

# Reti di Telecomunicazioni



Modelli

---



# Autori

---

Queste slides sono state scritte da

Michele Michelotto:

[michele.michelotto@pd.infn.it](mailto:michele.michelotto@pd.infn.it)

che ne detiene i diritti a tutti gli effetti



# Copyright Notice

Queste slides possono essere copiate e distribuite gratuitamente soltanto con il consenso dell'autore e a condizione che nella copia venga specificata la proprietà intellettuale delle stesse e che copia e distribuzione non siano effettuate a fini di lucro.



# Layer: Modello OSI e IP

---

Introduzione

**Layer: Modello ISO/OSI e TCP/IP**

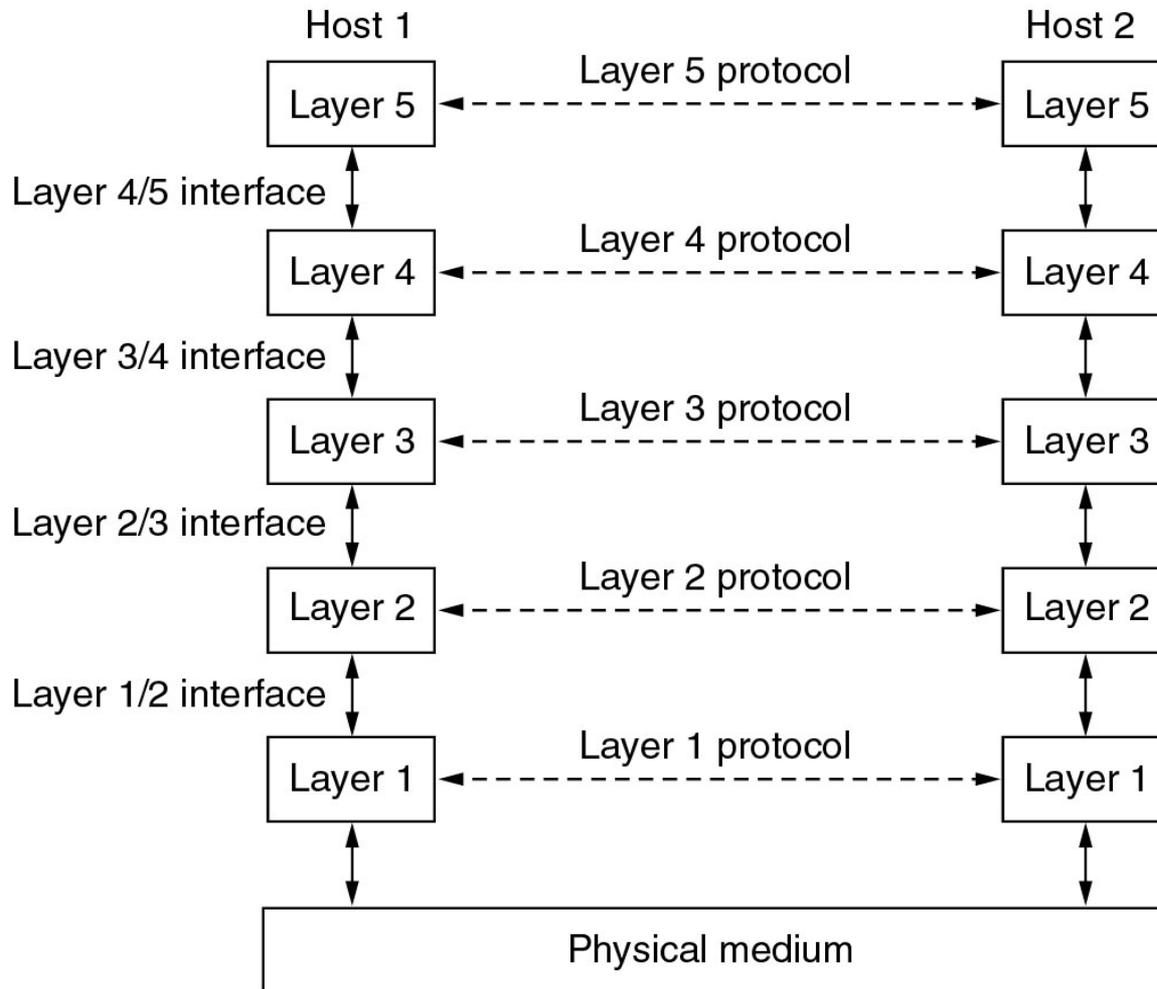
Physics Layer

Data Link Layer

MAC sublayer



# Reti a Layer di protocolli





# Esempi di rete a layer

- Vediamo due esempi di architetture di rete del tipo a layer
  - **ISO/OSI** (Open System Interconnect) Reference Model
  - **TCP/IP** Reference Model
- I protocolli OSI sono usati molto raramente ma il modello ha validità generale ed è sempre valido
- Il modello TCP/IP non è invece molto usato ma i suoi protocolli invece lo sono moltissimo



# OSI Reference Model



- Modello a sette livelli, ispirato ai seguenti principi:
  - Un nuovo layer va creato quando si richiede un nuovo livello di astrazione
  - Ogni layer deve eseguire una funzione ben definita
  - La funzione di ogni layer deve essere scelta oculatamente per favorire protocolli standard internazionali
  - I confini tra i layer vanno scelti cercando di minimizzare il flusso di informazioni tra le interfacce
  - Il numero di layer deve essere sufficiente per separare distinte funzioni in layers separati ma piccolo per non avere un'architettura troppo complicata.



# Modello OSI

## Livello OSI (inf. Scambiate)

## Funzioni fornite

Application (APDU)	Es: file o message transfer, emulazione di terminale
Presentation (PPDU)	Formattazione dei dati e criptazione
Session (SPDU)	Apertura e gestione delle sessioni
Transport (TPDU)	Consegna di dati end-to-end reliable e unreliable
Network (Packet)	Consegna pacchetti di informazioni, il routing
Data Link (Frame)	Trasferisce pezzi di informazioni, framing, error checking
Physical (Bit)	Transmissione di dati binari dal mezzo fisico



# Modello OSI

- Non specifica un'architettura di rete, infatti non specifica i servizi e i protocolli da usare in ogni layer
- Dice solo di cosa si deve occupare un layer
- ISO ha prodotto anche degli standard per tutti i layer, ma questi non fanno parte del modello stesso. Ognuno è stato pubblicato a parte in standard internazionali separati (X400, X500)



# Physical Layer

- Trasmette bit (senza significato o struttura) sopra un canale di comunicazione
  - Quando trasmetto un bit 1 devo ricevere un bit 1 e non un bit 0
    - Quanti Volt significano bit 1 e quanti bit 0?
    - Quanti (nano)secondi dura un bit?
    - La trasmissione può essere simultanea nelle due direzioni?
    - Come stabilisco e chiudo una connessione?
    - Quanti pin ha il connettore di comunicazione e che significato ha ogni pin?
  - I criteri di progettazione coinvolgono questioni meccaniche, elettriche e di temporizzazione e il mezzo fisico che sta sotto il physical layer (a volte detto Layer 0)



# Data Link Layer

- Trasforma la linea di comunicazione fisica (raw) in una linea che appare al Network Layer come libera da errori di trasmissione.
  - Spezzando i dati di ingresso in frammenti (centinaia, migliaia di bytes) chiamati **data frames** (trame, buste).  
Es. con bit pattern all'inizio e alla fine
  - Trasmette i frames in modo sequenziale
  - Gestisce i **frame di acknowledgements**
  - Decide come trattare i frame danneggiati, duplicati o perduti e il feedback verso il mittente per flow control

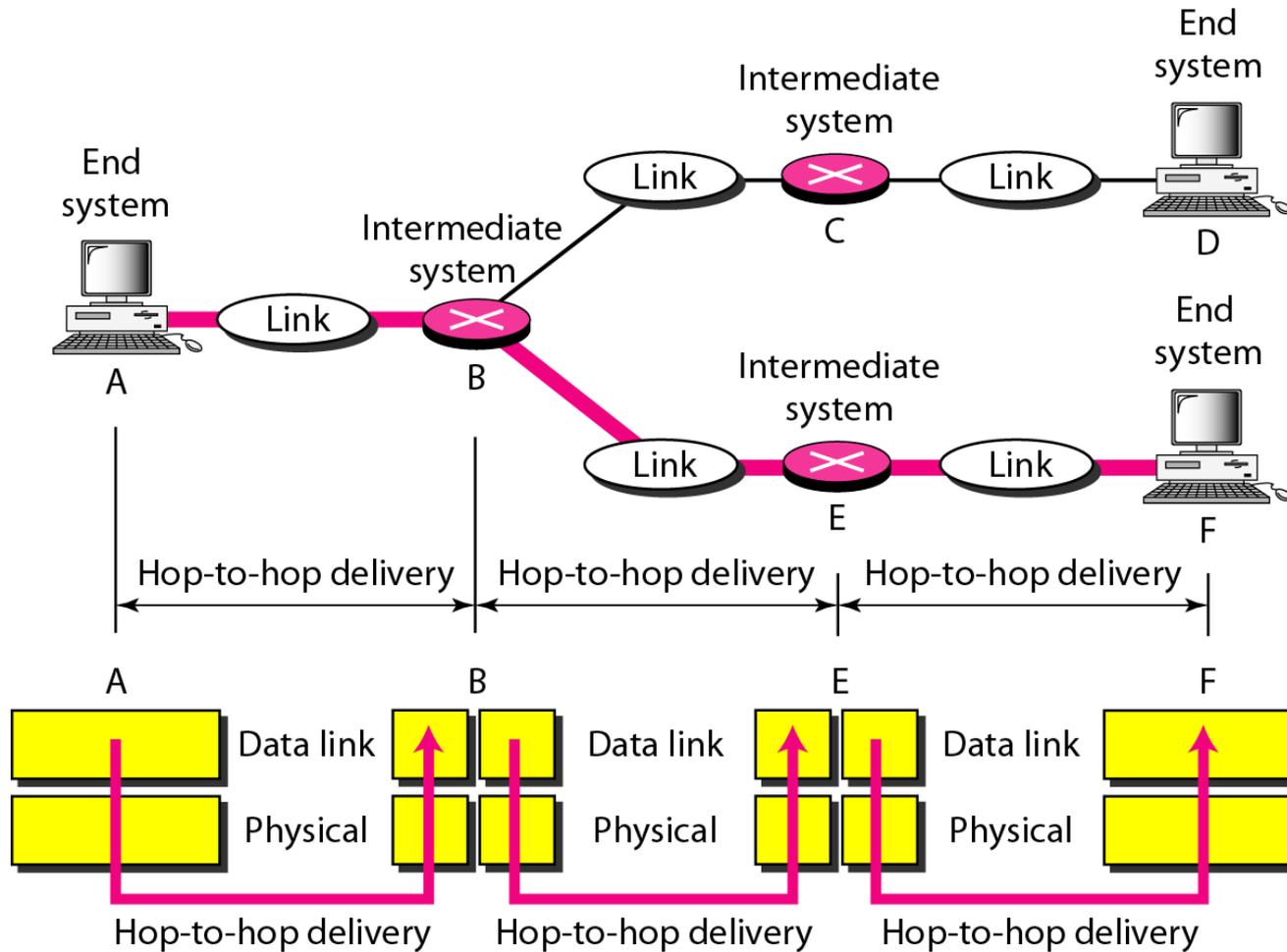


# MAC SubLayer

- In una rete broadcast c'è anche il problema di decidere i criteri di accesso al canale quando questo è condiviso da tanti attori
- Il livello DataLink viene diviso di due. Un sotto layer, **Medium Access Sublayer**, si occupa di questo problema
- Il livello OSI però non li vede come due livelli distinti



# Data Link



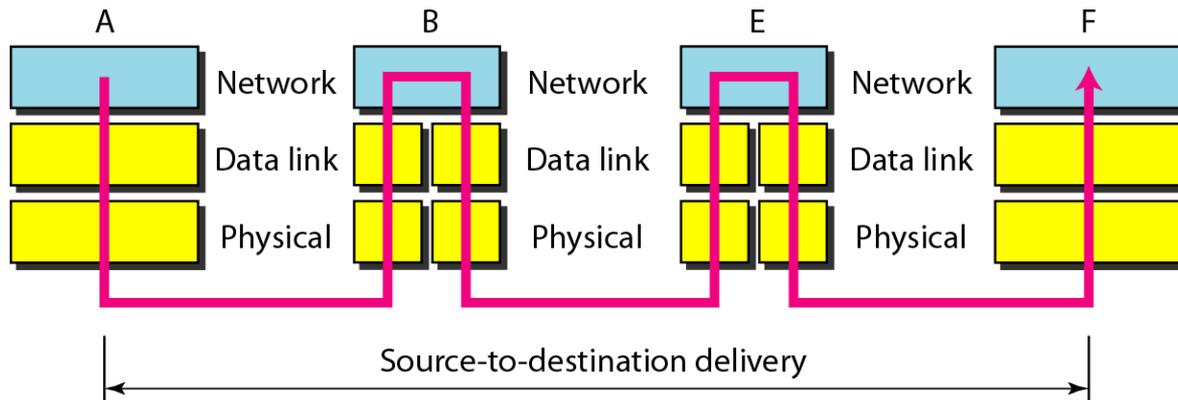
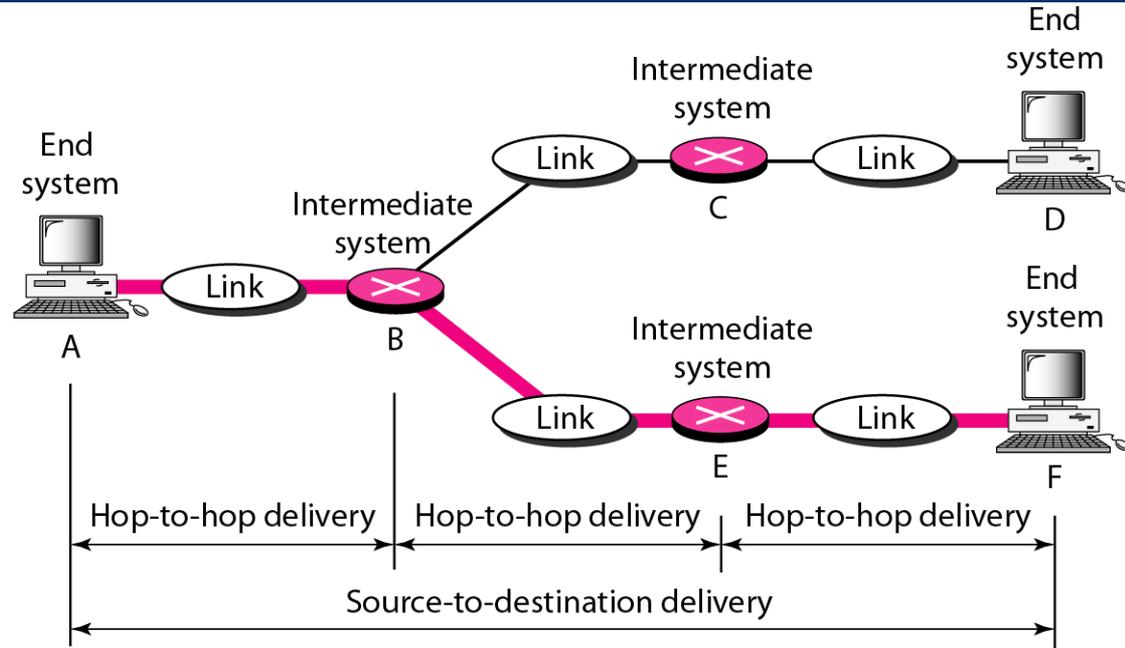


# Network Layer

- Controlla che i pacchetti vadano dal mittente al destinatario
  - I percorsi possono essere descritti in tabelle statiche
  - Possono essere determinati all'inizio di ogni conversazione
  - Possono essere invece molto dinamici ed ogni pacchetto decide la strada in base al carico della rete
- Controlla la congestione a livello di sottorete
  - In generale problemi di delay, transit time, jitter sono gestiti a livello Network
- Se il pacchetto viaggia tra due reti il formato di indirizzamento o le dimensioni del pacchetto potrebbe essere diversi. Il Network Layer gestisce queste funzioni



# Network Layer



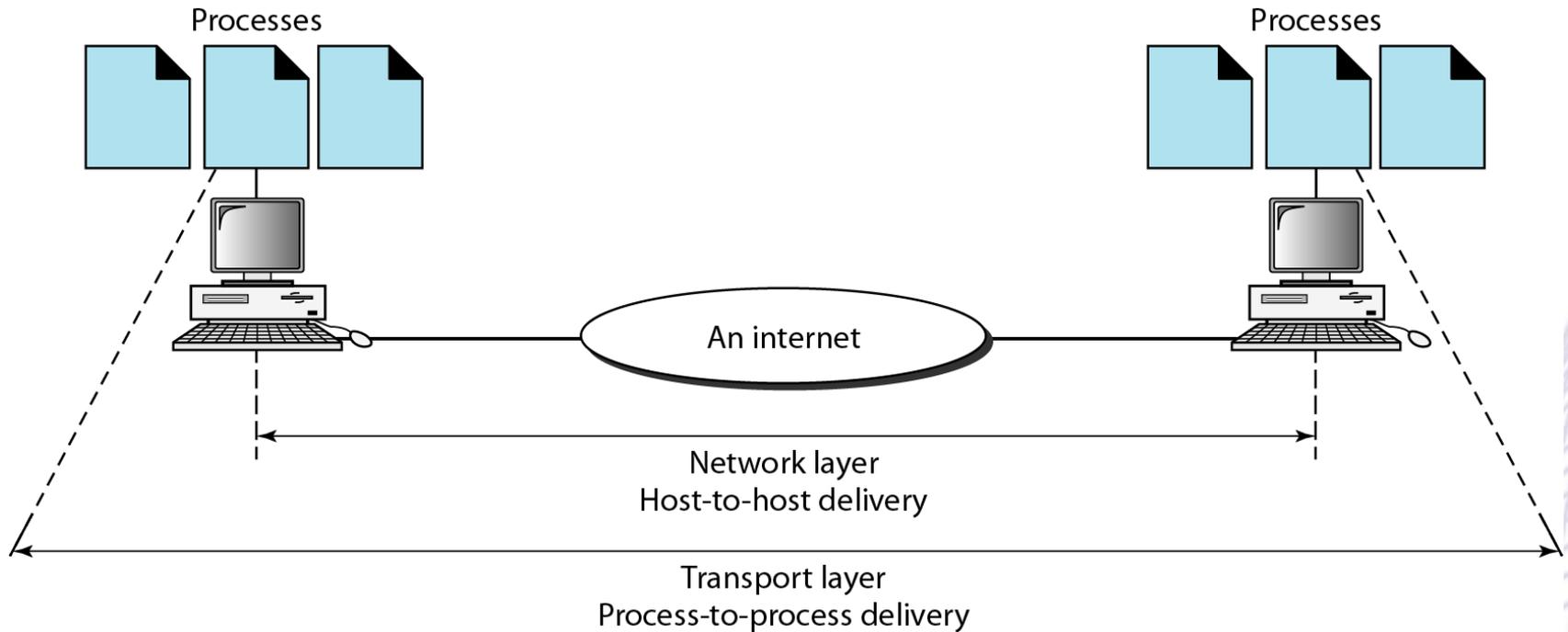


# Transport Layer

- Prende i dati dal Session Layer, li spezza eventualmente in pacchetti che vengono passati al Network e si assicura che i pacchetti arrivino correttamente all'altro lato
- Di solito crea una connessione distinta per ogni richiesta di trasporto ma se è richiesto un alto throughput può creare connessioni di network multiple o al contrario può multiplexare diverse connessioni di trasporto sulla stessa connessione di network



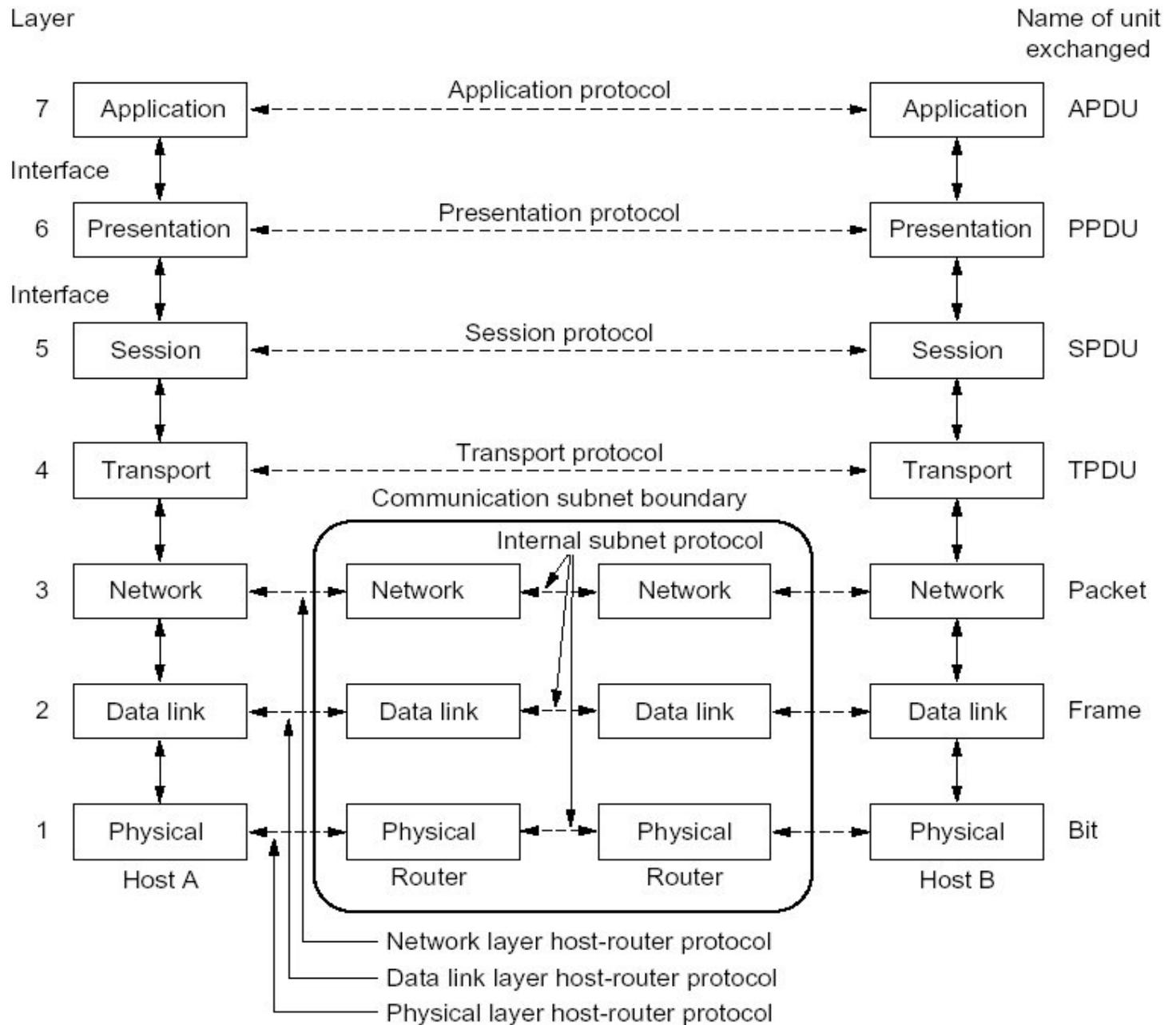
# Consegna end to end





# Transport Layer

- Determina che servizio fornire; il più comune è un canale punto a punto error free ma potrebbe mandare messaggi fuori sequenza o messaggi a destinazioni multiple.
- È un vero layer end-to-end. Il programma del mittente comunica con un programma del destinatario
- Ai livelli inferiori invece i protocolli sono tra le diverse macchine e i vicini immediati e non tra macchina mittente e destinataria che possono essere separate da diversi routers



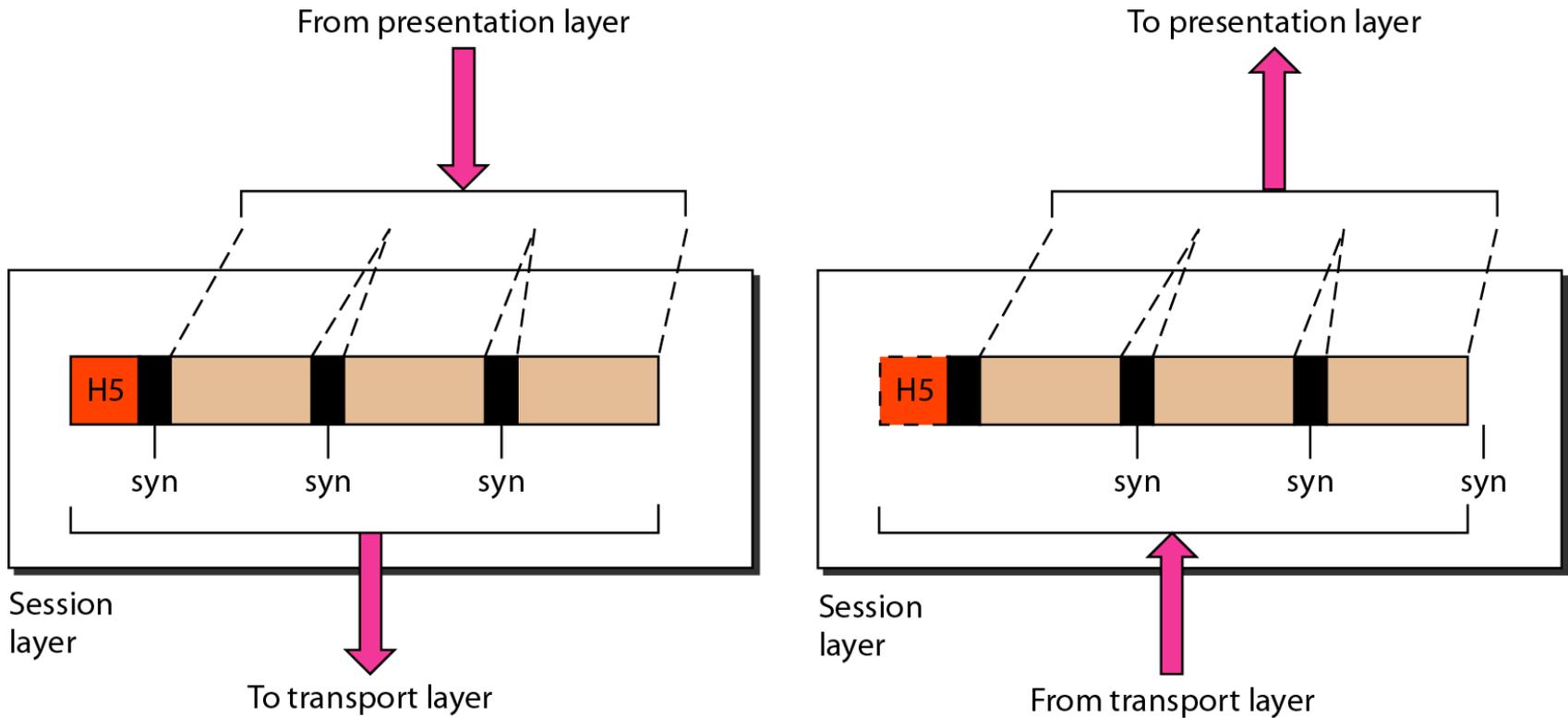


# Session Layer

- Permette di creare **sessioni** tra macchine diverse
  - Permette di trasferire dati come il Transport Layer ma fornisce servizi più evoluti come il file transfer o il login remoto
  - Gestisce il **dialogo di controllo** permettendo al traffico di andare in due direzioni o in una unica direzione e gestisce i turni
  - Gestisce i **token** per evitare che due parti eseguano contemporaneamente un'operazione critica
  - Gestisce la **sincronizzazione** (Es. Un file transfer di due ore che crasha in media ogni ora. Il S.L. offre la possibilità di inserire dei checkpoints permettendo di ritrasferire solo i dati non ancora arrivati)



# Session Layer



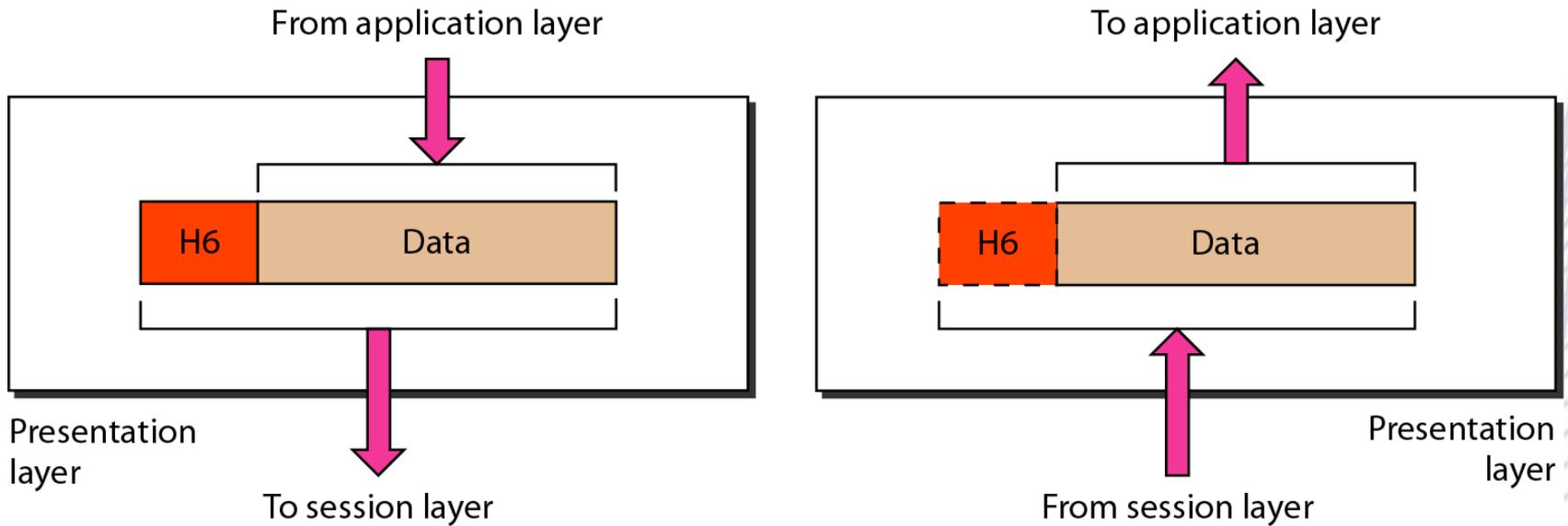


# Presentation Layer

- Offre funzioni richieste spesso e tali da richiedere una soluzione generale invece che lasciarla all'utente
  - Traduzione tra sistemi di codifica diversa
  - Cifratura e decifratura dei dati
  - Compressione
- Al contrario dei livelli inferiori che spostano bit questo layer si interessa della semantica e della sintassi delle informazioni trasmesse. (ad esempio ASCII, Unicode, EBDIC).



# Presentation





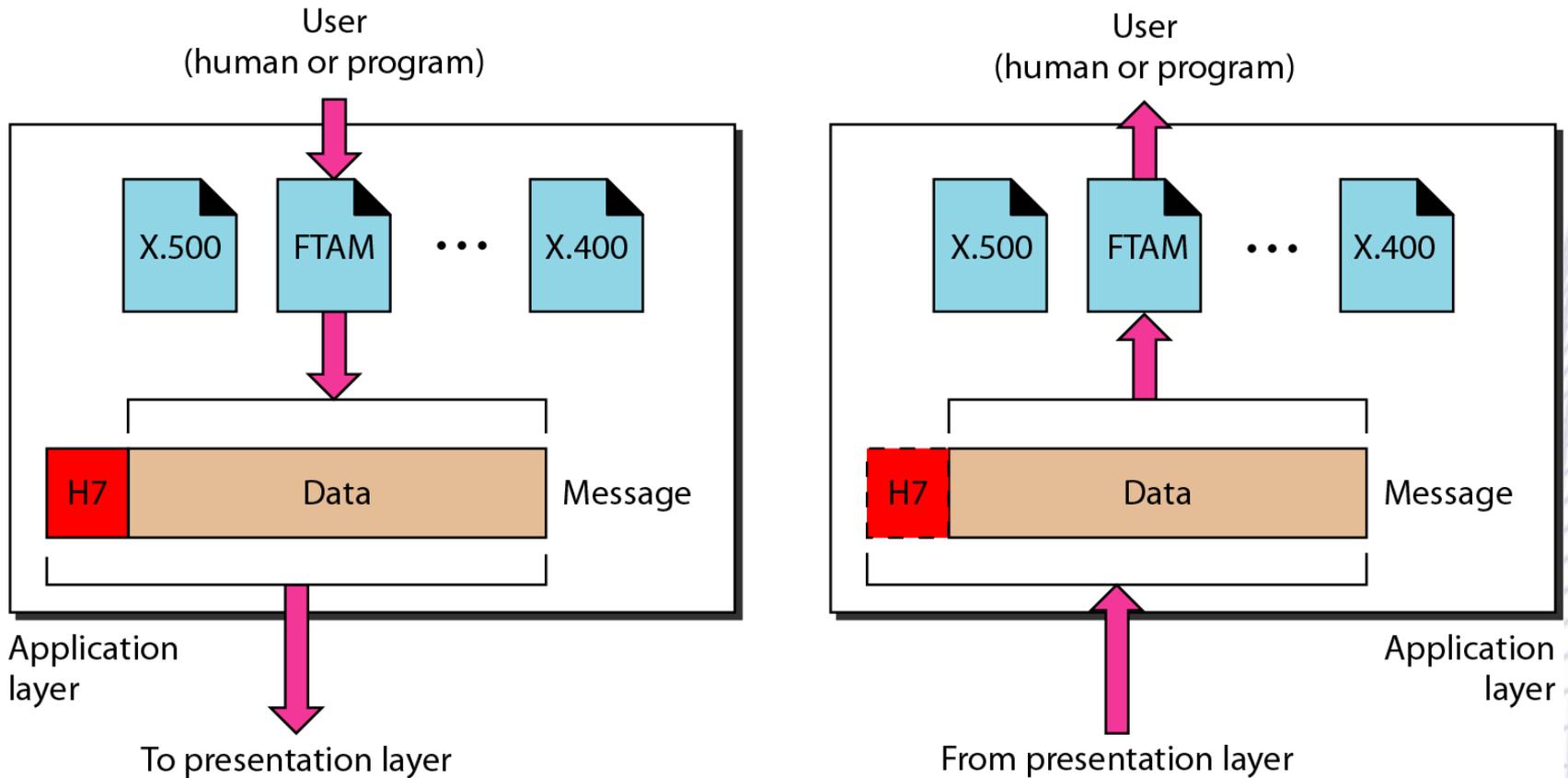
# Application Layer



- Vari protocolli per servizi di rete all'utente finale
  - Terminale Virtuale: Per supportare centinaia di tipi di terminali diversi che descrivono il layout dello schermo o come muovere il cursore
  - Gestione File: permette di accedere e modificare file remoti
  - Servizi di Posta Elettronica
  - Servizi di Directories (es gestione delle password)
  - File Transfer tra filesystem diversi, gestione delle convenzioni di naming, rappresentazioni diverse della linea di testo
  - HTTP che sta alla base del World Wide Web

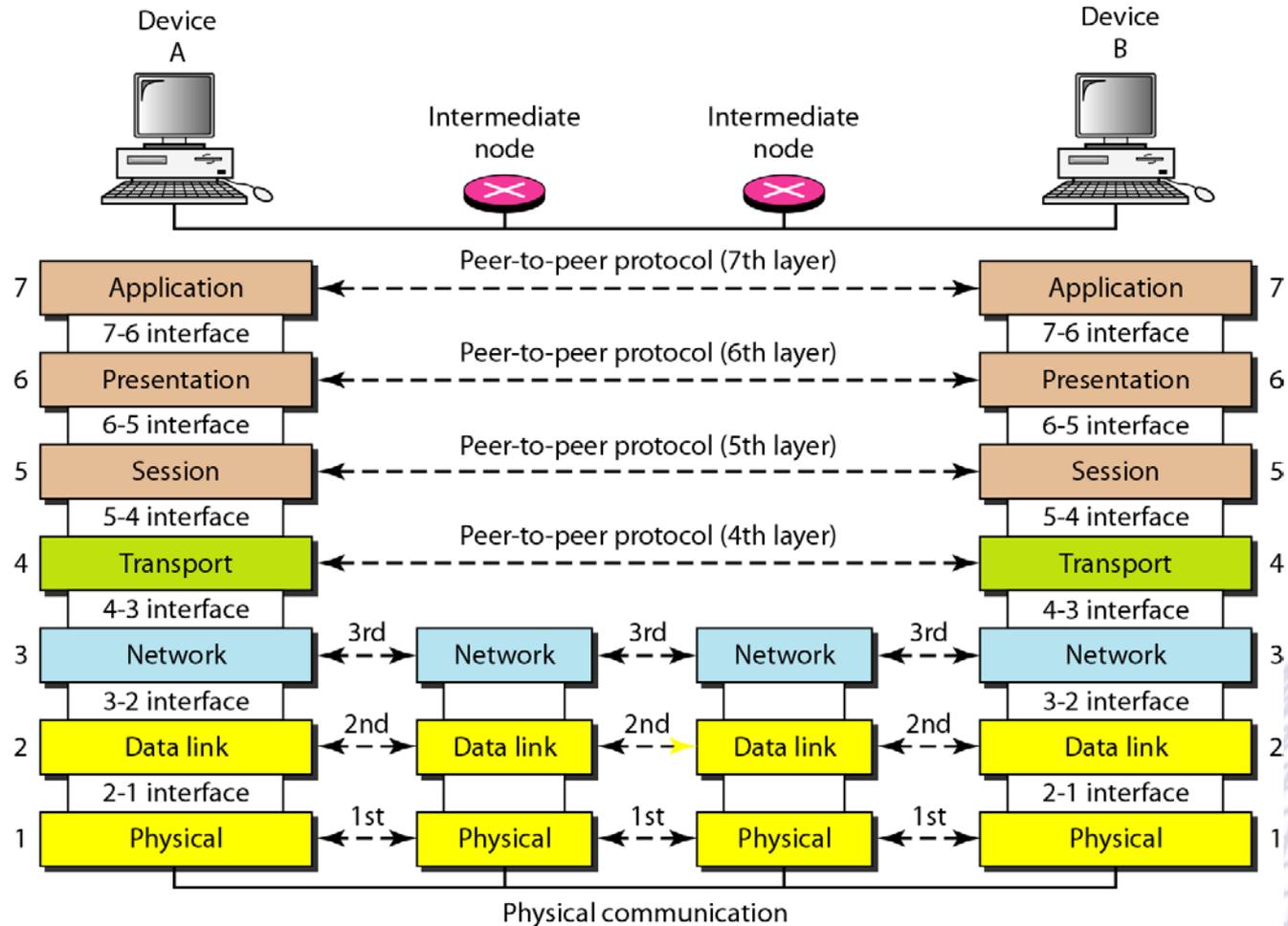


# Application layer





# Livelli OSI





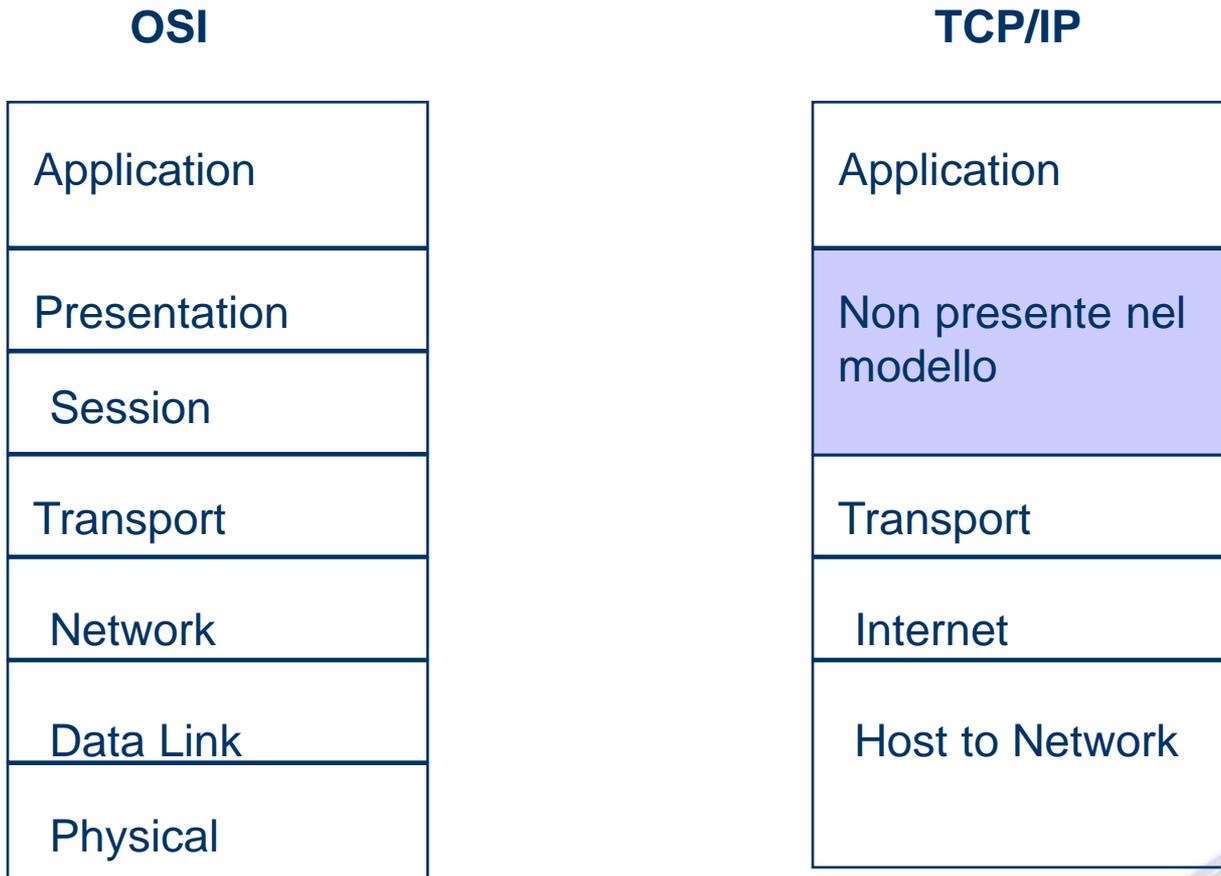
# TCP/IP Reference Model



- Modello nato negli anni '70 in università americane su finanziamenti DoD
- Esigenza di interconnettere centinaia di università e siti governativi via linee telefoniche affittate e poi via satellite/radio.
- Progettato per funzionare con linee inaffidabili o soggette a crolli improvvisi
- Architettura flessibile per accontentare diversi requirements applicativi (file-transfer, Voip)

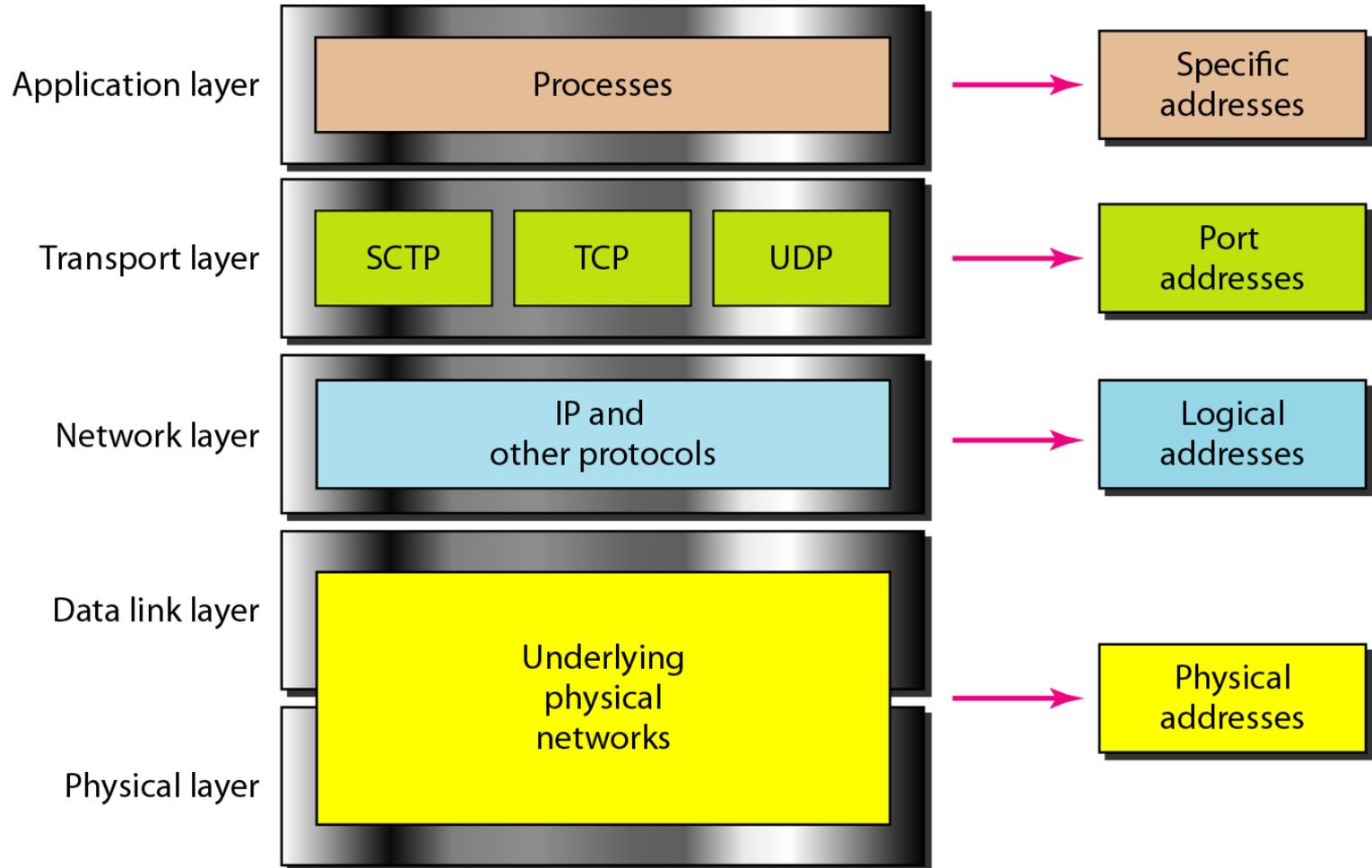


# TCP/IP Reference Model





# 4 livelli e 4 indirizzi





# Internet Layer

- I requirements sono stati soddisfatti da una rete packet switching basato su un layer internetwork connectionless. → **internet layer**
- Gli host possono iniettare pacchetti nella rete e lasciarli viaggiare in modo indipendente, questi possono arrivare in ordine diverso e viene lasciato ai layer superiori il problema di riordinarli
- Parola **internet** usata in senso generale sebbene questo layer sia usato in **Internet**



# Internet Layer

- Funziona come il sistema postale (cfr internet over snail mail)
- Se uno spedisce diverse lettere queste probabilmente arriveranno, seguendo forse strade diverse ma in modo trasparente all'utente.
- Il layer definisce un formato del pacchetto ufficiale e un protocollo **IP (Internet Protocol)**.
- Il problema del routing dei pacchetti è fondamentale → simile al Network Layer di OSI



# Transport Layer

- Permette la conversazione tra due peer entities sul nodo mittente e destinatario come il T.L. nel modello OSI.
- **TCP (Transmission Control Protocol)**
  - offre una connessione affidabile e connection oriented che permette ad un flusso di byte di essere consegnato senza errori ad un'altra macchina della internet, frammentandolo in pacchetti e riassemblelandoli a destinazione. Gestisce anche il controllo del flusso.



# Transport Layer

- **UDP (User Datagram Protocol)**

- Non affidabile e connectionless per applicazioni che non vogliono usare i servizi di TCP e preferiscono invece gestire per proprio conto la sequenzialità e il controllo del flusso
- Gestisce anche messaggi one-shot, richieste client-server di tipo request-reply
- Applicazioni voce e video dove la tempestività è più importante della consegna accurata

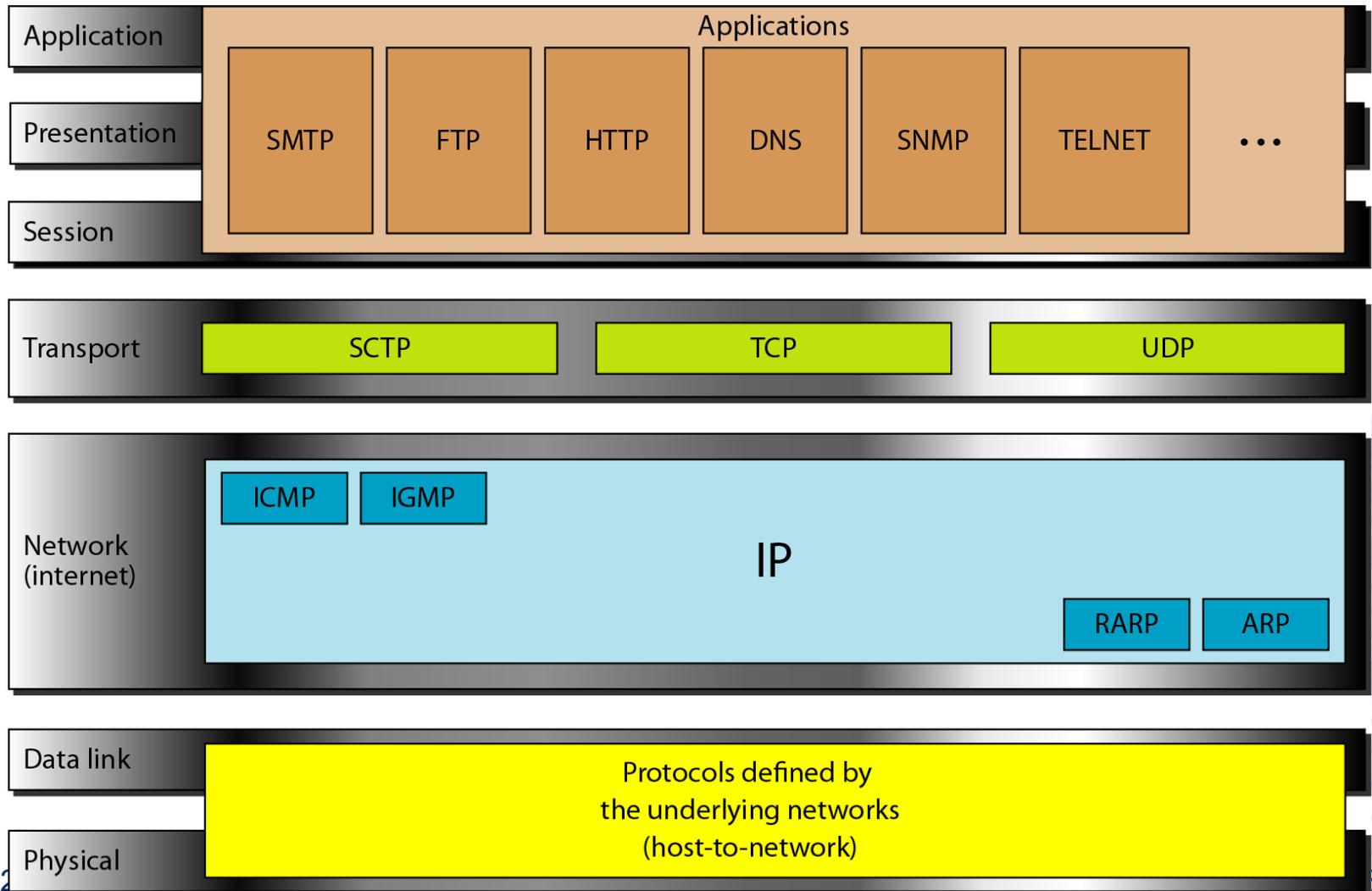


# Application Layer

- L'Application Layer contiene diversi protocolli di alto livello tra cui:
  - Terminale Virtuale: TELNET
  - File Transfer: FTP
  - Posta Elettronica: SMTP, POP, IMAP
  - Conversione nomi-indirizzi: DNS
  - Diffusione di articoli di notizie: NNTP
  - Iper testo distribuito: HTTP
- Non ci sono Session e Presentation Layer
  - non se ne sentiva l'utilità. In effetti nel modello OSI sono usati raramente dalle applicazioni



# TCP/IP model





# Host to Network Layer

- Il protocollo IP viene implementato al di sopra di praticamente tutte le tecnologie di interconnessione ma queste non fanno parte del modello TCP/IP.
- In realtà è una interfaccia tra il livello data link e tutto quello che si trova sotto
- Il modello dice solo che l'host deve connettersi alla rete e mandarvi sopra pacchetti IP



# Aspetti in comune

- Concetto di Stack di protocolli indipendenti
- Funzionalità dei layer sostanzialmente simili
  - In entrambi i layer dal basso fino a quello di trasporto forniscono un servizio di trasporto end-to-end indipendente dalla rete
  - I livelli superiori dal livello di trasporto sono orientati agli utenti e alle applicazioni



# Differenze tra i modelli

- Differenze tra i modelli (attenzione non tra gli stacks)
  - Il modello OSI distingue chiaramente tra **Servizi** (cosa fa), **Interfacce** (come accedere) e **Protocolli** (come lo fa internamente)
  - Distinzione meno chiara nel TCP/IP almeno in origine. Poi si è cercato di adattarlo al modello OSI
  - Quindi i protocolli in OSI sono nascosti meglio e quindi sono cambiabili più facilmente.



# Differenze tra i due modelli



- In OSI prima viene il modello generale e poi sono stati progettati i protocolli
- Il livello Data Link era pensato per link punto punto e non si aspettavano per esempio le reti broadcast → non hanno pensato al layer di Media Access Control inserito a forza in seguito nel modello.
- In TCP/IP sono venuti prima i protocolli quindi il modello non descrive bene le reti non TCP/IP.



# Differenze tra i due modelli

- OSI model supporta connection oriented e connectionless a livello network ma solo connection oriented a livello transport dove è importante essendo visibile all'utente
- TCP/IP invece è connectionless al livello network e permette sia connectionless che connection oriented al livello di trasporto permettendo all'utente di scegliere



# Critica di OSI

- Una critica ai modelli e ai protocolli OSI
- Alla fine degli anni 80 sembrava che i protocolli OSI avrebbero dominato il mondo e scalzato gli altri protocolli.
- Non è successo per **4 motivi**
  1. **Bad Timing**
  2. **Bad Technology**
  3. **Bad Implementations**
  4. **Bad Politics**



# Timing



- **Bad Timing**

- Prima c'è la fase di ricerca (discussioni, articoli, conferenze) e poi gli investimenti miliardari delle industrie
- Gli standard vanno scritti tra queste due fasi
- Se sono scritti troppo presto i problemi non sono ancora capiti, ne escono cattivi standard
- Se sono scritti troppo tardi le compagnie hanno ormai investito altrove
- Le implementazioni di OSI arrivarono in ritardo quando ormai TCP/IP era ampiamente diffuso



# Technology



- **Bad Technology**

- La scelta dei 7 layer era politica più che tecnica
- Due livelli (presentation e session) quasi vuoti e altri due (data link e network) troppo pieni
- Modello molto complesso, libroni su libroni
- Protocolli difficile da implementare e inefficienti una volta implementati
- Alcune funzioni come indirizzamento, flow control, error control riappaiono in diversi layer
  - Il controllo dell'errore per esempio è più efficiente se fatto nei livelli superiori mentre ripeterlo in ognuno dei livelli inferiori è spesso non necessario e inefficiente



# Implementazioni

- **Bad Implementations:**

- Prime implementazioni lente, complicate ed enormi
- Poi i prodotti sono migliorati ma l'immagine era compromessa
- Al contrario le prime implementazioni di TCP/IP erano parti del Berkeley Unix ed era semplice, veloce e gratis
- Si formarono grandi comunità di utenti che portarono rapidi miglioramenti e maggiore diffusione



# Politiche



- **Bad politics:**

- Sviluppi dominati dalle telecom, dai ministeri delle comunicazioni europei, Comunità Europea e poi il governo USA.
- Percepito come un insieme di standard inferiori imposti da burocrati



# Critica al modello TCP/IP

- Non distingue chiaramente tra servizi, protocolli, interfacce.
- Il modello TCP/IP non ha validità generale come OSI ma descrive bene solo i protocolli TCP/IP (e non p.es. Bluetooth)
- Il Layer Host to Network non è un vero layer ma un interfaccia (tra Network e Data Link)
- Non distingue e non menziona i livelli fisici e data link
- A volte i protocolli di alto livello sono stati scritti da programmatori poco esperti e nel tempo libero e poi adattati nel tempo per risolvere problemi (prestazioni, sicurezza).



# Conclusioni

- Il modello OSI si dimostra utilissimo per discutere le reti di telecomunicazione
  - Ma i protocolli OSI non sono mai diventati popolari
- Il modello TCP/IP praticamente non esiste
  - Ma i protocolli TCP/IP sono ampiamente usati
- Useremo a volte un modello ibrido, un OSI semplificato
  - Ma ci concentremo su TCP/IP e protocolli collegati oltre che nuovi protocolli

## Modello Ibrido

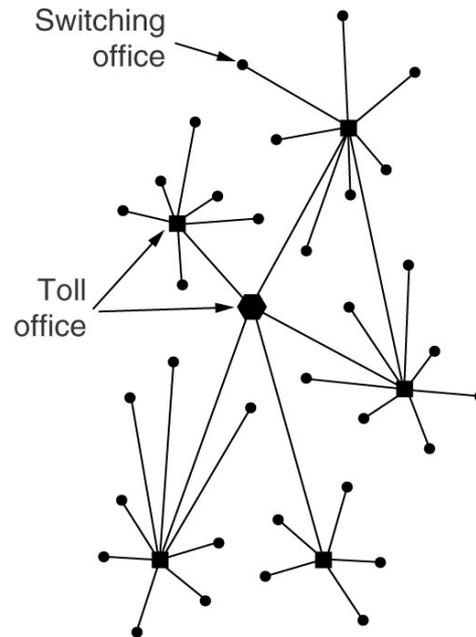
5 Application
4 Transport
3 Network
2 Data Link
1 Physical



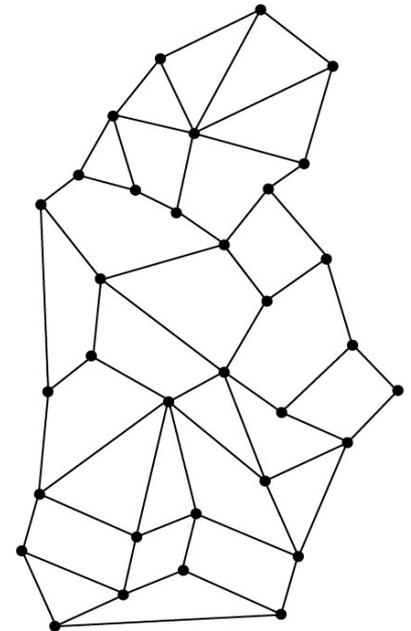
# Arpanet - preistoria



- La rete telefonica ha una struttura gerarchica. Se gli hub sono colpiti tutto si blocca
- Specifiche: Costruire una rete a prova di bomba
- 1960: Progetto a mesh scartato da AT&T



(a)



(b)

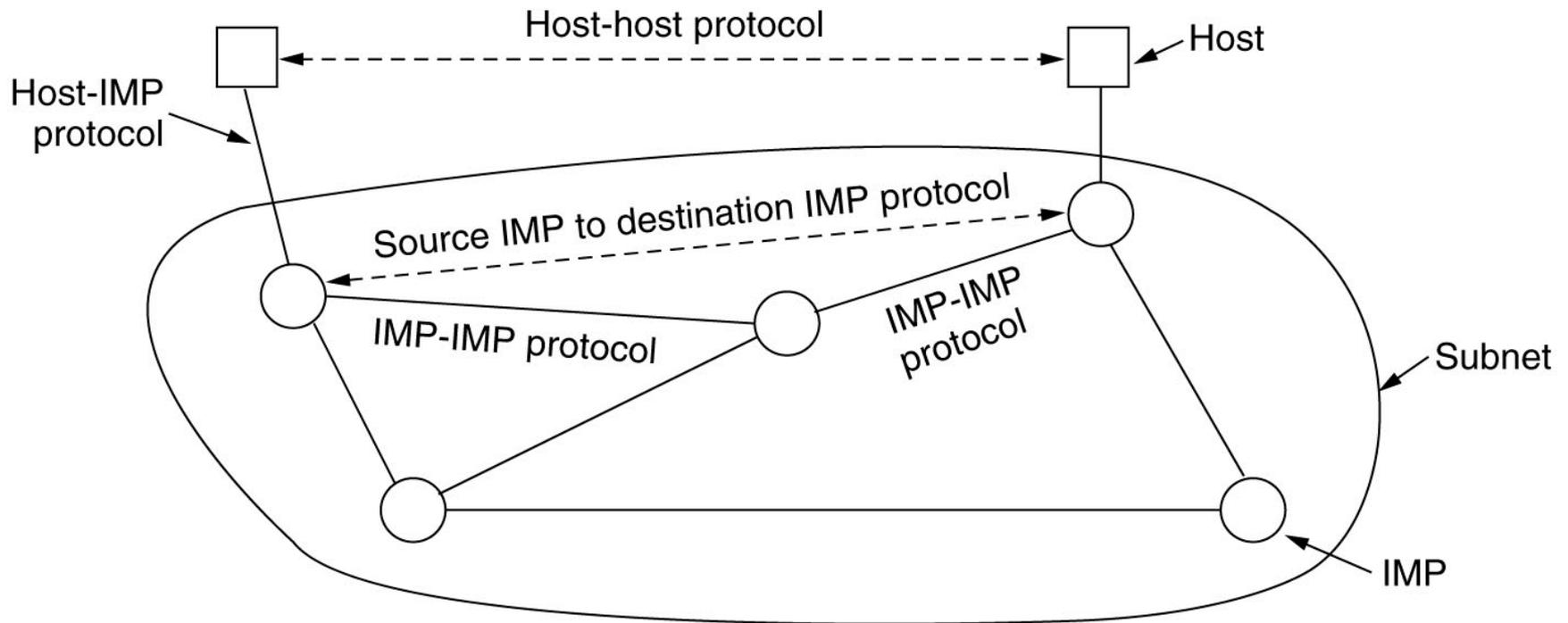


# Arpanet - storia

- ARPA (Advanced Research Project Agency) finanzia progetti di ricerca universitari
- Risultato ARPANET
- Ogni nodo ha
  - un host
  - un IMP (Interface Message Processor) attaccato ad un modem a 56kbps
- L'host manda un messaggio fino a 8063 bit all'IMP che lo spezza in pacchetti di max 1008 bits e li manda a destinazione



# Progetto originale





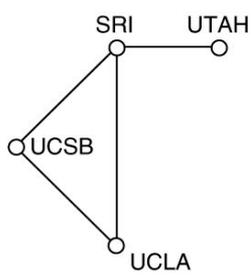
# Protocolli in causa



- Subnet:
  - La parte IMP della connessione host-IMP
  - Il protocollo IMP-IMP
- Host:
  - La parte host della connessione host-IMP
  - Il protocollo host-host
  - Il software applicativo

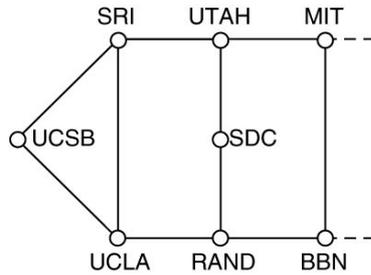


# Evoluzione della rete



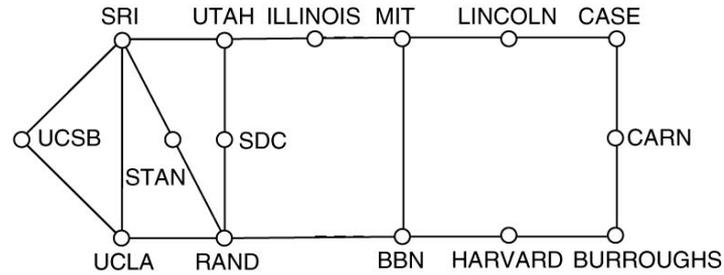
(a)

12/1969



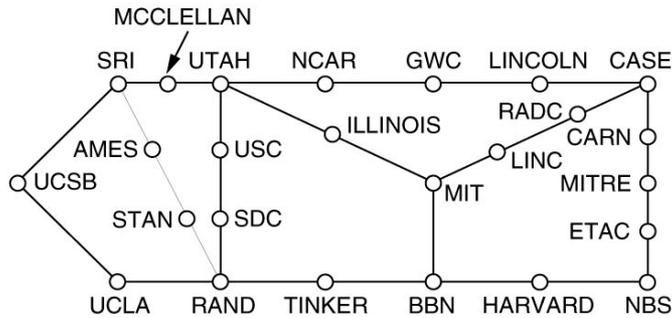
(b)

7/1970



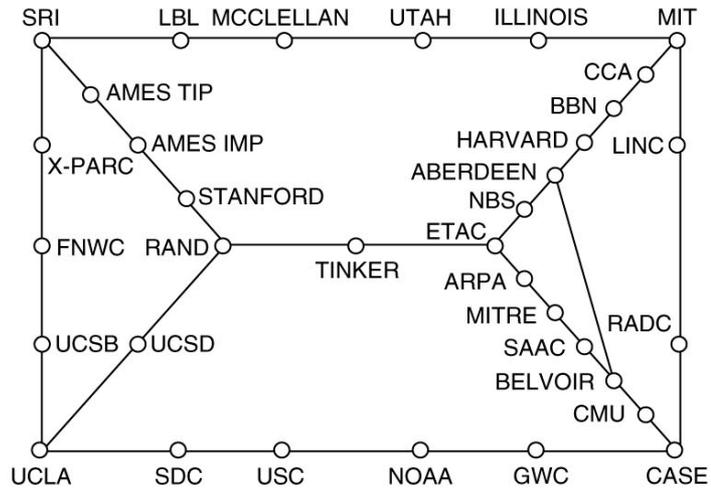
(c)

3/1971



4/1971

(d)



(e)

9/1972



# Fine di Arpanet

- Ricerche su trasmissioni via satellite e radiomobile
- Protocolli Arpanet non adatti a reti diverse
- Invenzione dei protocolli e modello TCP/IP (1974, Cerf e Kahn) pensati per gestire il cosiddetto **INTERNET**working
- Spinta all'integrazione con Berkeley Unix
- Difficile trovare gli host → Nascita di un sistema per mappare nomi e domini a numeri (DNS)



# Reti Connection Oriented



- Arpanet nasce per funzionare, magari anche male, ma funzionare nonostante gravi guasti
- I desideri delle telecom sono altri: far pagare la bolletta agli utenti → Facile con le Connessioni:
  - Quando la connessione inizia si stabilisce un cammino attraverso la rete telefonica che viene mantenuto per tutta la durata della chiamata
  - Tutti i pacchetti seguono la stessa strada



# Motivazioni

- Quality of Service
  - Stabilendo prima la connessione la rete prenota le risorse necessarie (buffer, cpu dei router)
  - Se non ci sono risorse si sente un segnale di “occupato”
  - Se la connessione si stabilisce abbiamo un buon servizio
- Fatturazione
  - Facile: Si paga al minuto per la durata della connessione



# Costi di fatturazione



- Gestire la fatturazione è costoso
- In alcuni casi conviene fatturare “flat” come è sempre successo per la tv
- Alcune telecom lo fanno per le telefonate locali



# Reti a circuiti virtuali



OSI layer	ATM layer	ATM sublayer	Functionality
3/4	AAL	CS	Providing the standard interface (convergence)
		SAR	Segmentation and reassembly
2/3	ATM		Flow control Cell header generation/extraction Virtual circuit/path management Cell multiplexing/demultiplexing
2	Physical	TC	Cell rate decoupling Header checksum generation and verification Cell generation Packing/unpacking cells from the enclosing envelope Frame generation
1		PMD	Bit timing Physical network access

- Vedremo più avanti alcuni tipi di reti a circuito virtuale
- X.25
- Frame Relay
- ATM
- Ora vediamo solo il modello di ATM e il confronto con lo stack OSI

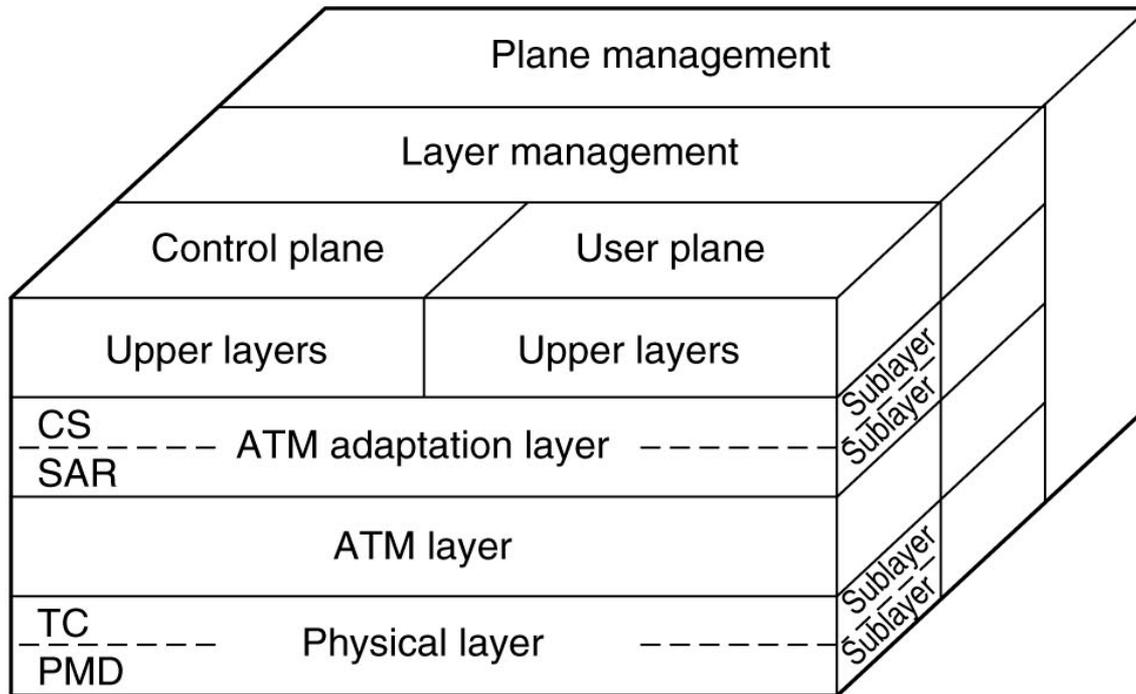


# ATM Reference Model

- Ha il suo modello, distinto da quello OSI o TCP/IP
- Il layer fisico si occupa di tensioni, bit timing etc..
  - Non ci sono regole ma solo si dice che le celle ATM possono essere mandate sul filo da sole o inserendole in un carrier → è indipendente dal sistema trasmissivo



# ATM Reference Model



- CS: Convergence sublayer
- SAR: Segmentation and reassembly sublayer
- TC: Transmission convergence sublayer
- PMD: Physical medium dependent sublayer