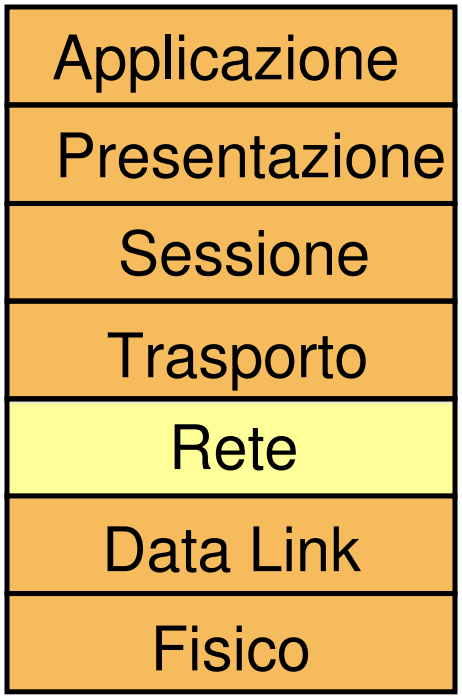
Reti di calcolatori

TCP/IP

Slide a cura di Simon Pietro Romano
spromano@unina.it

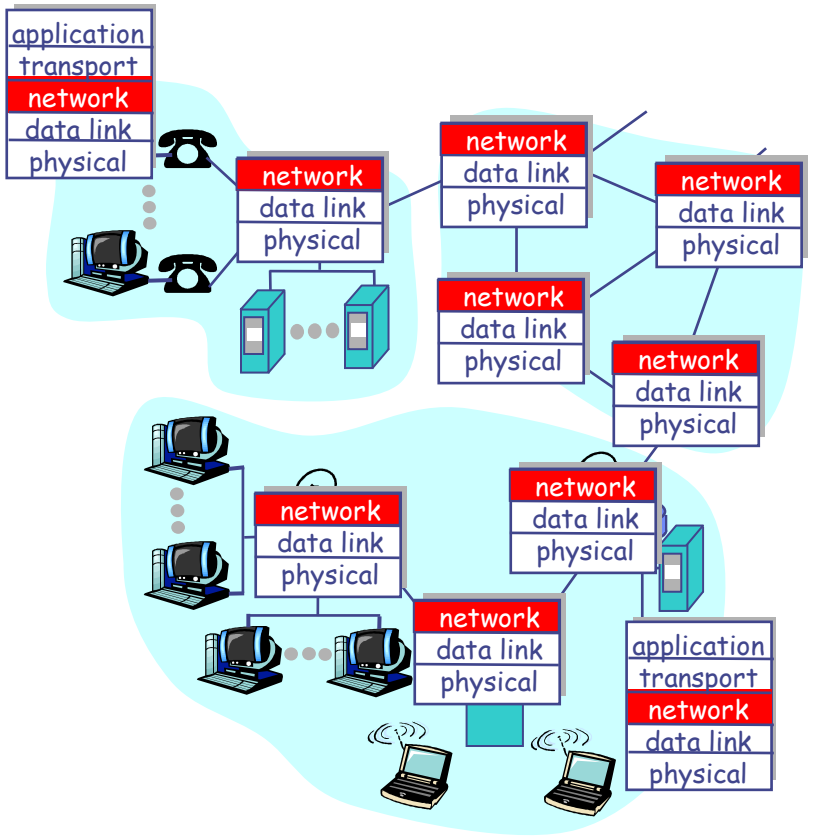
Il livello rete

Stack ISO/OSI



Le funzioni del livello rete

- Trasportare i pacchetti dall'host mittente a quello ricevente
- Implementare protocolli di livello rete in *tutti* i router e in *tutti* gli host



Le funzioni del livello rete

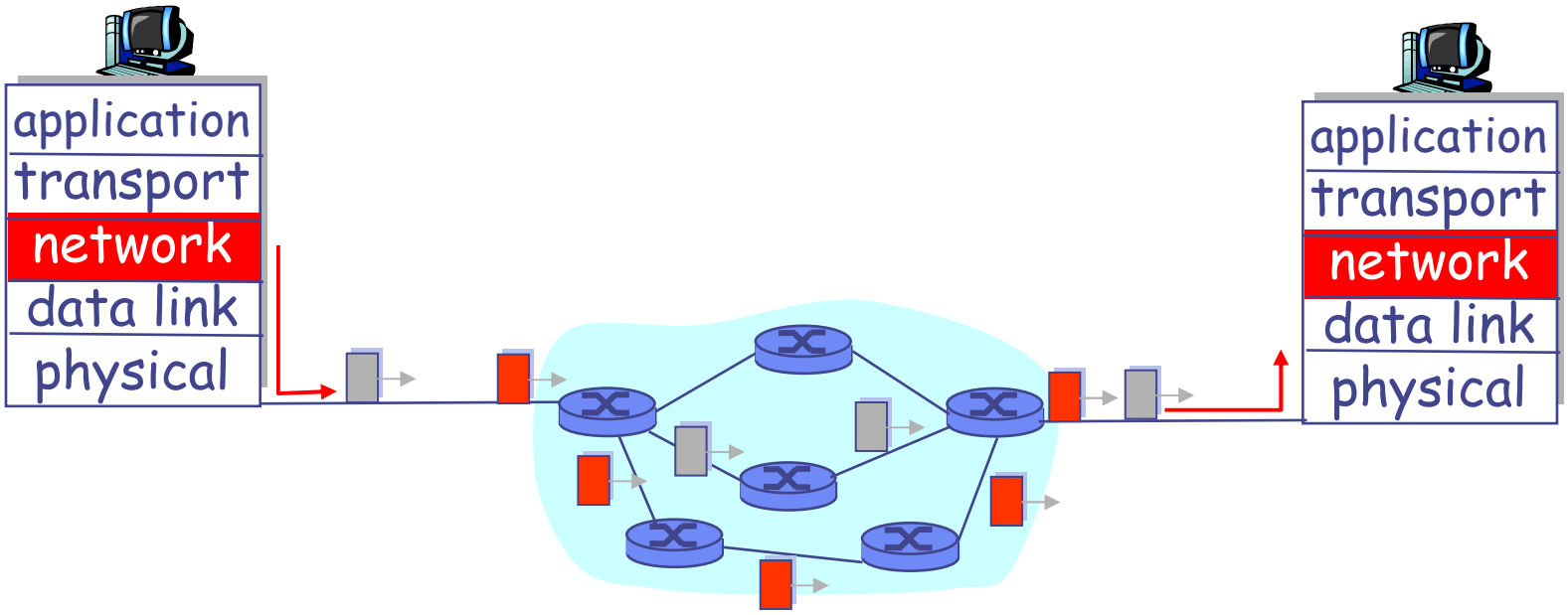
- Le reti possono essere classificate a seconda del metodo utilizzato per trasportare i pacchetti dalla sorgente alla destinazione:
  - Reti a datagrammi

    ogni pacchetto è instradato indipendentemente dagli altri pacchetti dello stesso flusso
  - Reti a circuiti virtuali

    viene precalcolato un percorso e tutti i pacchetti del flusso seguono questo percorso
- NB: parliamo comunque di reti a commutazione di pacchetto!

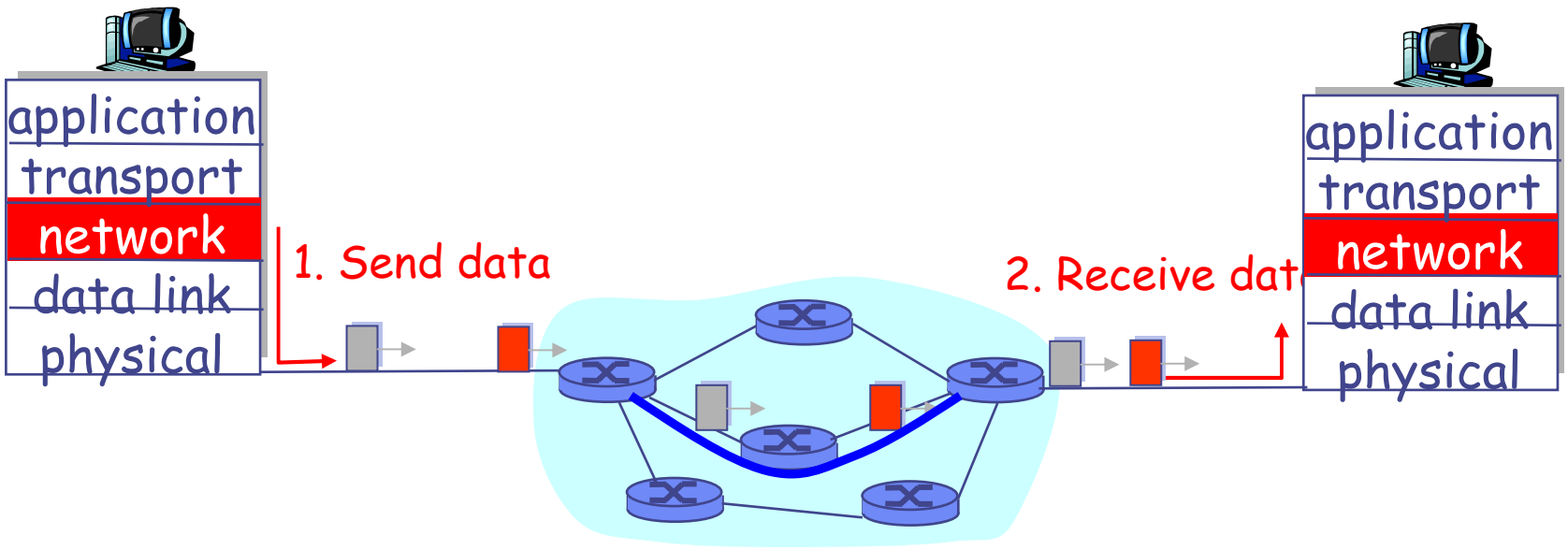
Reti a datagrammi

- Ogni router che riceve un pacchetto decide indipendentemente a chi mandarlo sulla base dell'indirizzo destinazione contenuto nel pacchetto
- Pacchetti tra la stessa coppia sorgente-destinazione possono seguire percorsi differenti



Reti a circuiti virtuali

- Ogni pacchetto contiene il numero del circuito virtuale
- Il circuito virtuale è stabilito prima della trasmissione dei dati
- I nodi devono conservare informazioni sui circuiti virtuali che li attraversano



IP (Internet Protocol)

- IP è un protocollo di livello rete usato per lo scambio di dati tra reti di calcolatori
- I dati sono trasportati con la tecnica dei datagrammi
- Offre un servizio di comunicazione *connection-less*
- Gestisce indirizzamento, frammentazione, riassemblaggio
- Costituisce la base sulla quale poggiano tutti gli altri protocolli, collettivamente noti come *TCP/IP suite*
  - TCP, UDP, ICMP
- È responsabile dell'instradamento dei pacchetti

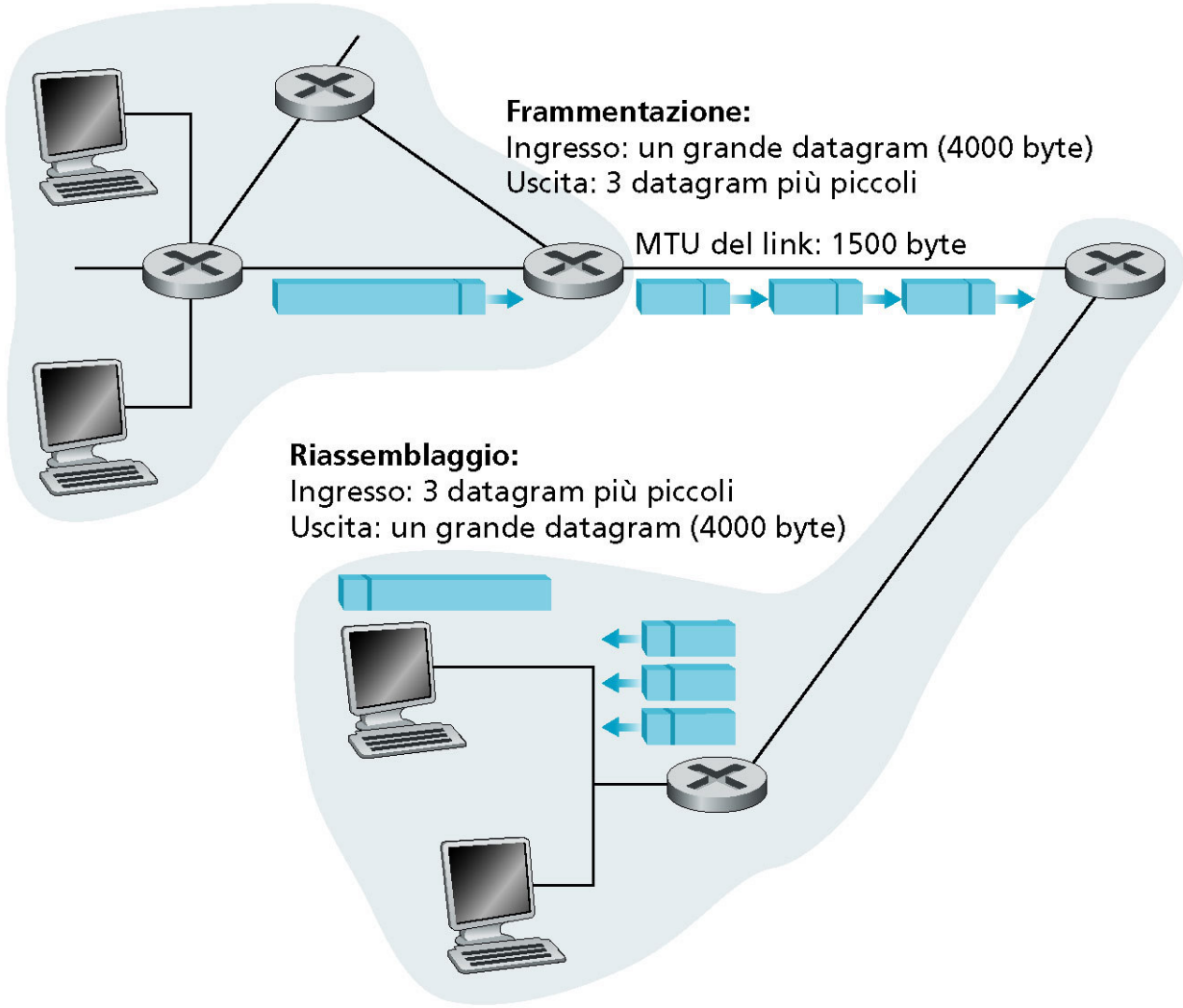
Il datagramma IP

- Un pacchetto IP è anche chiamato *datagramma*
- È costituito da un *header* e un'area dati
- I datagrammi possono avere dimensioni diverse
- La dimensione dell'header è solitamente fissata (20 byte) a meno che non siano presenti opzioni
- Un datagramma può contenere fino a un massimo di 65535 byte ($2^{16} - 1$)

L'header IP

- L'header contiene tutte le informazioni necessarie per la consegna del datagramma alla destinazione
  - Indirizzo destinazione
  - Indirizzo sorgente
  - Identificativo
  - Ed altro ancora…
- I router esaminano l'header di ogni datagramma e inoltrano il pacchetto lungo il percorso verso la destinazione
  - Usano tabelle di routing per calcolare il *next hop*
  - Aggiornano tali tabelle usando protocolli di routing dinamici

Frammentazione e riassemblaggio IP



IP è consegna best effort

- IP *non* garantisce di prevenire:
  - Datagrammi duplicati
  - Consegna ritardata o fuori ordine
  - Corruzione di dati
  - Perdita di pacchetti
- La consegna affidabile dei pacchetti può avvenire grazie a meccanismi di controllo da parte di protocolli di livello superiore

Indirizzi IP

- Ad ogni host è assegnato un indirizzo IP o indirizzo Internet
  - È un numero di 32 bit = 4 byte
  - Unico in tutta Internet
- Ogni indirizzo IP è diviso in un prefisso e un suffisso
  - Il prefisso indica la rete alla quale l'host è collegato
    - Due reti non possono avere lo stesso numero di rete
  - Il suffisso identifica l'host all'interno della rete
    - Due host sulla stessa rete non possono avere lo stesso suffisso, ma host su reti diverse possono avere lo stesso suffisso

Chi assegna gli indirizzi IP?

- ICANN:
  - Internet Corporation for Assigned Names and Numbers
- Assegna gli indirizzi
- Gestisce il DNS
- Assegna i nomi dei domini
- Risolve eventuali dispute (conflitti di nomi e/o indirizzi)

Notazione *dotted decimal*

- La notazione dotted decimal rappresenta gli indirizzi IP come 4 numeri decimali separati da punto
- Ogni numero decimale, in quanto rappresenta un byte, è compreso tra 0 e 255

32-bit Binary Number	Equivalent Dotted Decimal
10000001 00110100 00000110 00000000	129.52.6.0
11000000 00000101 00110000 00000011	192.5.48.3
00001010 00000010 00000000 00100101	10.2.0.37
10000000 00001010 00000010 00000011	128.10.2.3
10000000 10000000 11111111 00000000	128.128.255.0

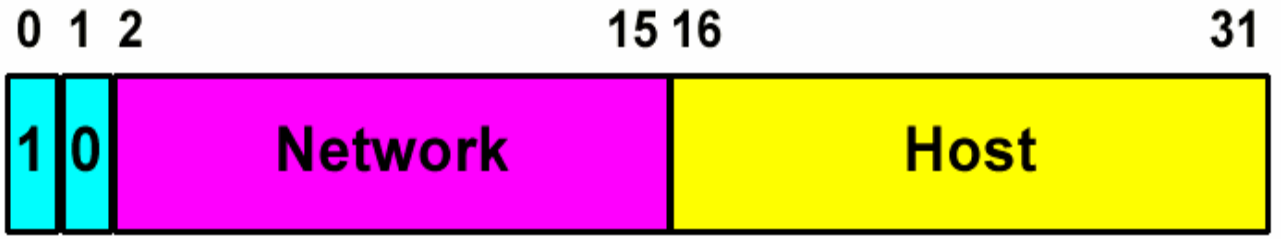
Classi di indirizzi

- La parte di indirizzo che specifica la rete e quella che specifica l'host non hanno lunghezza fissa, ma variano a seconda della *classe* a cui appartiene l'indirizzo
- Sono state definite 5 classi:
  - 3 (A, B, C) sono usate per gli indirizzi degli host e si differenziano per la lunghezza della parte rete/host
  - 1 (D) è usata per il *multicast*
  - 1 (E) è riservata per usi futuri

Indirizzi di classe A

- Campo rete
  - 7 bit
  - Massimo 128 reti
  - Il primo byte è compreso tra 0 e 127
- Campo host
  - 24 bit
  - Massimo $2^{24} \approx$ 16M host

Indirizzi di classe B



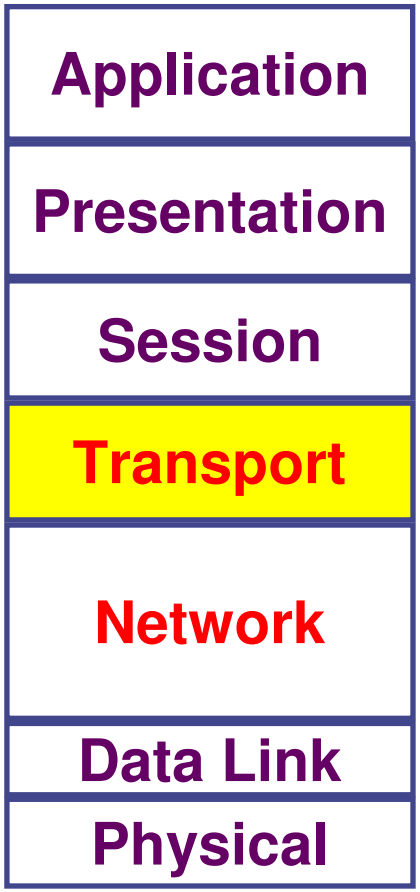
- Campo rete
  - 14 bit
  - Massimo 16k reti
  - Il primo byte è compreso tra 128 e 191
- Campo host
  - 16 bit
  - Massimo $2^{16} \approx$ 64k host

Indirizzi di classe C

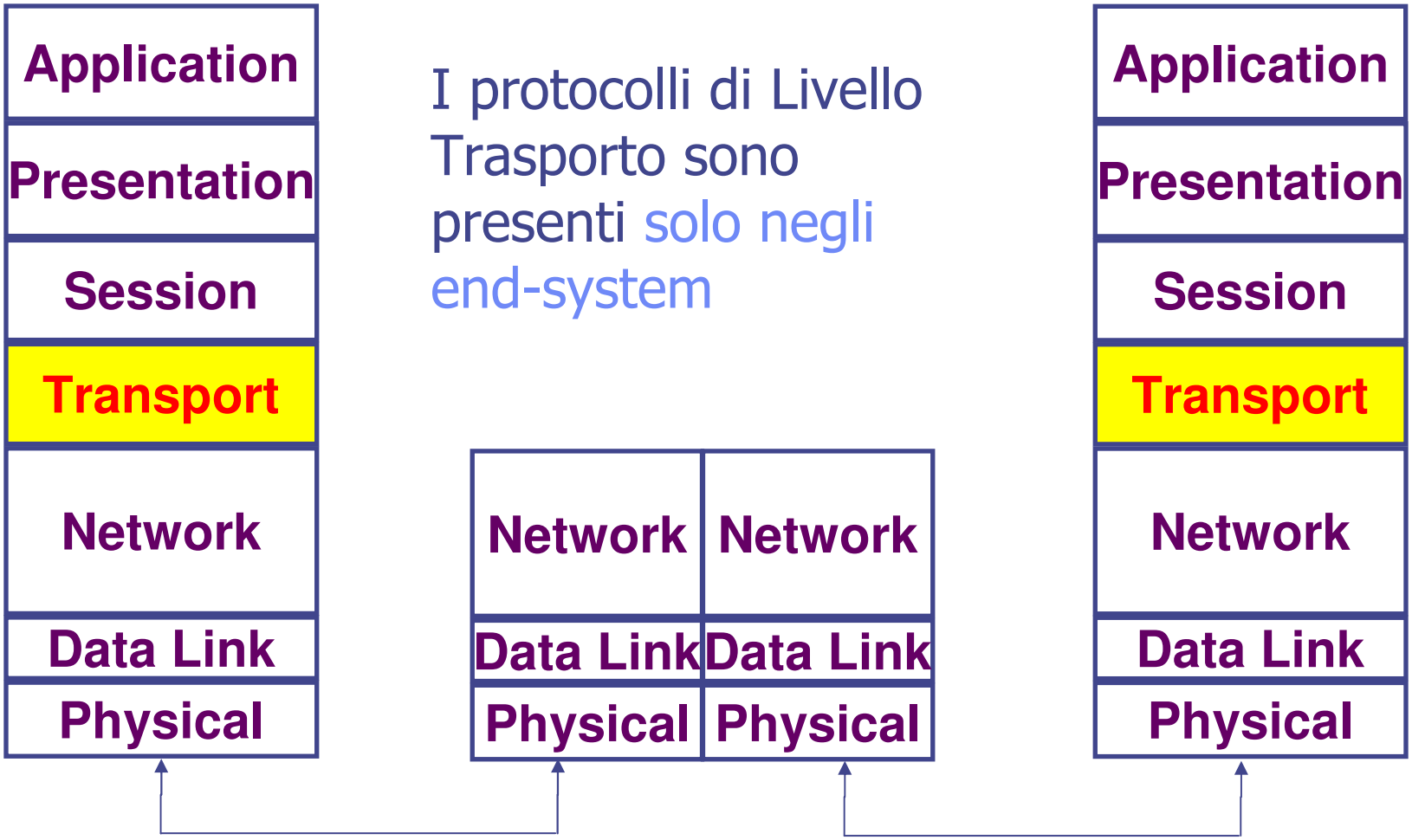
- Campo rete
  - 21 bit
  - Massimo 2M reti
  - Il primo byte è compreso tra 192 e 223
- Campo host
  - 8 bit
  - Massimo 256 host

Livello Trasporto

Stack ISO/OSI



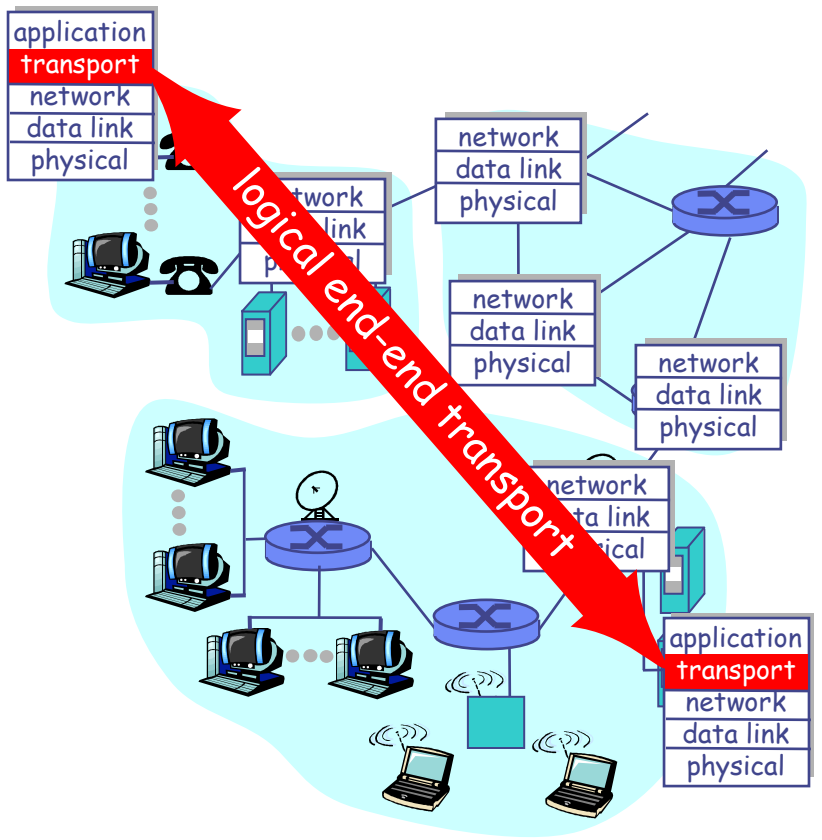
Livello Trasporto



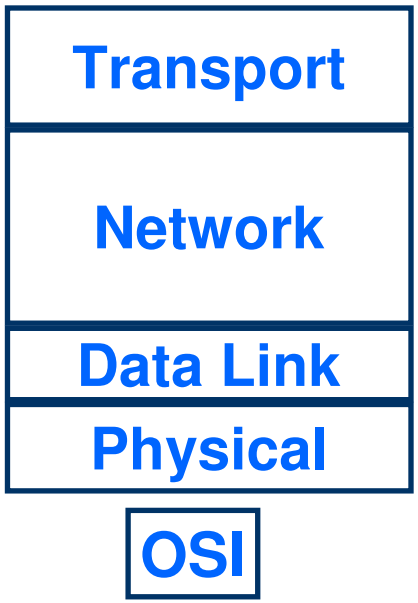
I protocolli di Livello Trasporto sono presenti solo negli end-system

Servizi e Protocolli del Livello Trasporto

- Offre un canale di comunicazione logica tra applicazioni attive su differenti host
- Differenze tra livello trasporto e livello rete:
  - Network-layer: trasferimento dati tra end-system
  - Transport-layer: trasferimento dati tra processi. Naturalmente necessita dei servizi offerti dal livello rete.



Servizi del Livello Trasporto - 1



- affidabile
- inaffidabile
- affidabile
- inaffidabile

Il livello rete offre un servizio inaffidabile, quindi:

Il livello trasporto deve rimediare:

- aumentare l'efficienza
- aumentare l'affidabilità

In particolare:

- controllo degli errori
- sequenza ordinata
- controllo di flusso
- controllo di congestione

Servizi del Livello Trasporto - 2

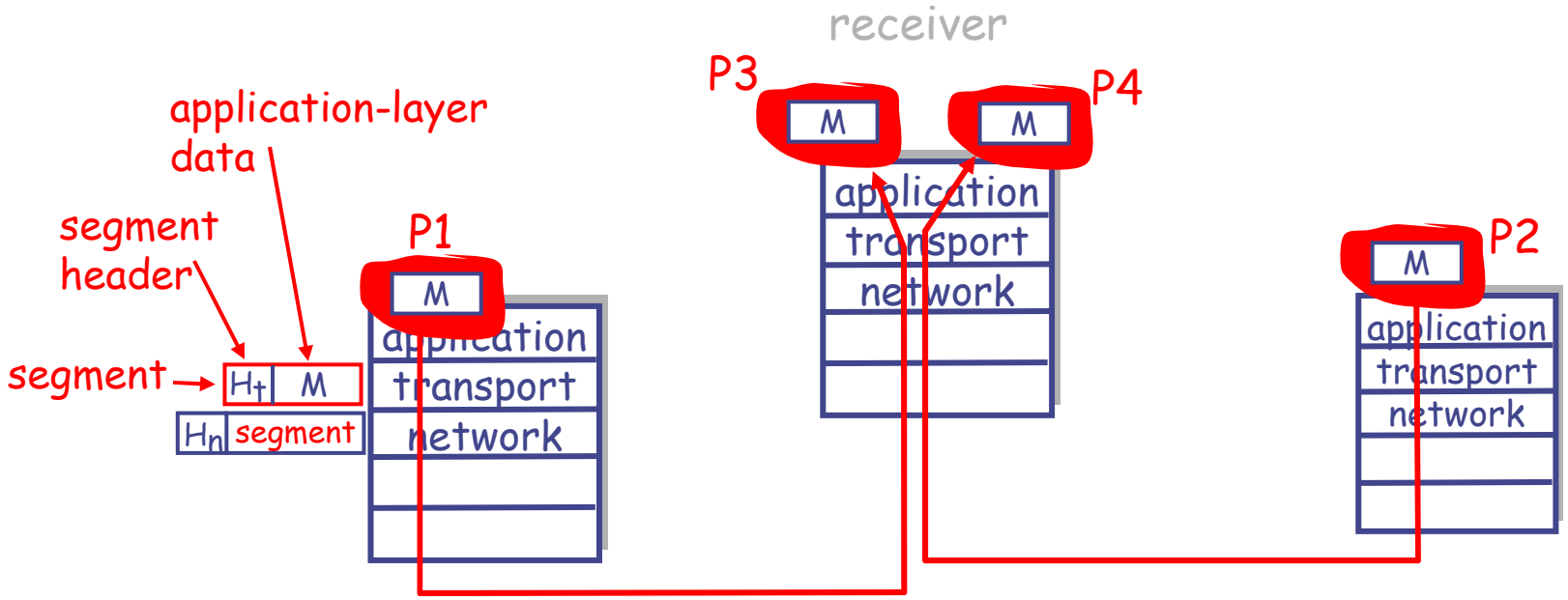
- I protocolli di Livello Trasporto sono realizzati al di sopra del Livello Rete, quindi è necessario gestire:
  - apertura della connessione (setup)
  - memorizzazione dei pacchetti all'interno della rete
  - un numero elevato di connessioni …
    - Multiplexing e Demultiplexing

Multiplexing e Demultiplexing - 1

segment – dati che sono scambiati tra processi a livello trasporto

Demultiplexing: inoltrare i segment ricevuti al corretto processo cui i dati sono destinati

TPDU: transport protocol data unit

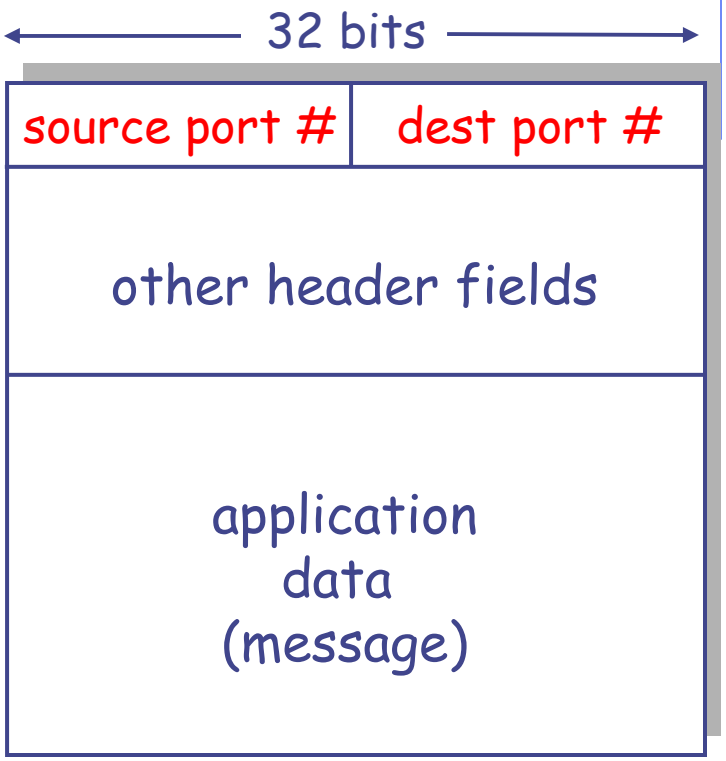


Multiplexing e Demultiplexing - 2

Multiplexing:
Raccogliere i dati provenienti dalle applicazioni, imbustare i dati con un header appropriato (per il de-multiplexing)

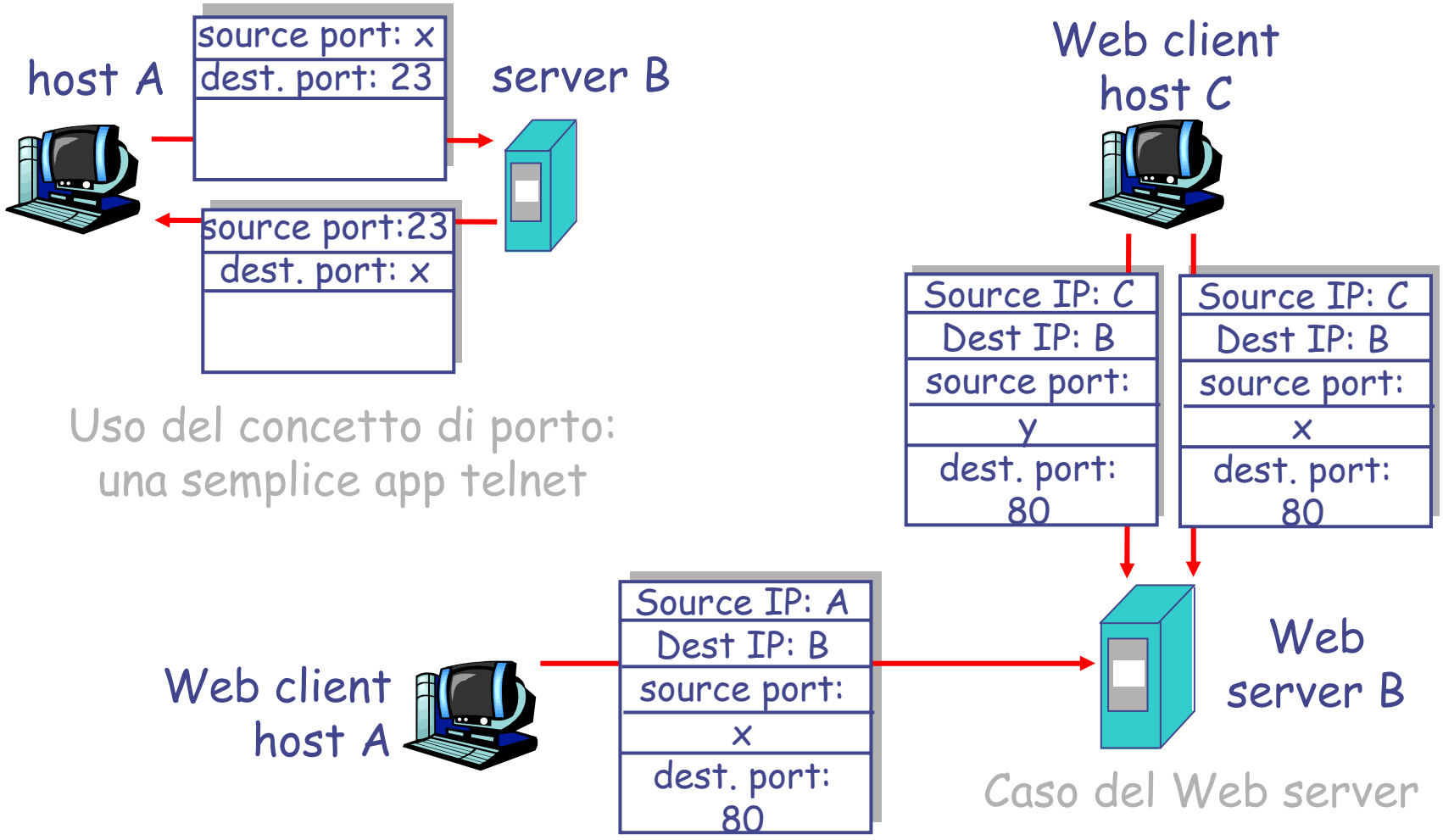
multiplexing/demultiplexing:

- Realizzato attraverso la coppia <indirizzo IP, numero di porto>
  - source, dest port # è presente in ogni segmento
  - numeri di porto "well-known" per applicazioni particolari

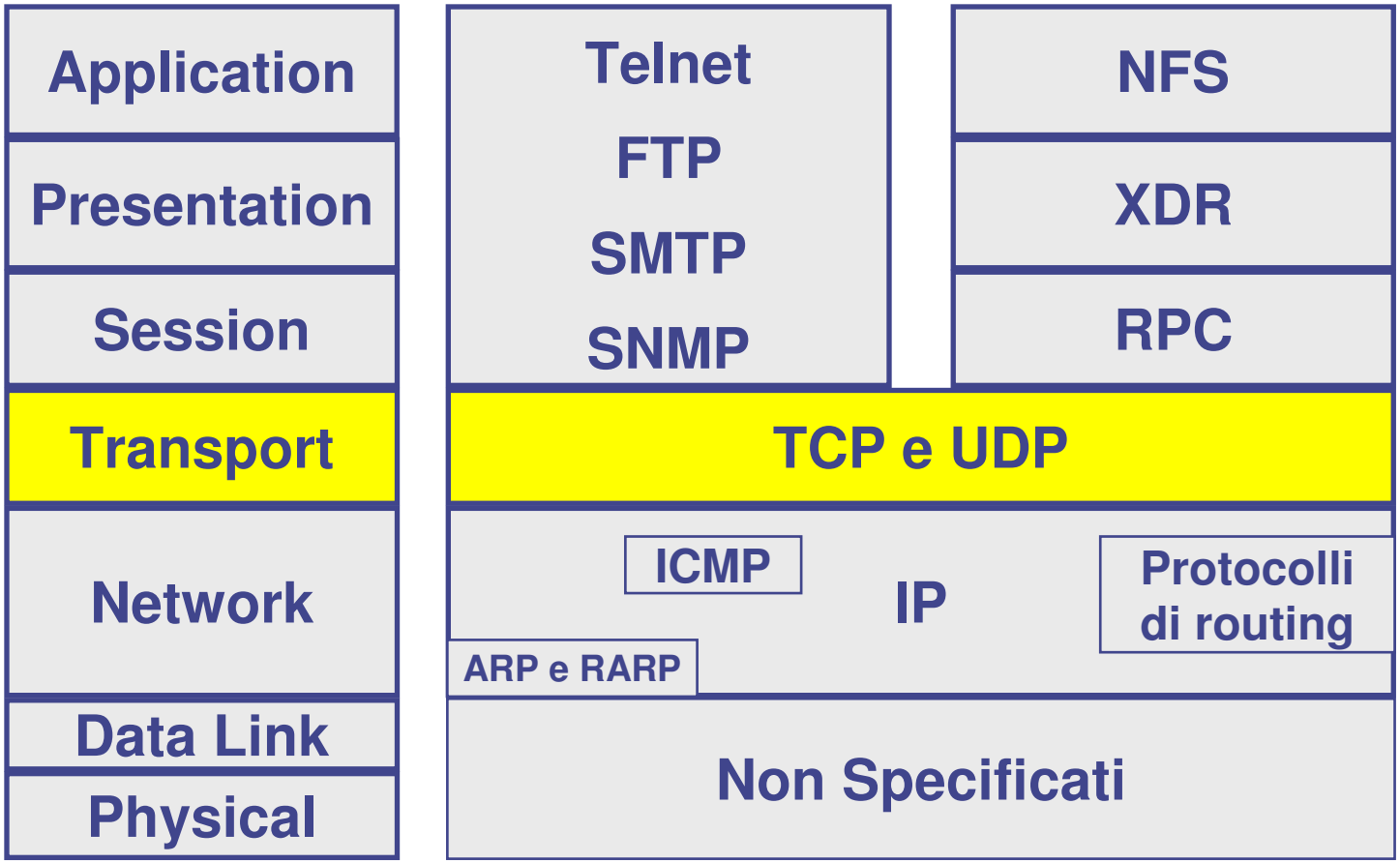


TCP/UDP segment format

Multiplexing e Demultiplexing: esempi



I protocolli TCP e UDP



UDP: User Datagram Protocol

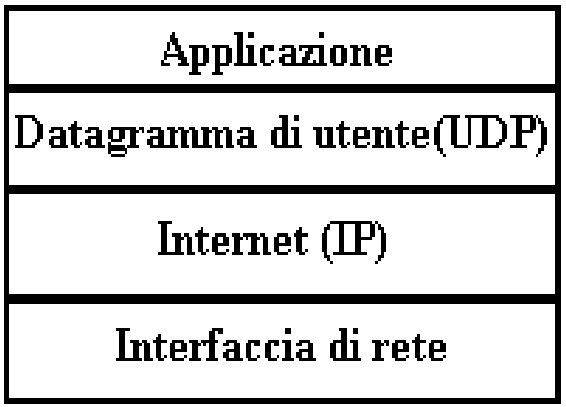
- Aggiunge poco ad IP:
  - servizio "best effort":
    - i pacchetti UDP possono:
      - subire perdite
      - giungere a destinazione in ritardo, (o non arrivare affatto)
      - giungere a destinazione non ordinati
  - *servizio connectionless:*
    - non è prevista una fase di inizializzazione
    - ogni segmento UDP è inviato indipendentemente dagli altri

UDP: User Datagram Protocol - 2

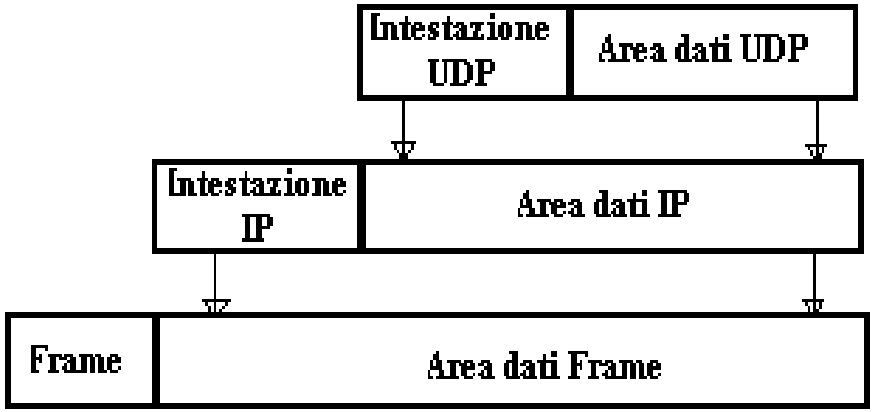
- Perché è stato introdotto UDP?
  - non è necessaria la fase di inizializzazione (setup) che introduce delay
    - Es: DNS è basato su UDP
  - semplice: sender e receiver non devono conservare informazioni di stato
  - intestazione di dimensioni contenute:
    - basso overhead
  - controllo della congestione assente:
    - le applicazioni possono inviare alla velocità desiderata
      - utile per alcune applicazioni
      - rischioso per la rete

Incapsulamento di segmenti UDP

Stratificazione e Incapsulamento



Stratificazione concettuale



Incapsulamento

TCP: Transmission Control Protocol

- End-to-end:
  - Una connessione unica tra mittente e ricevente
- Senza errori, sequenza ordinata.
- Buffers su mittente e ricevente
- controllo di congestione
- full duplex data:
  - Flusso di dati bi-direzionale all'interno della stessa connessione
  - MSS: maximum segment size
- connection-oriented:
  - handshaking prepara mittente e ricevente prima della comunicazione
- controllo di flusso:
  - Il mittente non invia più di quanto il ricevente non possa accettare