei mezzi fisici di trasmissione alle funzionalità di elaborazione

2) Protocolli di elaborazione (application level): insieme di meccanismi per il controllo delle applicazioni

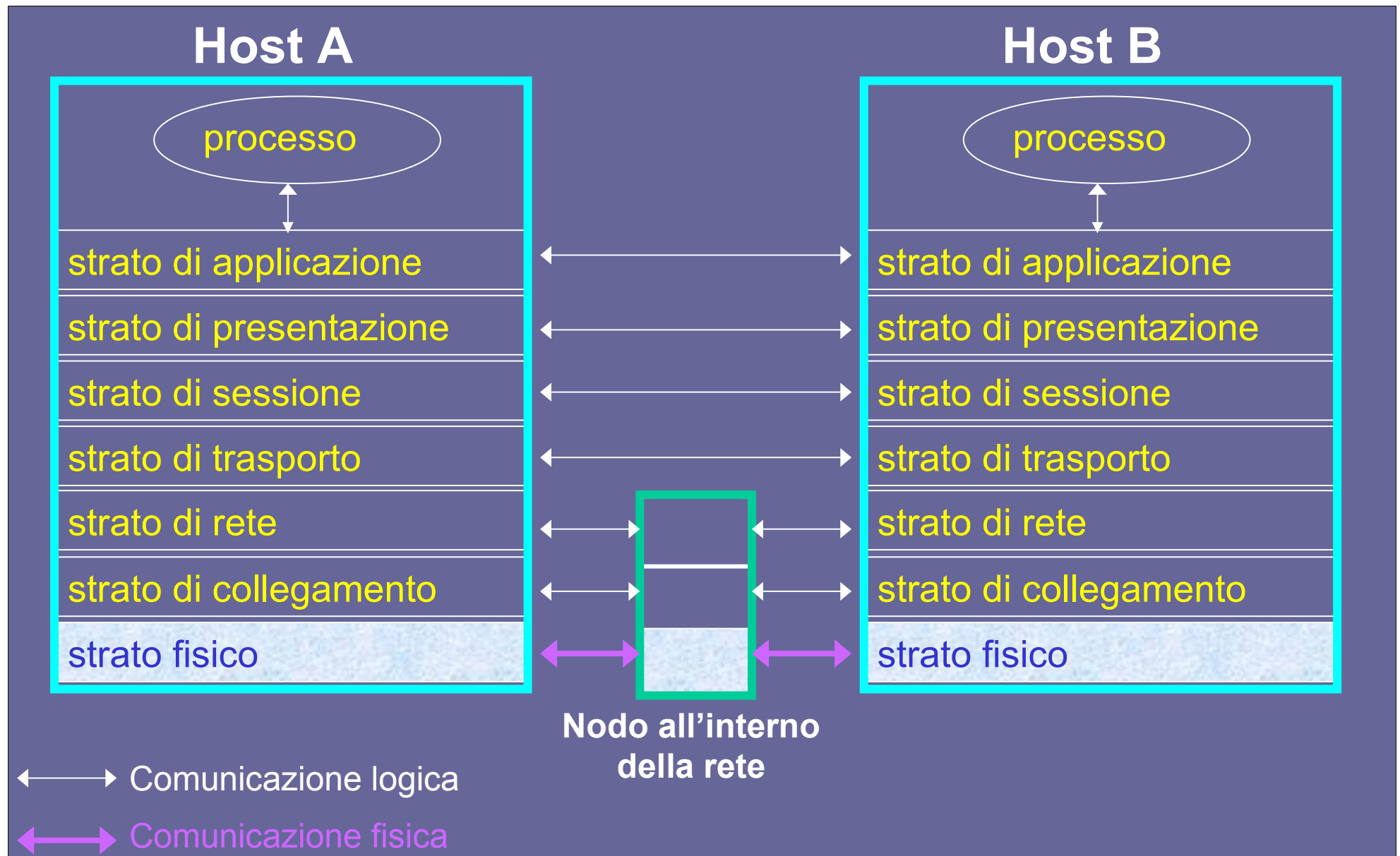


I 7 livelli dello stack ISO/OSI



7 livelli

I 7 livelli dello stack ISO/OSI



Livelli ISO/OSI

- **Livello fisico (1)**: Gestisce i particolari meccanici ed elettrici della trasmissione fisica di un flusso di bit



- **Livello di collegamento dati (2)**: Gestisce i *frame* o i *pacchetti* trasformando la semplice trasmissione in una linea di comunicazione priva di errori non rilevati.
 - Gestisce l'accesso e l'uso dei canali fisici, gestisce il formato dei messaggi suddividendo (ove necessario) i dati in frame.
 - Gestisce la corretta sequenza dei dati trasmessi, comprendente l'uso di codifiche ridondanti (ad es., bit di parità) per l'individuazione e la correzione di errori che si sono verificati nello strato fisico, e la conferma dell'avvenuta ricezione

- **Livello di rete (3)**: Fornisce i collegamenti e l'instradamento dei pacchetti nella rete, comprese la gestione dell'indirizzo dei pacchetti in uscita, la decodifica dell'indirizzo dei pacchetti in ingresso e la gestione delle informazioni di instradamento (ad es., router)

Livelli ISO/OSI (cont.)

- **Livello di trasporto (4)**: Effettua il controllo end-to-end della sessione di comunicazione (accesso alla rete da parte del client e trasferimento dei messaggi tra i client) e garantisce l'affidabilità del trasporto
- **Livello di sessione (5)**: Consente a utenti su macchine eterogenee di stabilire sessioni, implementando funzioni di coordinamento, sincronizzazione e mantenimento dello stato (di sessione)

Livelli ISO/OSI (cont.)

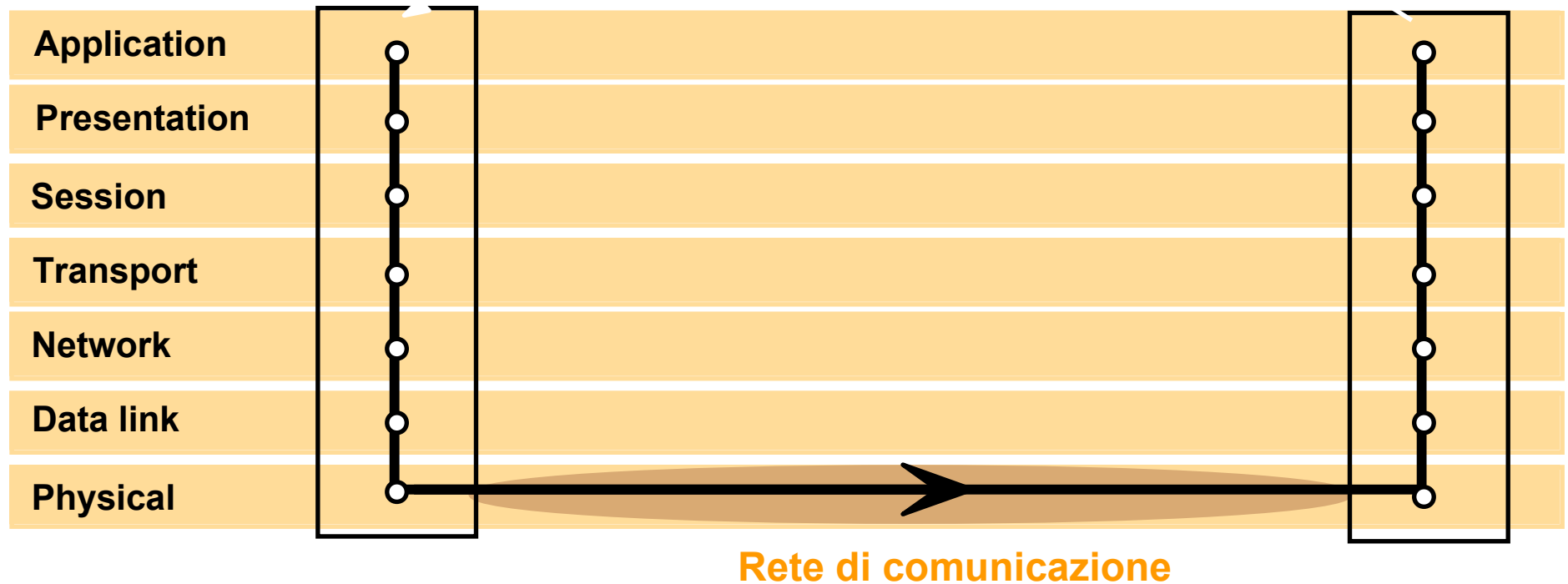
- **Livello di presentazione (6)**: Risolve le differenze di formato che possono presentarsi tra diversi nodi della rete (ad es., conversione tra caratteri ASCII, Unicode, EBCDIC, conversione di codifica tra little- e big-endian), ma gestisce anche la compressione dei dati, la sicurezza e l'autenticità dei messaggi attraverso tecniche di crittografia
- **Livello di applicazione (7)**: Fornisce un'interfaccia standard per i programmi applicativi che utilizzano la rete, mascherando le peculiarità e la complessità del sistema sottostante

Formato del messaggio inviato

- Messaggio (PDU) composto da intestazione (header) e dati
- Ogni livello aggiunge una propria intestazione



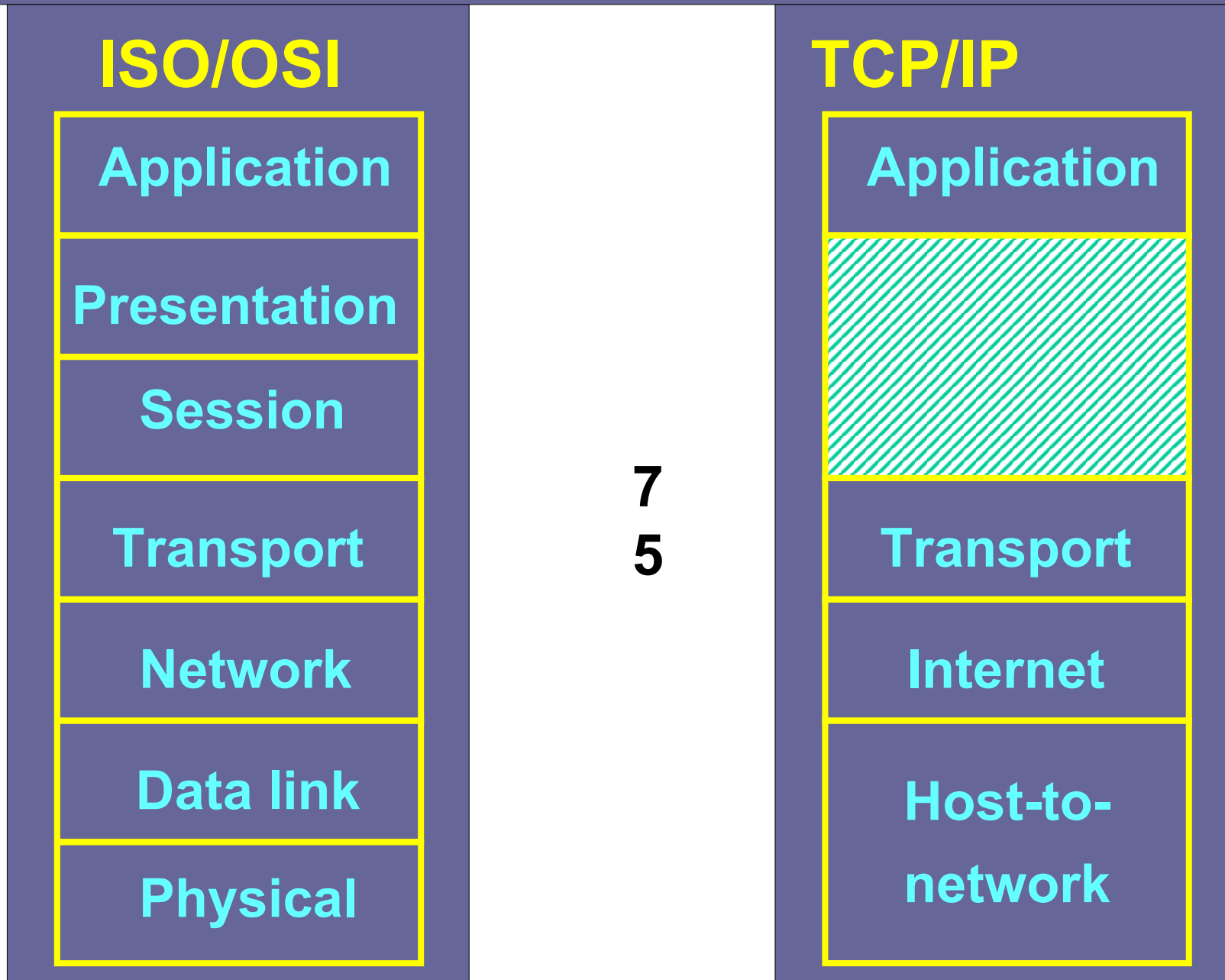
Comunicazione nel Modello ISO/OSI



Ma l'ISO/OSI non è riuscito ad affermarsi perché nel frattempo stava esplodendo

Modulo 1b: Stack TCP/IP ("Protocolli di Internet")

I livelli dei due Protocol Stack



Al modello ISO/OSI

- **Cattiva tempistica**
- **Cattiva tecnologia**
 - Influenzato dal modello IBM-SNA
 - Ridondanze
- **Cattiva implementazione**
 - Complessità
 - Eccessivi 7 livelli per la tecnologia (reti-computer) del tempo
- **Pessima politica**
 - Modello imposto contro il libero TCP/IP legato a Unix

Al modello TCP/IP

- **Poco generale**
- **Meno concettuale e più orientato al funzionamento**
- **Livelli host-to-network confusi e interdipendenti**
- **Protocolli sviluppati ad hoc invece che protocolli generali**

Altro motivo del successo di TCP/IP

- **Disponibilità di una buona implementazione dello stack in versione open source a metà degli anni '80 su BSD Unix**
- **Disponibilità di un buon insieme di API (BSD socket API) per sviluppare applicazioni di rete: non perfette, ma funzionanti**
- **Al contrario,**
 - il comitato ISO/OSI definì le specifiche dello stack
 - le implementazioni funzionanti delle specifiche ISO/OSI erano molto in ritardo rispetto a quelle già disponibili TCP/IP

Due standard non più in concorrenza



ISO/OSI

(de iure)

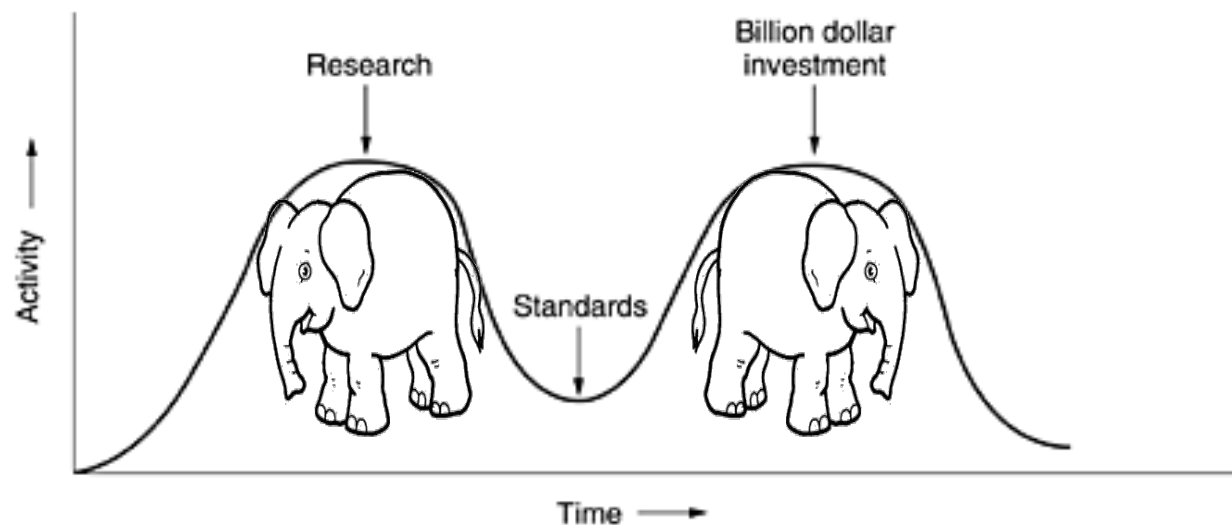


TCP/IP

(de facto)

Altra spiegazione: apocalisse dei due elefanti

- **Fase 1: Ricerca**
 - Non si possono rilasciare gli standard perché la teoria non è matura
- **Fase 2: Investimenti**
 - Non si possono rilasciare gli standard perché ci sono già implementazioni e investimenti
- **L'unico momento per gli standard è tra i due elefanti.**
 - Si rischia di essere schiacciati → Vedi ISO/OSI



Internet: Cosa non è ...

- **Non è una singola rete, ma un insieme di reti esteso in tutto il mondo**
- **Non è governata da un gruppo né da un ente né da un'unica azienda**
- **Non è gestita in modo centralizzato perché tutte le singole sottoreti che compongono Internet hanno una gestione autonoma**

**E soprattutto:
Non è sinonimo di *World Wide Web***

Internet: storia e leggenda

- **La leggenda:**

Un progetto finanziato dal Ministero della Difesa USA con lo scopo di realizzare una rete in grado di comunicare anche in seguito ad attacchi nucleari

- **La realtà:**

- Finanziata dal Ministero della Difesa USA
- Motivazione: successi spaziali dell'URSS
- Obiettivo: consentire l'accesso alle poche risorse di calcolo potenti (e costose) da vari centri di ricerca e Università USA

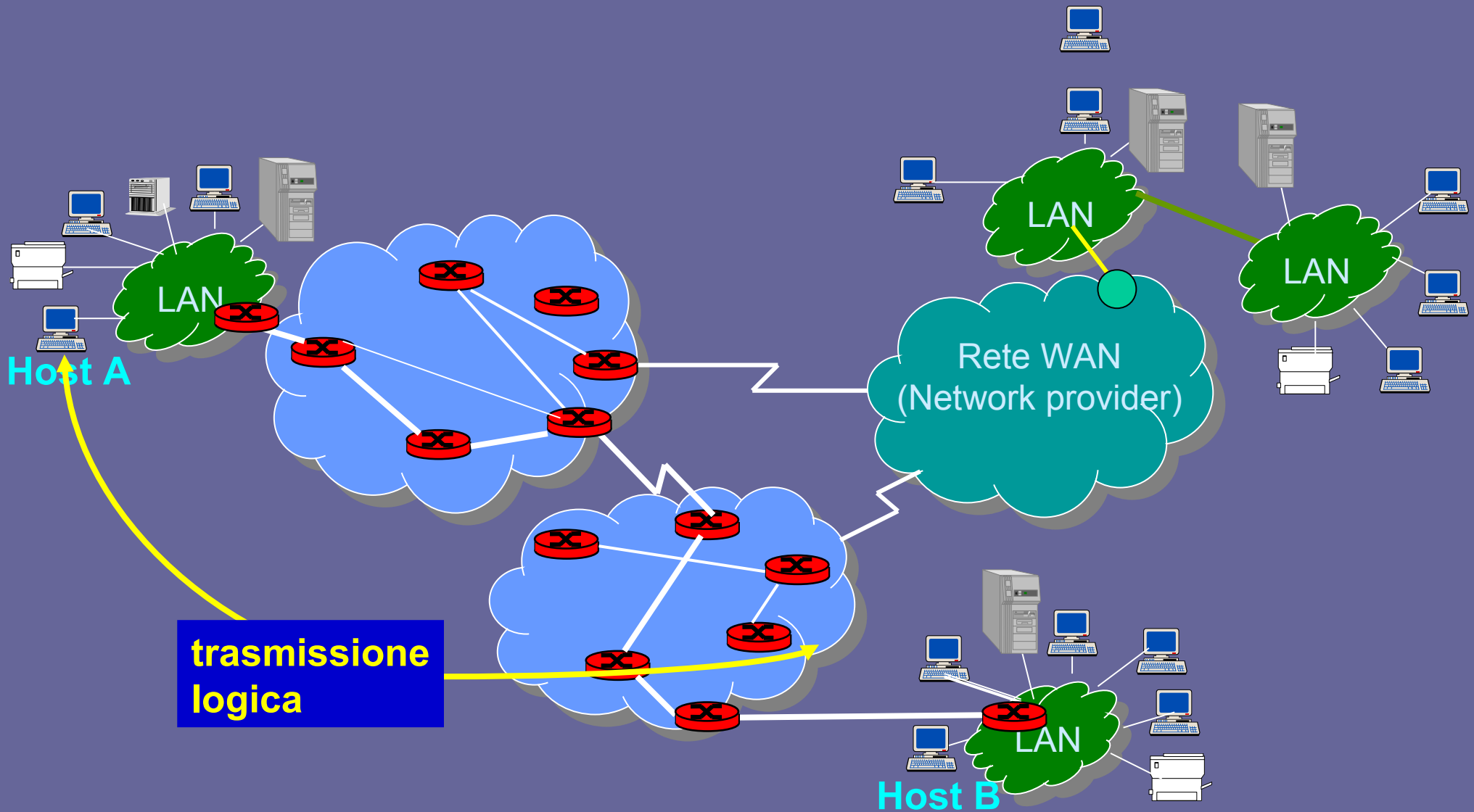
Modulo 2: Internet: funzionamento

Obiettivi progettuali di Internet

- **Architettura**
 - Connettere diversi host e diverse reti
- **Tecniche di trasmissione**
 - Store-and-forward
 - Packet switching

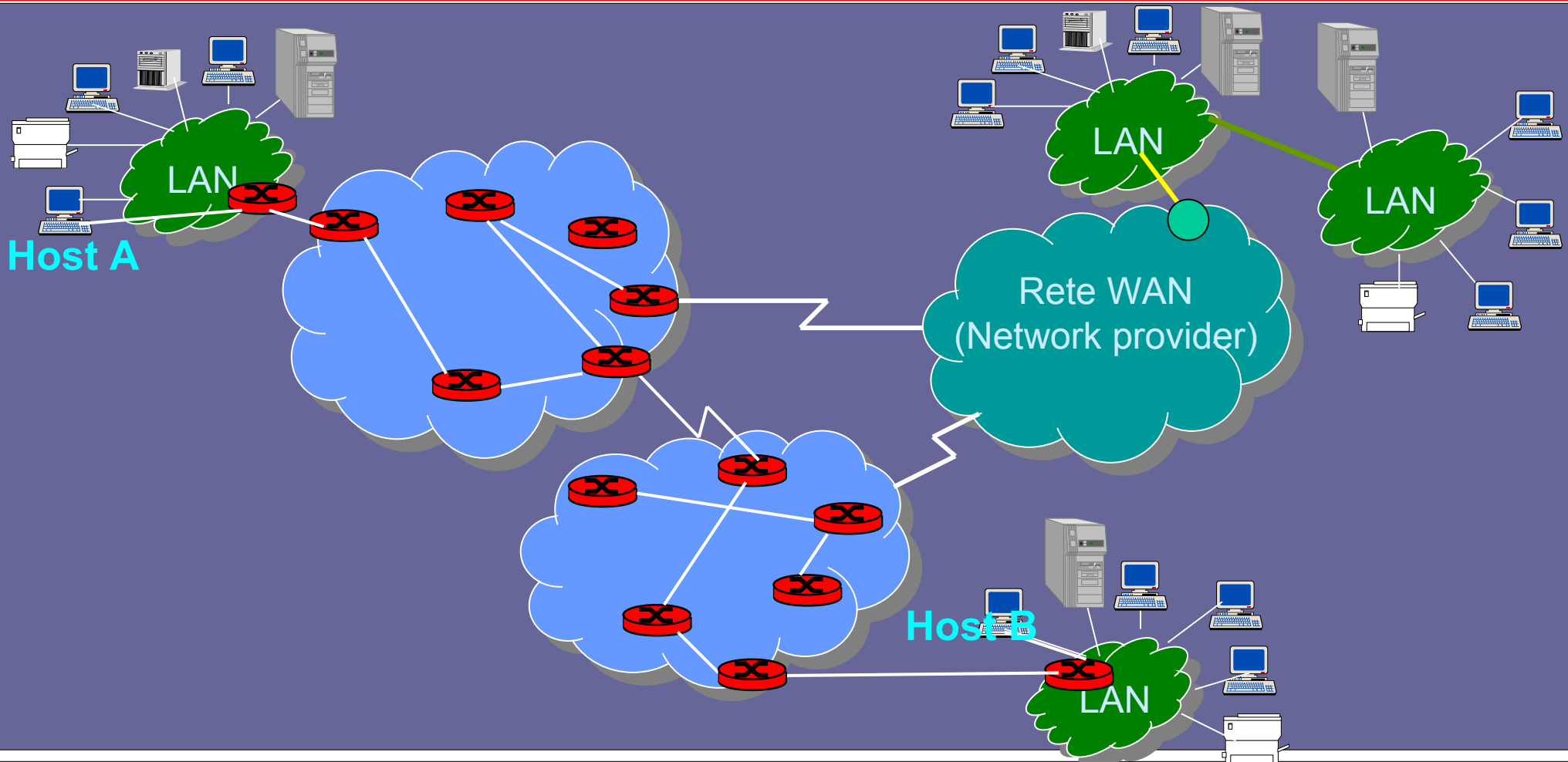
Comunicazione in Internet [vista 1]

Logicamente comunicano i due host terminali



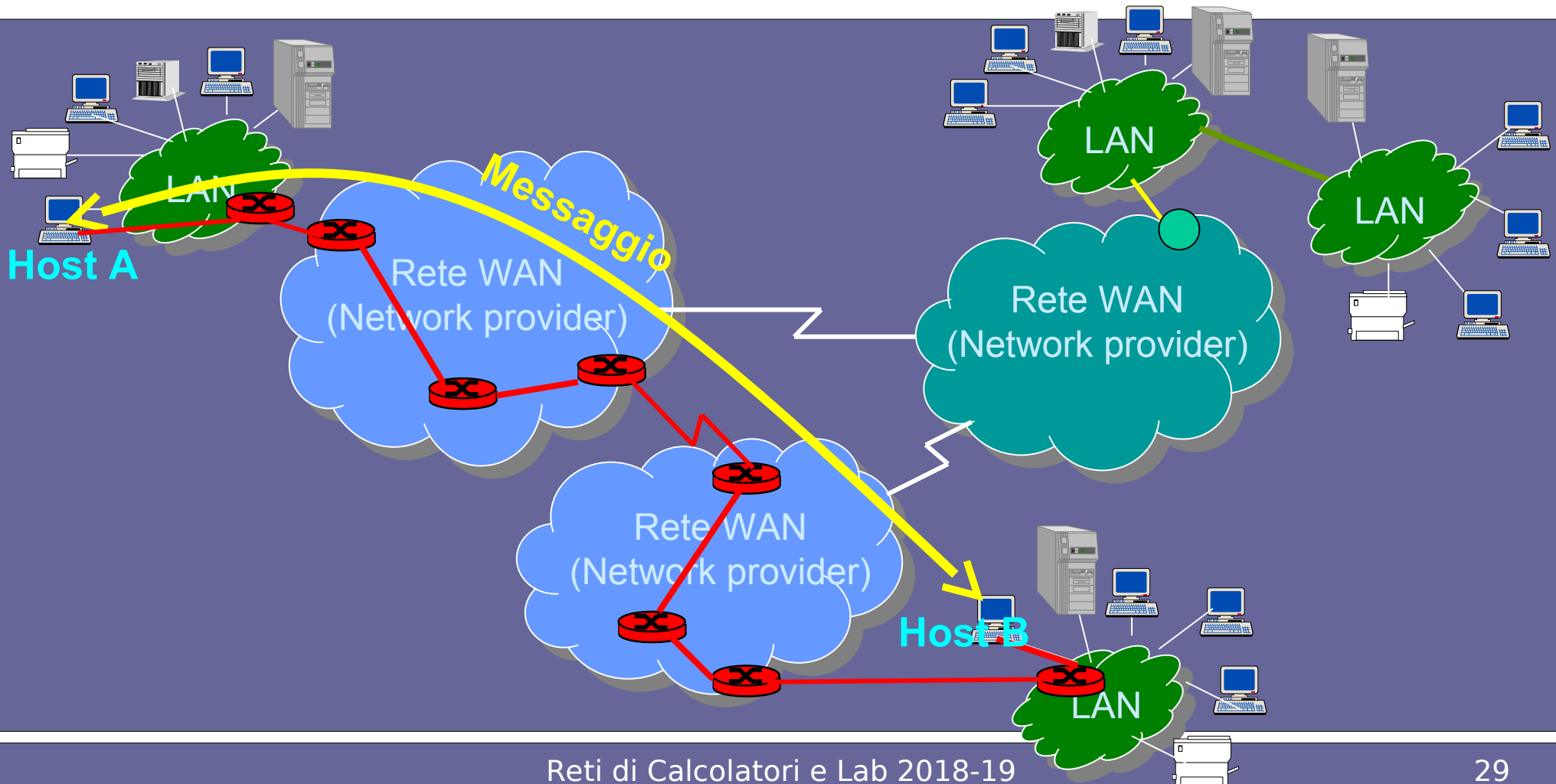
Architettura Internet

In realtà, Internet consiste in milioni di **host** (computer, PDA, TV,...), di dispositivi che instradano i messaggi (**router**) e di **link** di comunicazione (cavi, fibra ottica, satellitari,...)



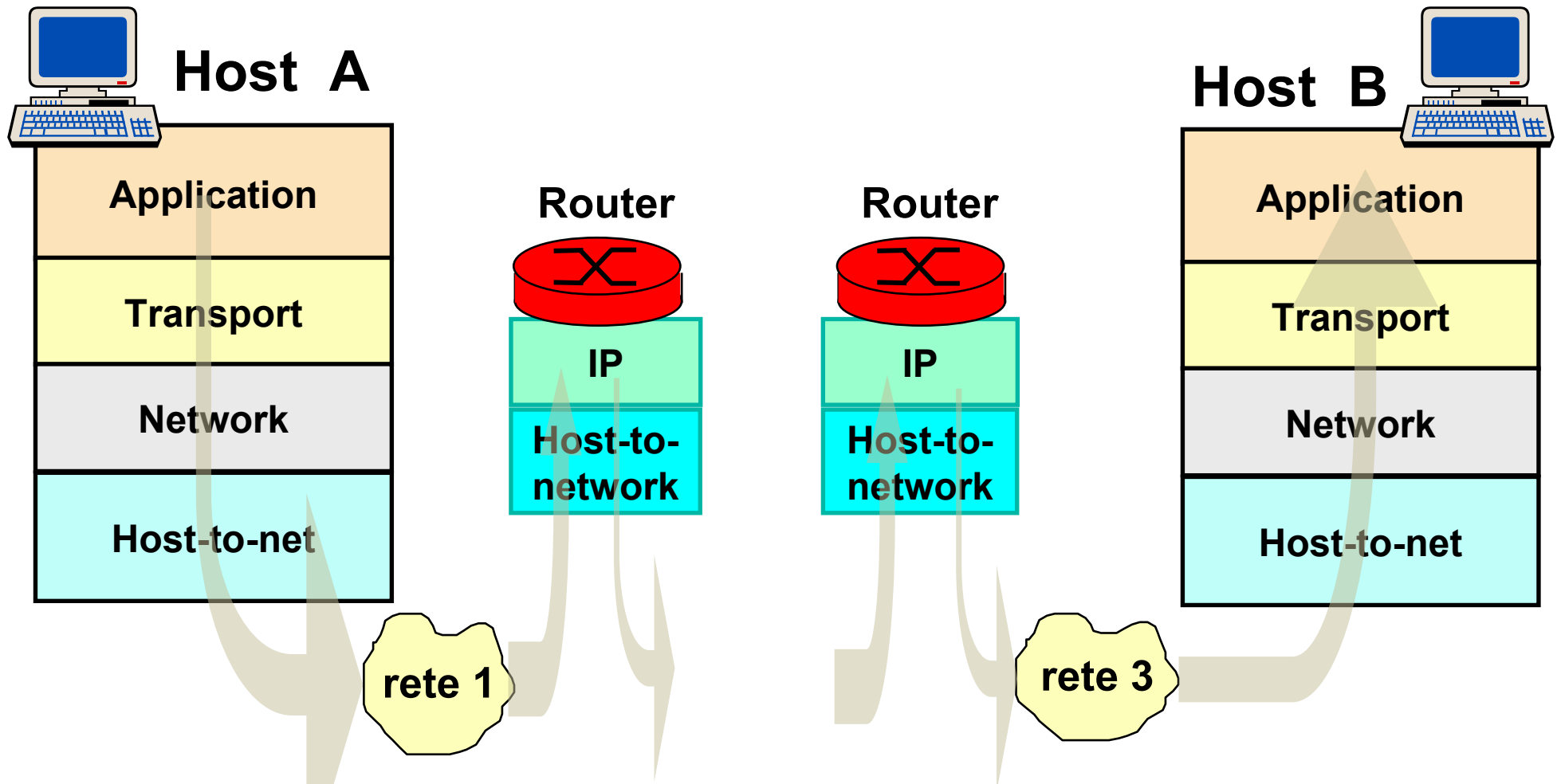
Comunicazione in Internet [vista 2]

Quindi, in realtà il messaggio deve attraversare vari *nodi intermedi (router)* con un meccanismo di *store and forward*



Comunicazioni in Internet [vista 3]

In ciascun nodo, l'informazione attraversa tutti i livelli necessari (5 per host, 3 per router)



Modulo 3: Packet switching vs. Circuit switching

La prima idea rivoluzionaria: Packet switching

- **1961: Kleinrock** mediante la teoria delle reti di code dimostra l'efficacia delle comunicazioni **packet-switching**
- **Per tutti gli anni '60 (e molti anche in seguito...), gli "esperti" di telecomunicazioni, sostenitori delle comunicazioni circuit-switching, sentenziavano "It will never work"**

[da "La storia di Internet scritta da coloro che l'hanno creata", 1997]

Due modalità per trasferire dati

- **Circuit switching**

- Un circuito virtuale dedicato per ogni comunicazione
- → L'idea alla base del sistema telefonico

- **Packet switching**

- I dati sono suddivisi in “parti” ed inviati attraverso la rete
- → L'idea alla base di Internet

Circuit switching

- **Necessità di riservare tutte le risorse (link e switch) end-to-end prima di trasmettere**
- **Risorse dedicate**
 - Non c'è possibilità di condivisione
 - Necessaria una fase di setup per ogni chiamata
 - Prestazioni garantite rispetto alla tipologia di risorse riservate

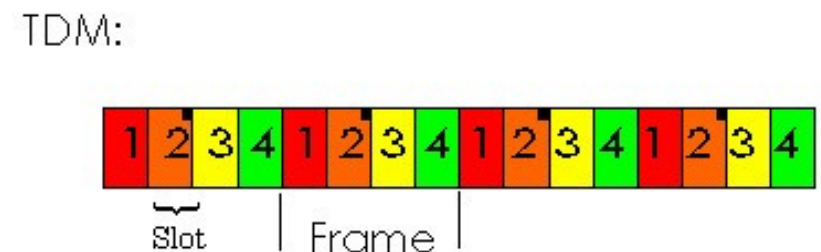
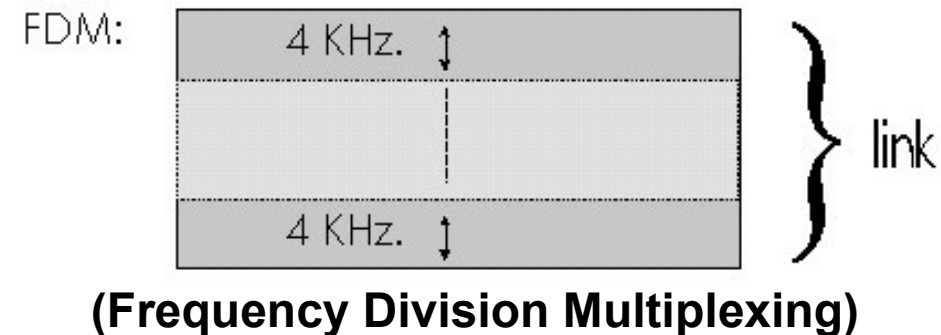
Circuit switching (cont.)

- **In realtà, anche nel circuit switching le risorse di rete (per es., la banda) non sono completamente dedicate, ma suddivise in “parti”**
- **Le parti sono assegnate alle chiamate**
- **Le parti di risorse, riservate per una chiamata, non sono utilizzabili da altre anche se non sono utilizzabili dalla chiamata che le possiede (non c'è possibilità di condivisione)**

Circuit switching (cont.)

Vi sono due metodi per suddividere la risorsa (link):

- Metodi basati sulla frequenza (FDM)
- Metodi basati sul tempo (TDM)

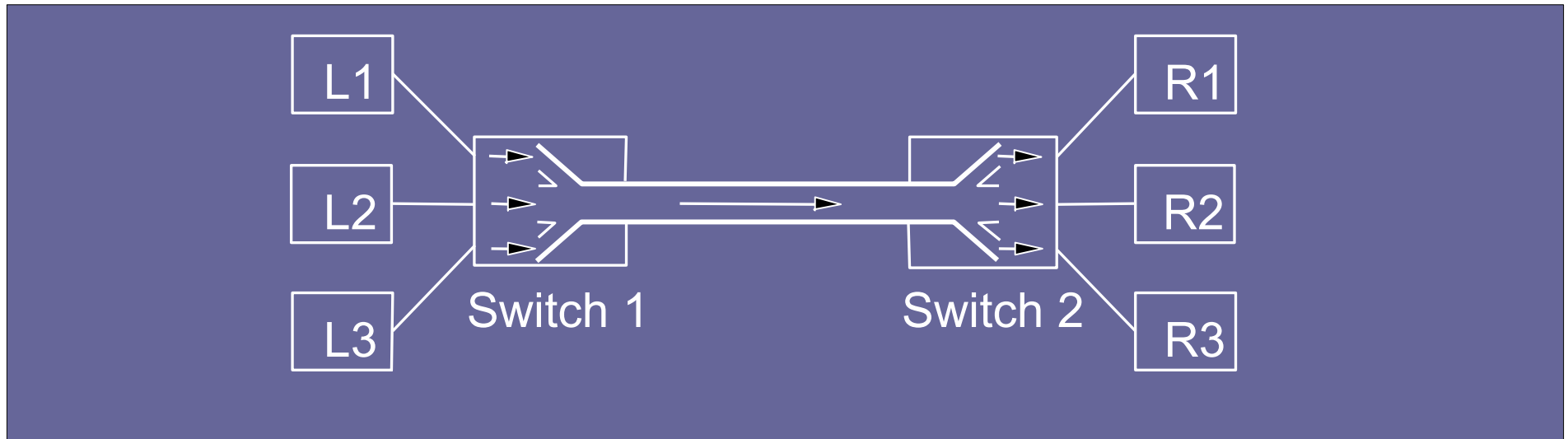


(Time Division Multiplexing)

... e per ogni canale, il server parla...

Multiplexing

- **Multiplexing (*condivisione*) di risorse**

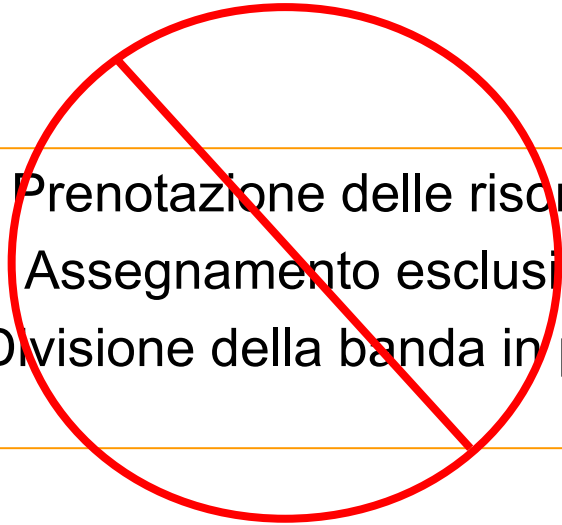


- **Deterministico (*del circuit switching*):**
- **Time-Division Multiplexing (TDM)**
- **Frequency-Division Multiplexing (FDM)**

Packet switching

Ogni comunicazione è suddivisa in pacchetti

- **I pacchetti condividono le risorse della rete**
- **Ogni pacchetto utilizza tutta la capacità trasmissiva di un link**
- **Le risorse sono utilizzate sulla base della necessità e non della prenotazione**



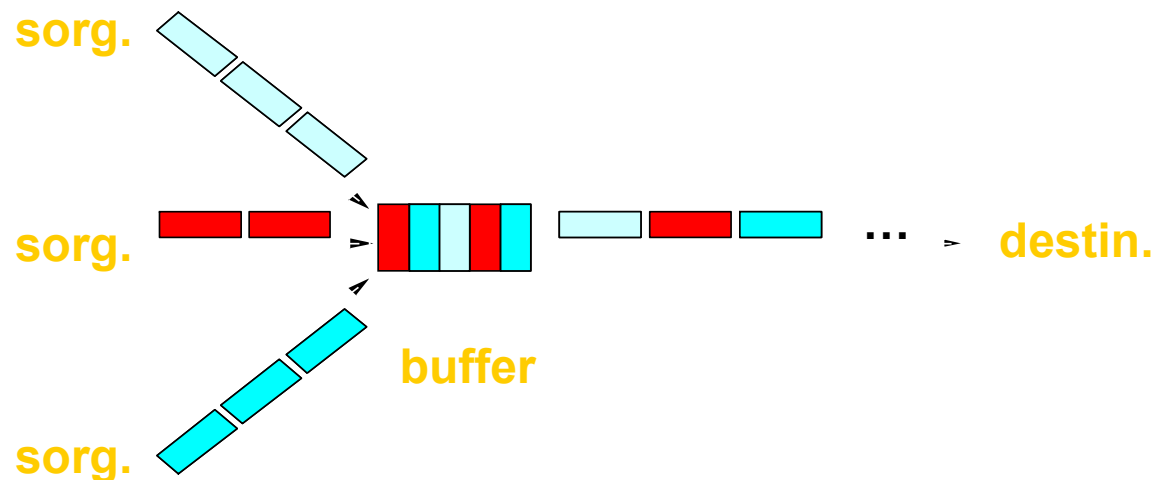
Prenotazione delle risorse
Assegnamento esclusivo
Divisione della banda in parti

Multiplexing statistico del packet switching

- **Si può dire che il packet switching segua un principio di multiplexing statistico a suddivisione di tempo, ma su richiesta invece che ad intervalli prefissati (come nel caso del TDM del circuit switching)**
- **Pacchetti provenienti da diverse sorgenti sono “mescolati” sullo stesso link**
- **Poiché non c’è garanzia di avere una risorsa disponibile, ci può essere conflitto**
- **I pacchetti in conflitto per lo stesso link sono inseriti in un buffer del router**

Gestione del conflitto

- Si bufferizzano i pacchetti in conflitto per lo stesso link
 - Il buffer determina in pratica una coda di pacchetti che può essere processata in ordine FIFO (First-In-First-Out), ma non necessariamente (es., in base alla priorità)
- **Congestione** = riempimento del buffer



Trasmissioni e conflitti nel packet switching

Comunicazione store and forward:

(i pacchetti si muovono di un hop alla volta)

- **trasmessi su un link, arrivano ad un router**
- **aspettano (presso il router), il loro turno per poter essere trasmessi sul successivo**

Trasmissioni e conflitti nel packet switching

Conflitto di risorse

- **La domanda aggregata di risorse può eccedere la quantità disponibile**
- **Non essendoci prenotazione, si possono creare congestioni (impreviste):**
 - i pacchetti rimangono accodati (se c'è spazio) in attesa di poter utilizzare il link
 - Se la coda è piena, il pacchetto viene perduto
- **Possibilità di utilizzare un link differente a seconda dello stato della rete**

Packet switching: pro e contro

PRO:

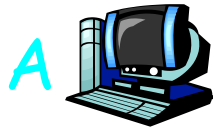
- C'è condivisione di risorse
- Non c'è la necessità di prenotare risorse end-to-end
- Il packet switching è ottimo per dati che arrivano in gruppi

CONTRO: Rischi di congestione:

- Ritardo e perdita di pacchetti
- E' necessario un protocollo che garantisca almeno le seguenti due proprietà:
 - Trasferimento dei dati affidabile (in grado di capire se c'è perdita di pacchetti e in grado di provvedere)
 - Controllo della congestione

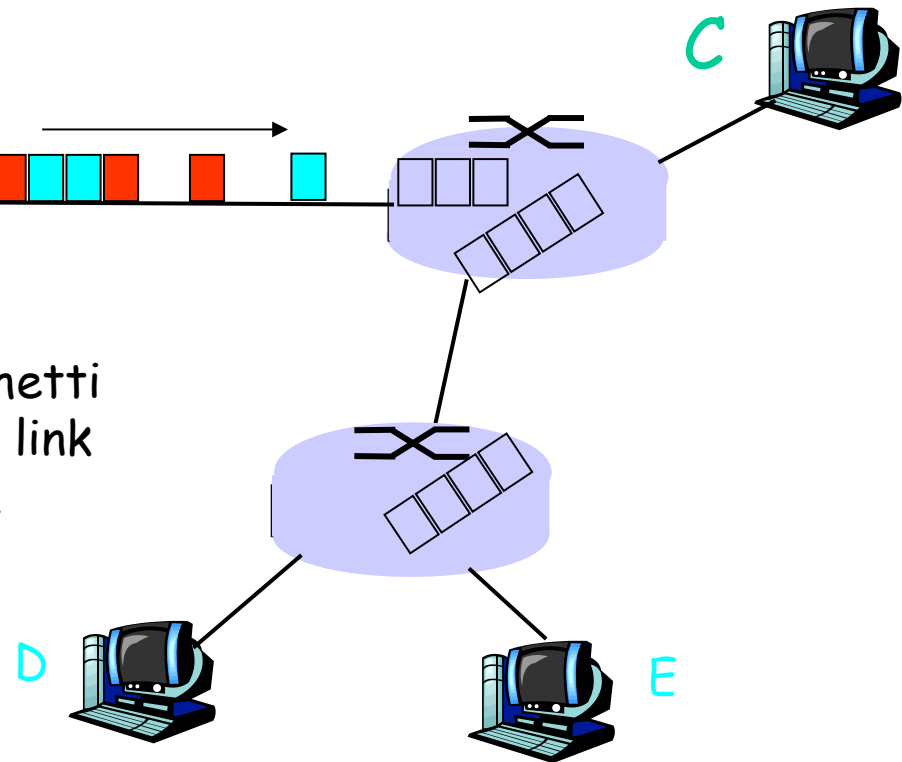
Packet switching

Invia messaggio
composto da 8 pacchetti



Invia messaggio
composto da 4 pacchetti

Coda di pacchetti
in attesa del link
di uscita



“Analogia del ristorante”

- **Circuit switching** = con prenotazione del tavolo
- **Packet switching** = senza prenotazione

Metrica di prestazione

- **Bandwidth (banda di trasmissione):**
quantità di dati trasmessi per unità di tempo
- **Tipicamente:**
 - Unità di tempo = secondo
 - Quantità di dati trasmessi = multipli di bit
- **Quindi, metriche tipiche sono:**
 - Kbps o Kbit/s → Kilo bit per secondo
 - Mbps o Mbit/s → Mega bit per secondo
 - Gbps o Gbit/s → Giga bit per secondo

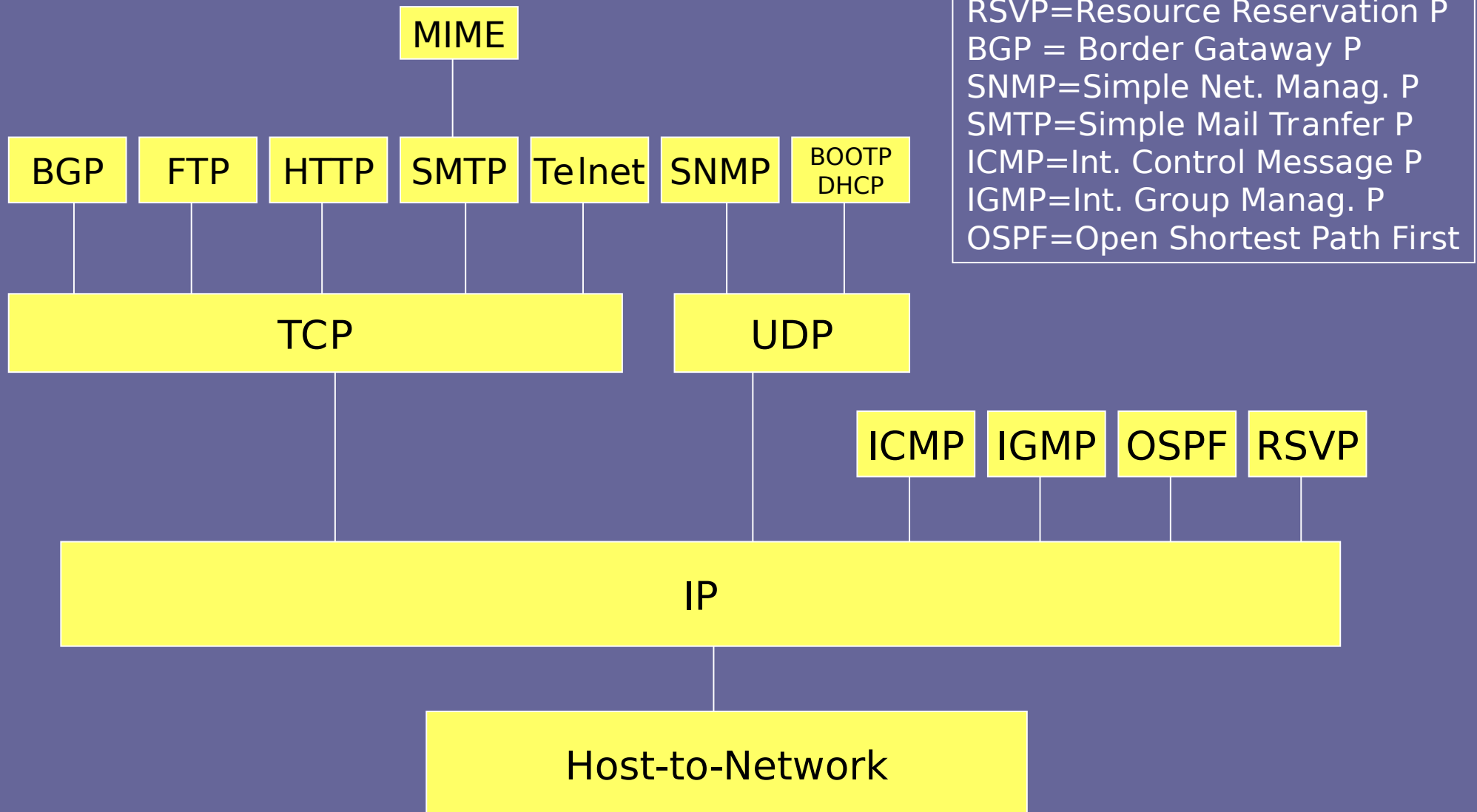
Vantaggi del packet switching

Esempio:

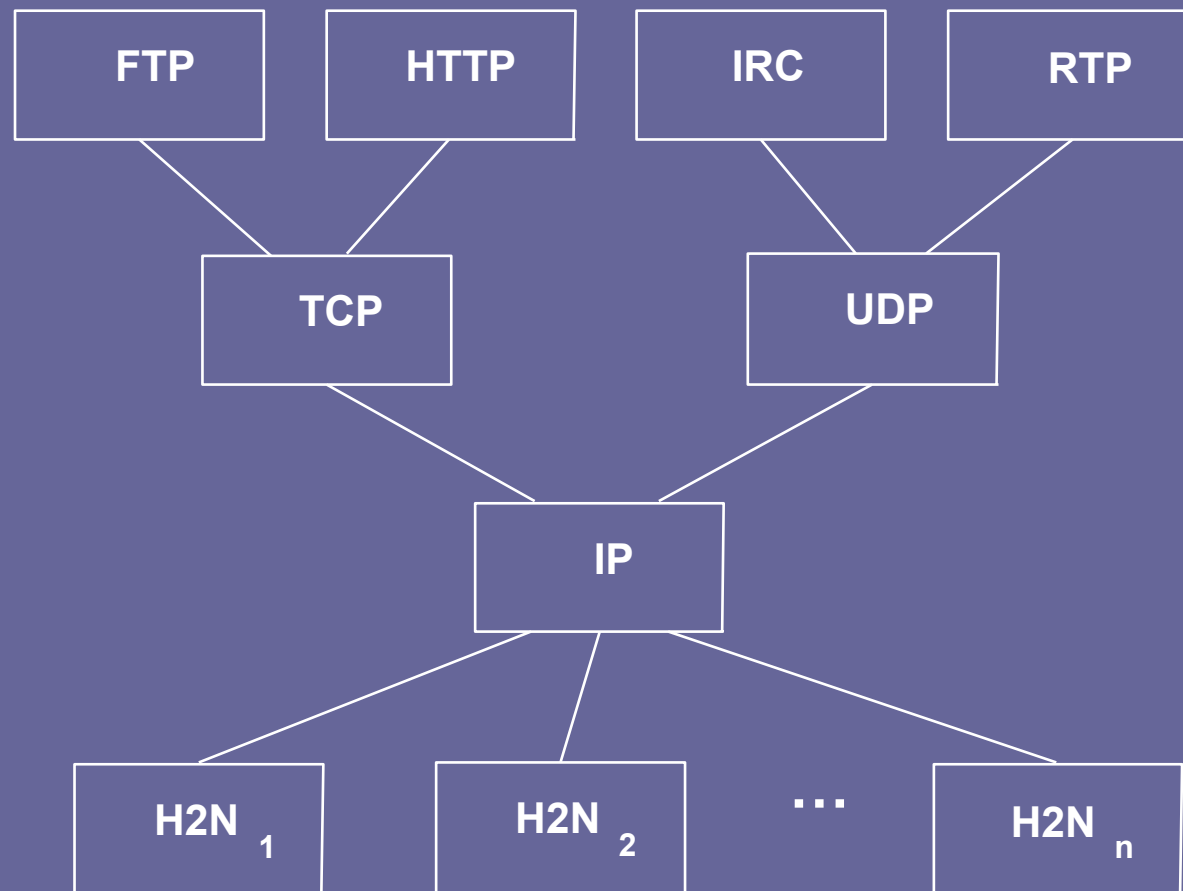
- **Link a 1 Mbps**
- **Ciascun utente richiede 0.1 Mbps quando trasmette, ed è attivo il 10% del tempo**
- **Circuit switching: può supportare al più 10 utenti**
- **Packet switching: con 35 utenti, la probabilità che più di 10 utenti trasmettano contemporaneamente è bassissima (0.0004), quindi è possibile far comunicare 35 utenti sulla stessa linea con minimi rischi di conflitti**

Modulo 4: Analisi dello stack TCP/IP

Molti protocolli, ma non a tutti i livelli



Progetto Internet "a clessidra"



Livello 1-2 (host-to-network)

- **I primi due livelli (fisico e data link) non sono separati, nel senso che connessione fisica e protocollo data link sono interdipendenti**
- **Pertanto, nel caso dello stack TCP/IP è più corretto parlare di un livello host-to-network (h2n) che comprende i primi due livelli**
- **Esempi di protocolli h2n:**
 - Protocollo per LAN: Ethernet, token-ring
 - Protocollo per connessioni via modem: PPP
 - Protocollo per connessioni LAN wireless: 802.11

Livello 3 (network): Protocollo IP

- **Protocollo per la consegna dei pacchetti da un host mittente ad un host destinatario**
- **Servizi aggiuntivi rispetto a h2n**
 - identificativo univoco di ciascun host (indirizzo IP)
 - comunicazione logica tra host

Livello 3 (network): Protocollo IP

- **Ma**

- privo di connessione: ogni pacchetto è trattato in modo indipendente da tutti gli altri
- non affidabile: la consegna non è garantita (i pacchetti possono essere persi, duplicati, ritardati, o consegnati senza l'ordine di invio)
- consegna con impegno: tentativo di consegnare ogni pacchetto (l'inaffidabilità deriva dalle possibili congestioni della rete o guasti dei nodi/router)

Livello 4 (transport)

- **Il livello transport estende il servizio di consegna con impegno proprio del protocollo IP tra due host terminali ad un servizio di consegna a due processi applicativi in esecuzione sugli host**
- **Servizi aggiuntivi rispetto a IP**
 - multiplazione e demultiplazione messaggi tra processi
 - rilevamento dell'errore (mediante checksum)
- **Esempi di protocolli transport**
 - UDP (User Datagram Protocol)
 - TCP (Transmission Control Protocol): offre servizi aggiuntivi rispetto a UDP

Livello 4 (transport) [UDP]

- **Protocollo che fornisce un livello di trasporto dell'informazione connectionless**
- **Specifica in [RFC 768]**

Livello 4 (transport) [TCP]

- **Protocollo che fornisce un livello di trasporto affidabile e orientato alla connessione**
- **Servizi aggiuntivi rispetto a UDP**
 - orientato alla connessione: comprende l'instaurazione, l'utilizzo e la chiusura della connessione
 - orientato al flusso di dati: considera il flusso di dati dall'host mittente fino al destinatario (→ considera sia rete sia host terminali)
 - ...

Livello 4 (transport) [TCP]

- **Servizi aggiuntivi rispetto a UDP**

- ...
- trasferimento con buffer: i dati sono memorizzati in un buffer e poi inseriti in un pacchetto quando il buffer è pieno
- connessione full duplex (bi-direzionale): una volta instaurata una connessione, è possibile il trasferimento contemporaneo in entrambe le direzioni della connessione

Livello 5 (application)

- **Il livello application utilizza il livello di trasporto dell'informazione tra processi in esecuzione su host terminali per realizzare applicazioni di rete**
- **Esempi protocolli applicativi**
 - ftp
 - telnet
 - http
 - smtp
 - irc
 - ...

NB: Applicazioni di rete \neq protocolli applicativi