

Reti di Elaboratori

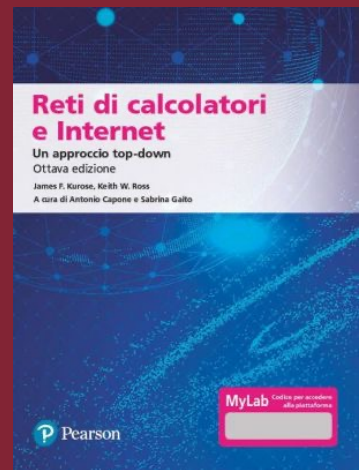
Livello di Trasporto, Multiplexing e Demultiplexing



SAPIENZA
UNIVERSITÀ DI ROMA

Alessandro Checco

alessandro.checco@uniroma1.it



Capitolo 3

Livello di trasporto: panoramica

Obiettivi:

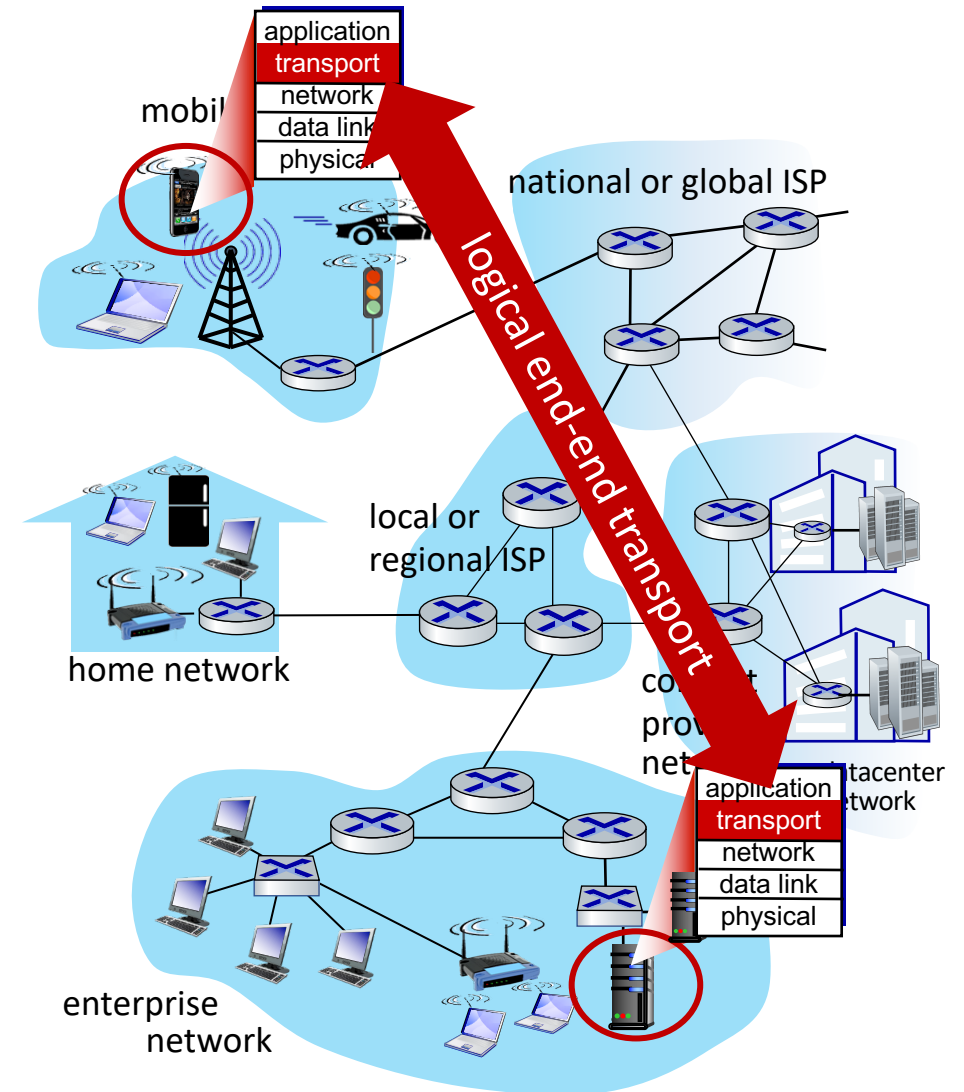
- comprendere i principi alla base dei servizi del livello di trasporto:
 - multiplexing, demultiplexing
 - trasferimento affidabile dei dati
 - controllo del flusso
 - controllo della congestione
- informazioni sui protocolli del livello di trasporto Internet:
 - UDP: trasporto senza connessione
 - TCP: trasporto affidabile orientato alla connessione
 - Controllo della congestione TCP

Livello di trasporto: sommario

- Servizi a livello di trasporto
- Multiplexing e demultiplexing
- Trasporto senza connessione: UDP
- Principi di trasferimento affidabile dei dati
- Trasporto orientato alla connessione: TCP
- Principi di controllo della congestione
- Controllo della congestione TCP
- Evoluzione della funzionalità del livello di trasporto

Servizi e protocolli di trasporto

- forniscono *comunicazione logica* tra processi applicativi in esecuzione su host diversi
- azioni dei protocolli di trasporto negli end systems:
 - mittente: suddivide i messaggi dell'applicazione in *segmenti* e li passa al livello di rete
 - ricevitore: riasmonta i segmenti in messaggi e li passa al livello dell'applicazione
- due protocolli di trasporto disponibili per le applicazioni Internet
 - TCP, UDP



Esempio

Analogia con la posta ordinaria:

1 persona di un condominio inviano una lettera a 1 persona di un altro condominio consegnandola/ricevendola a/da un portiere

- ☐ processi = persone
- ☐ messaggi delle applicazioni = lettere nelle buste
- ☐ host = condomini
- ☐ protocollo di trasporto = portieri dei condomini
- ☐ protocollo del livello di rete = servizio postale

N.B. i portieri svolgono il proprio lavoro localmente, non sono coinvolti nelle tappe intermedie delle lettere (così come il protocollo di trasporto)

servizi e protocolli a livello di rete vs. trasporto

- livello di rete:
comunicazione logica tra *host*
- livello di trasporto :
comunicazione logica tra *processi*
 - si basa su, e migliora, i servizi a livello di rete

analogia:

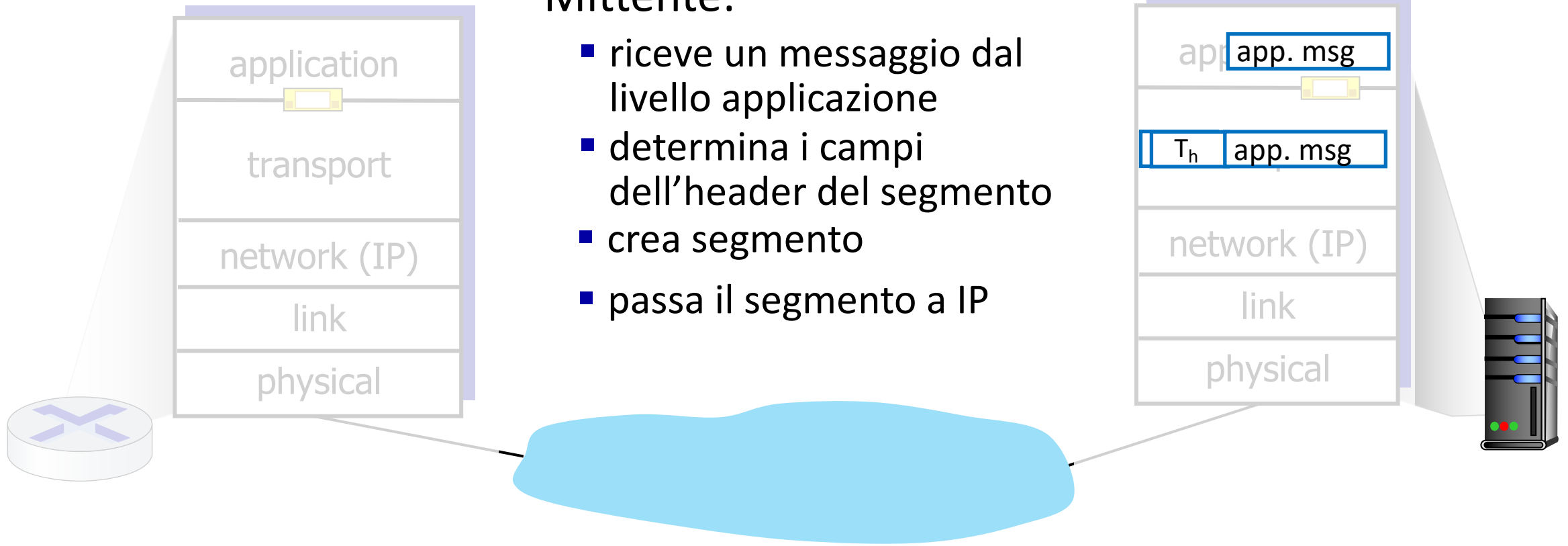
N lettere dal condominio A a N persone nel condominio B:

- host = condomini
- processi = persone
- app messaggi = lettere in buste
- protocollo di trasporto = portieri
- protocollo a livello di rete = servizio postale

Azioni del livello di trasporto

Mittente:

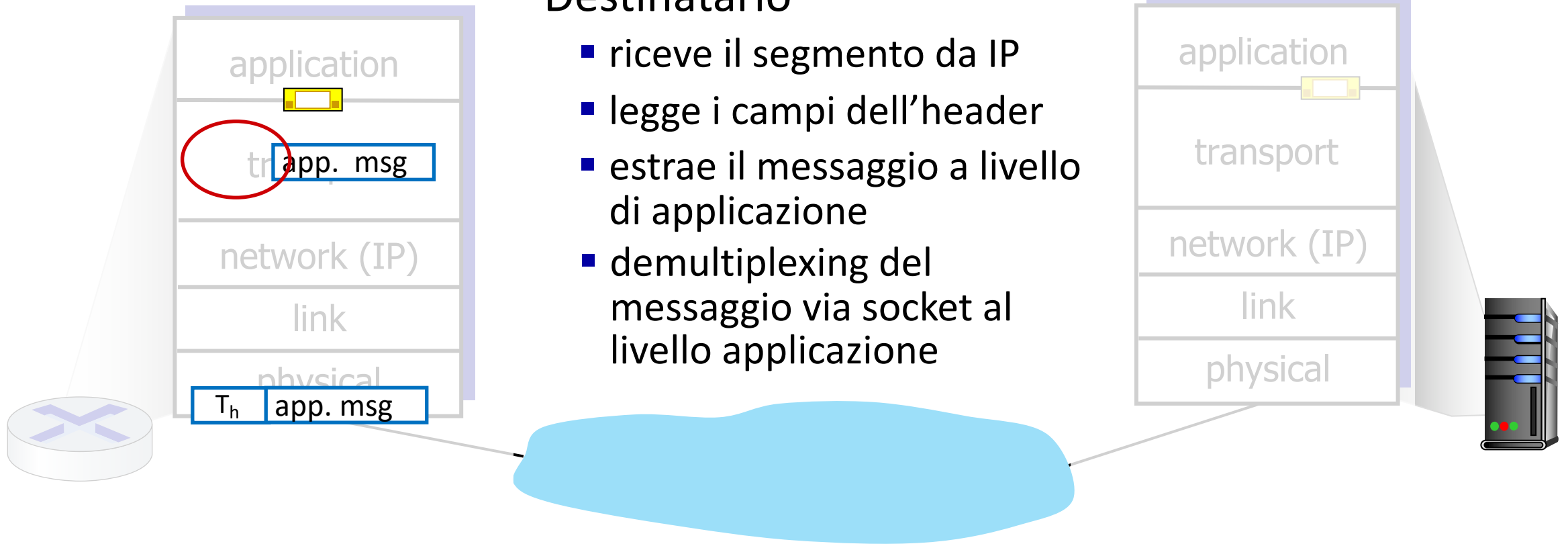
- riceve un messaggio dal livello applicazione
- determina i campi dell'header del segmento
- crea segmento
- passa il segmento a IP



Azioni del livello di trasporto

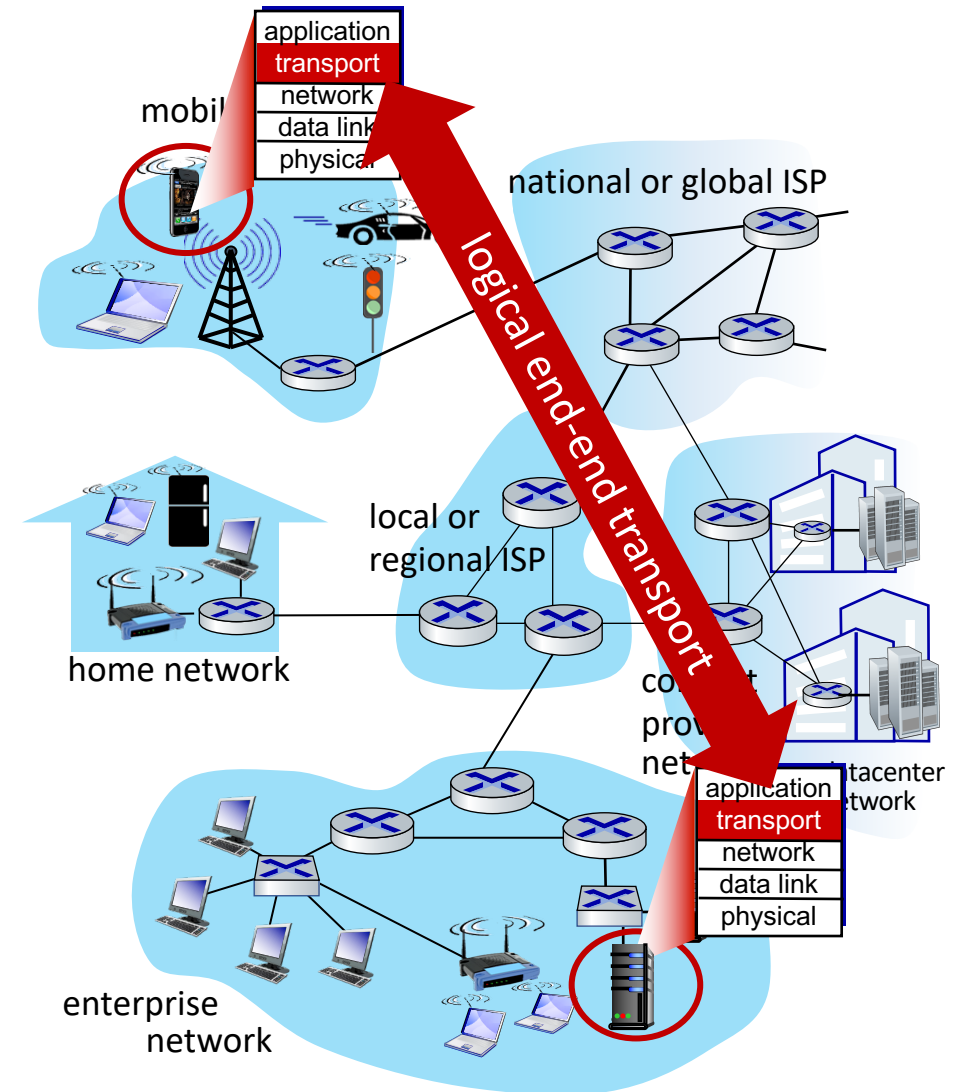
Destinatario

- riceve il segmento da IP
- legge i campi dell'header
- estrae il messaggio a livello di applicazione
- demultiplexing del messaggio via socket al livello applicazione



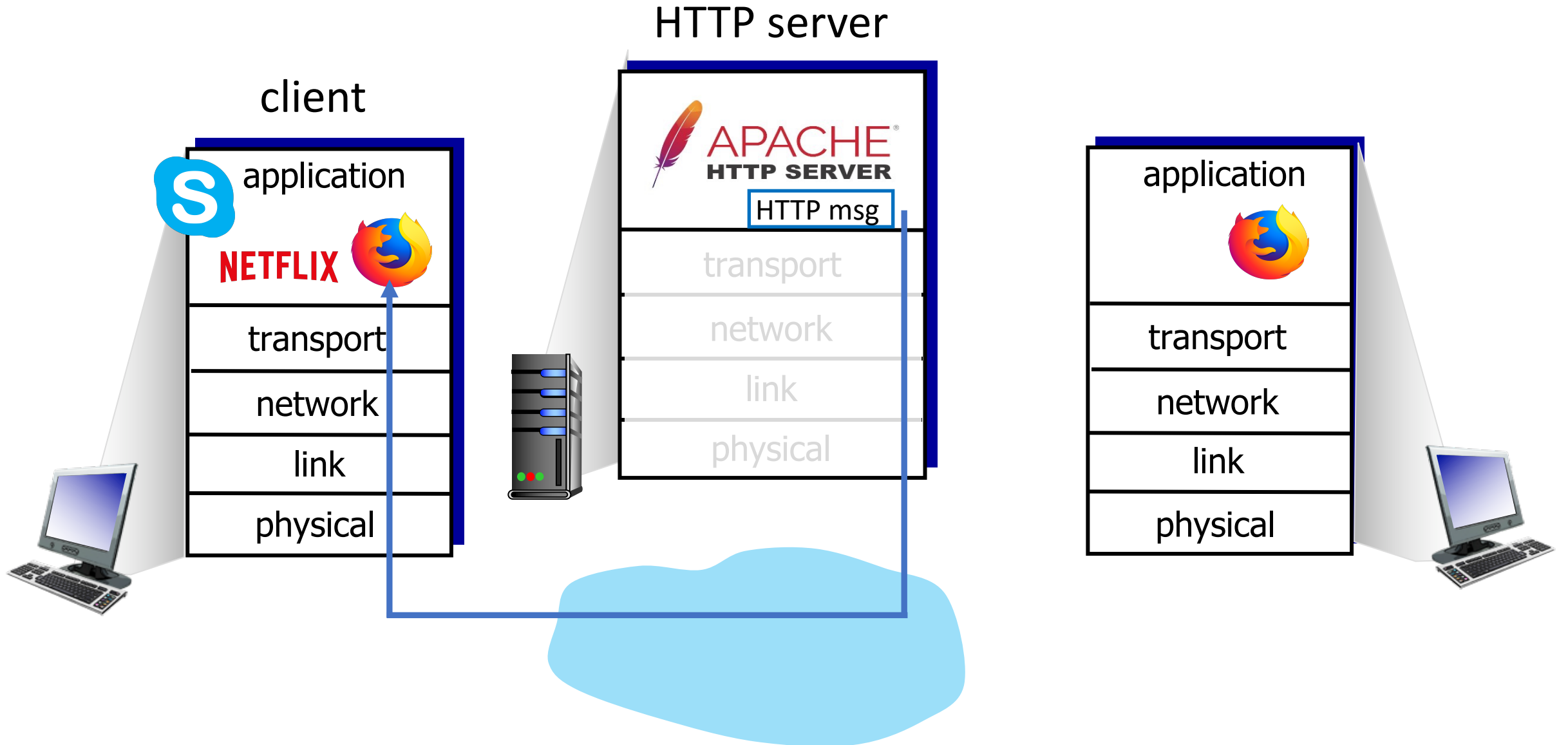
Due principali protocolli di trasporto Internet

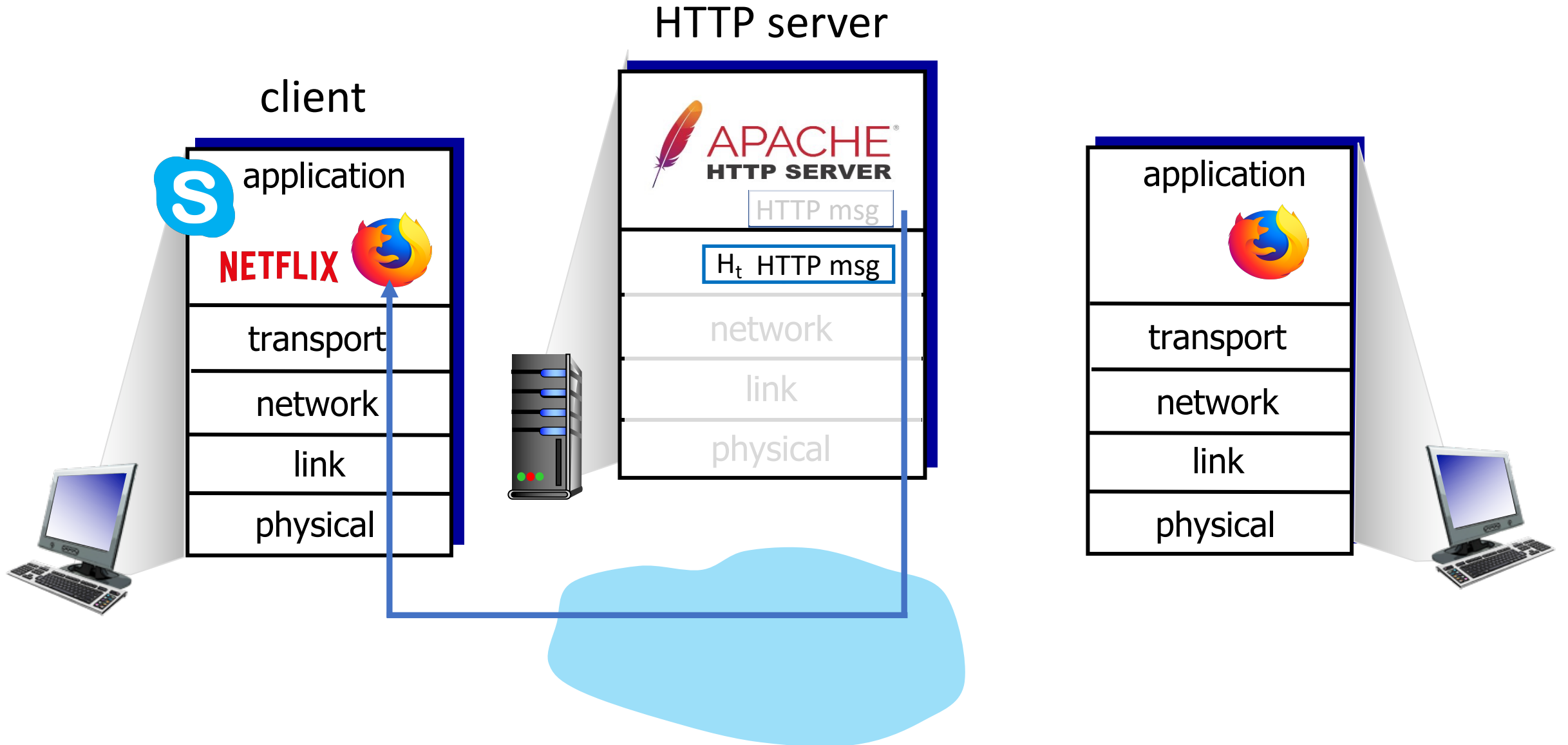
- **TCP:** Transmission Control Protocol
 - consegna affidabile e in ordine
 - controllo della congestione
 - controllo del flusso
 - stabilimento di connessione
- **UDP:** User Datagram Protocol
 - consegna inaffidabile e non ordinata
 - estensione senza fronzoli dell'IP "best-effort"
- servizi non disponibili:
 - garanzie di ritardo
 - garanzie di larghezza di banda

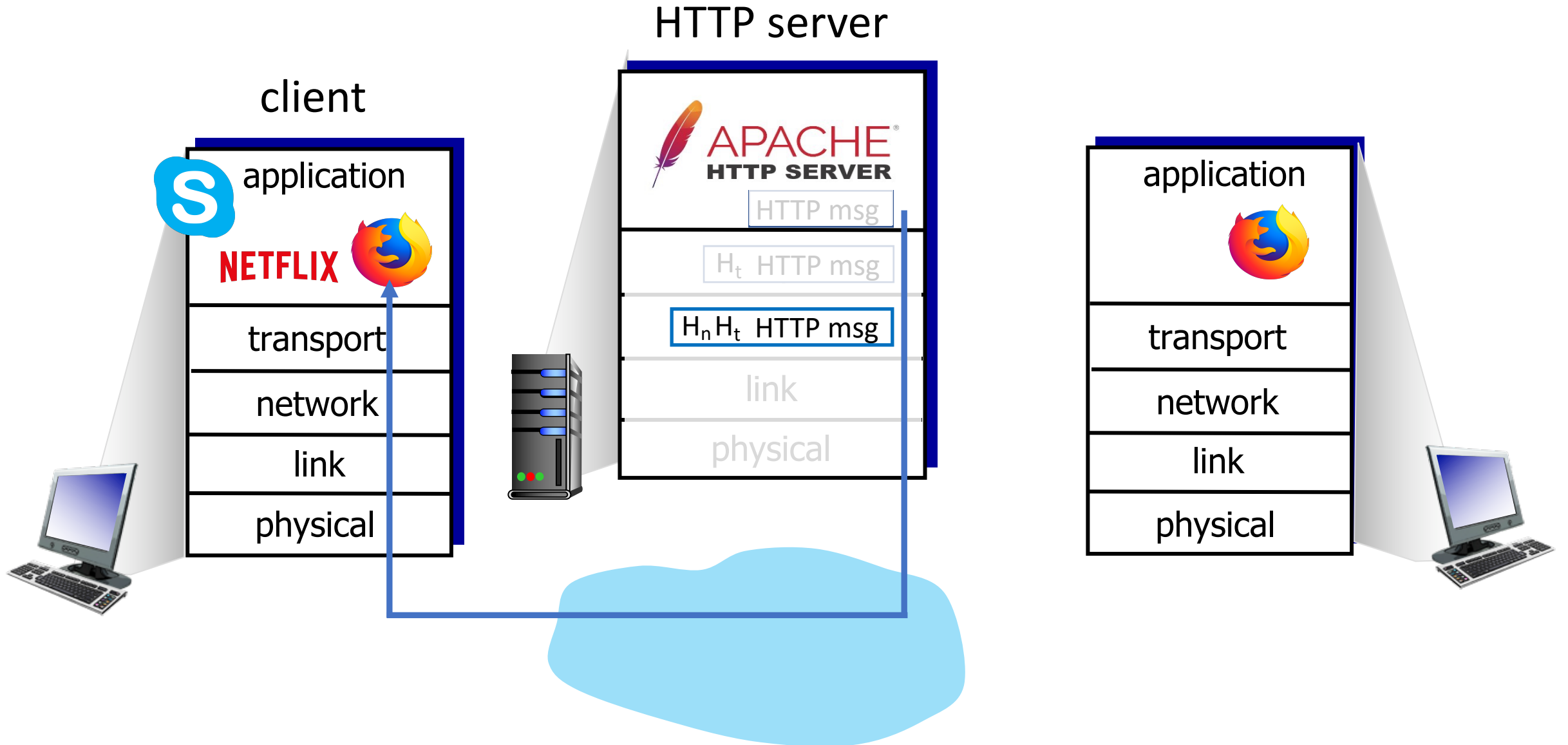


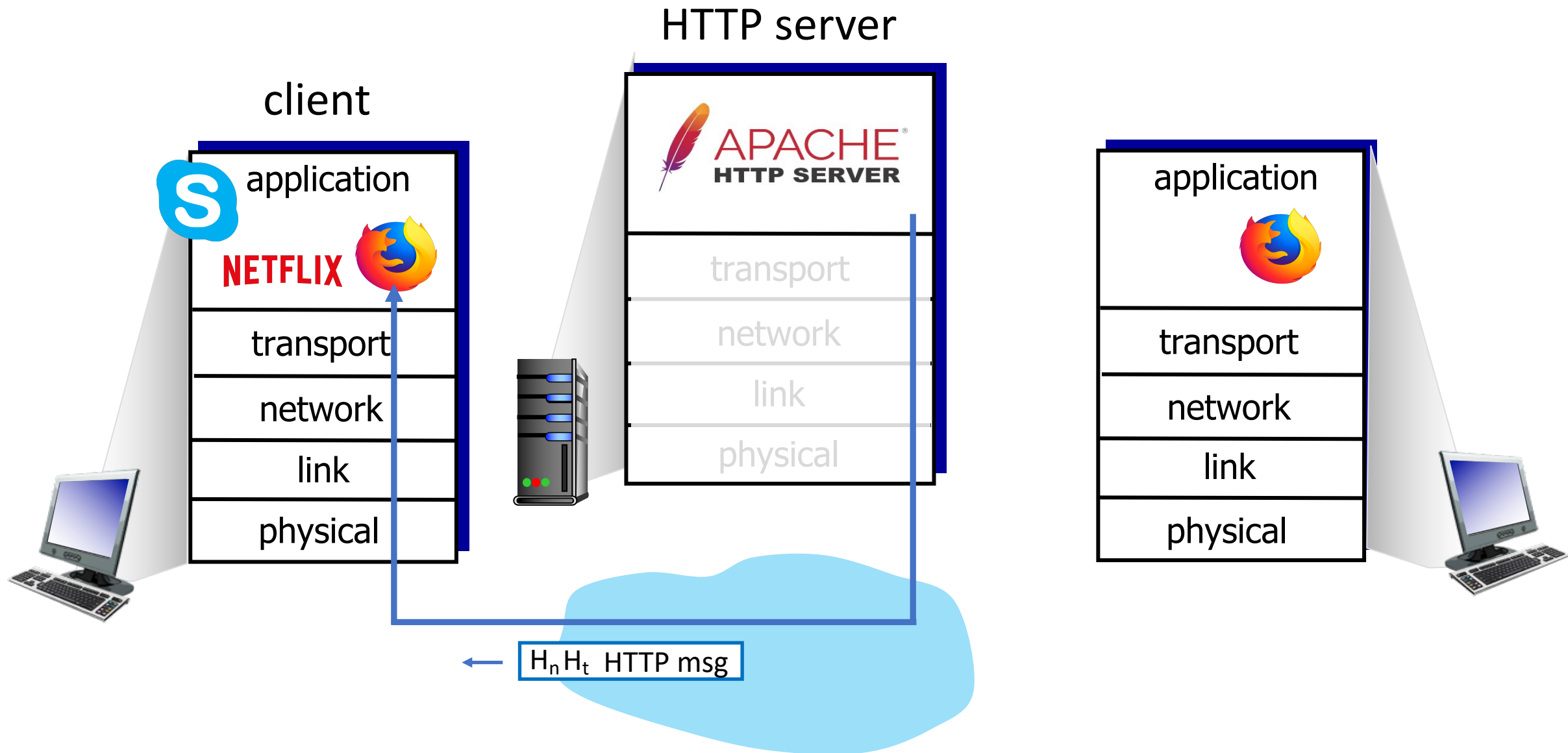
Livello di trasporto: sommario

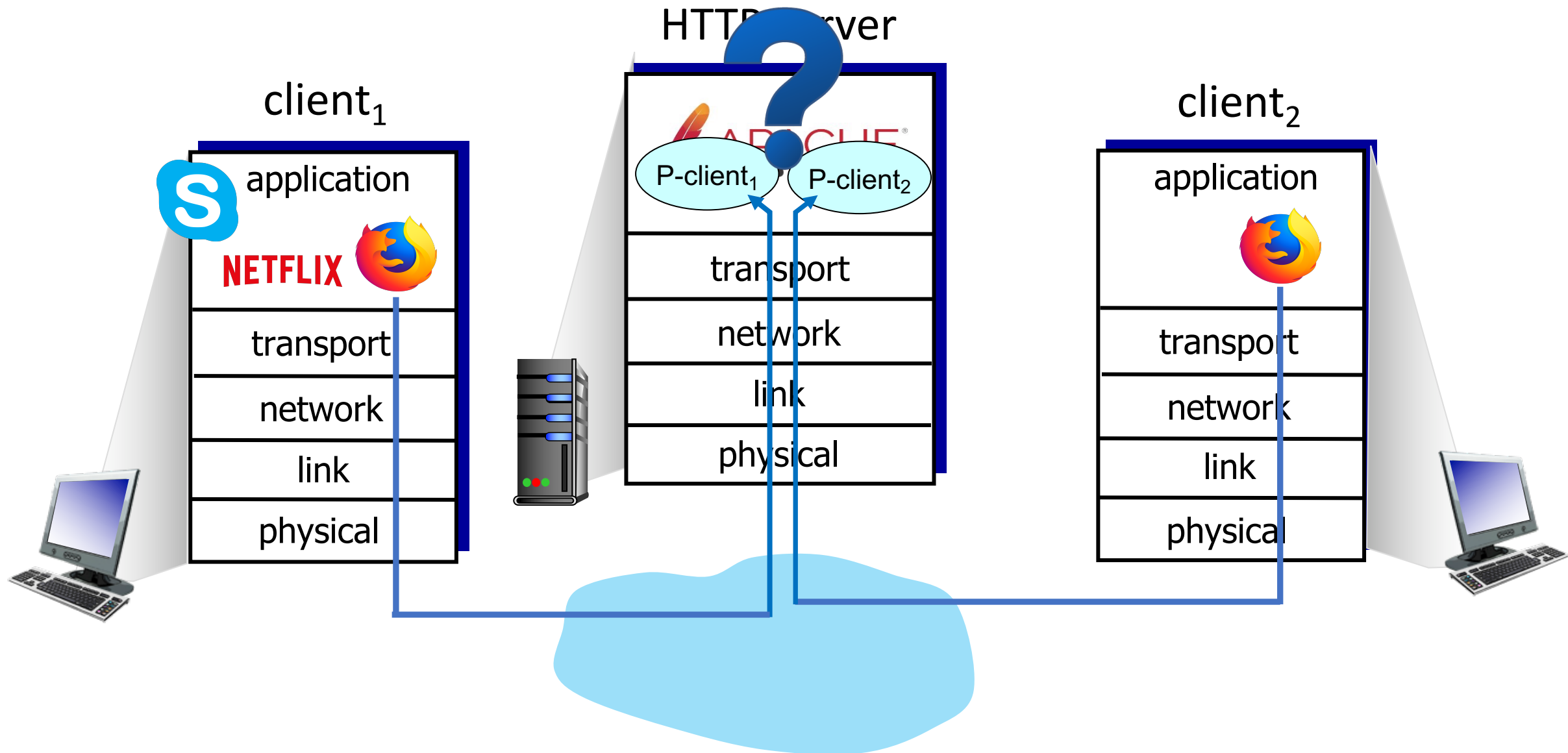
- Servizi a livello di trasporto
- Multiplexing e demultiplexing
- Trasporto senza connessione: UDP
- Principi di trasferimento affidabile dei dati
- Trasporto orientato alla connessione: TCP
- Principi di controllo della congestione
- Controllo della congestione TCP
- Evoluzione della funzionalità del livello di trasporto











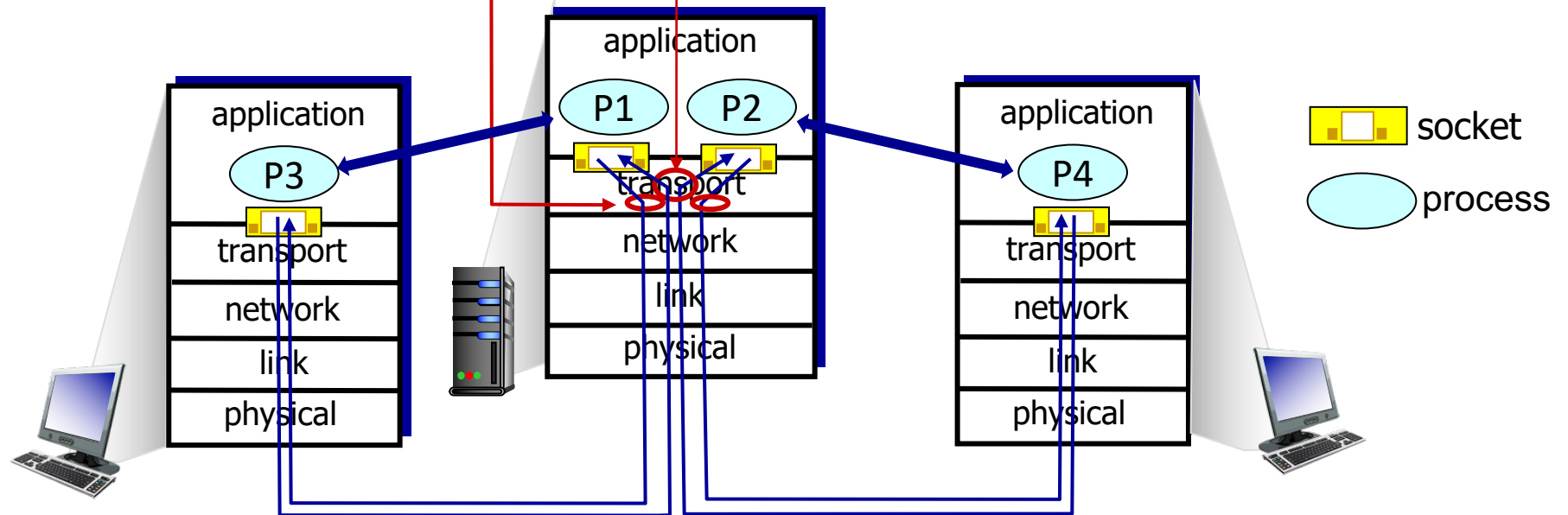
Multiplexing/demultiplexing

multiplexing al mittente

gestisce i dati da più socket (P1 e P2), aggiunge un header di trasporto (successivamente utilizzato per il demultiplexing)

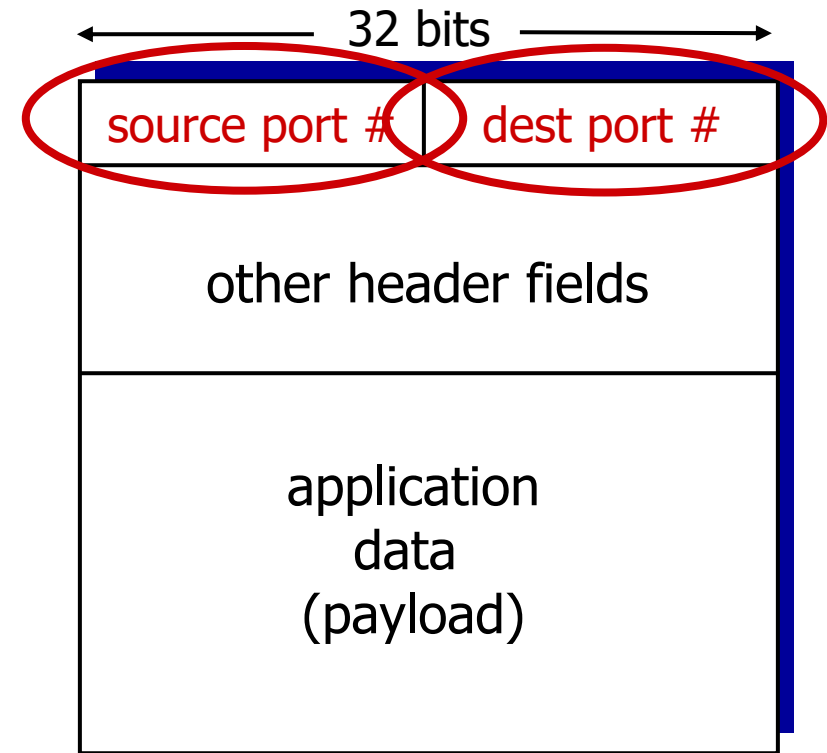
demultiplexing al destinatario

usa le informazioni nell'header per consegnare i segment ricevuti al socket corretto (P1 o P2)



Come funziona il demultiplexing

- l'host riceve i datagrammi IP
 - ogni datagramma ha indirizzo IP di origine, indirizzo IP di destinazione
 - ogni datagramma trasporta un segmento del livello di trasporto
 - ogni segmento ha numero di porta per origine e destinazione
- host utilizza *indirizzi IP e numeri di porta* per indirizzare il segmento al socket appropriato



TCP/UDP segment format

Demultiplexing senza connessione (UDP)

- durante la creazione del socket, è necessario specificare la porta *locale dell'host*:

```
DatagramSocket mySocket1  
= new DatagramSocket(12534);
```

- durante la creazione del datagramma da inviare al socket UDP, è necessario specificare
 - indirizzo IP di destinazione
 - porta di destinazione (ad es. 23)

quando l'host destinatario riceve il segmento *UDP*:

- controlla la porta di destinazione nel segmento
- indirizza il segmento UDP al socket con quel # di porta (demultiplexing)



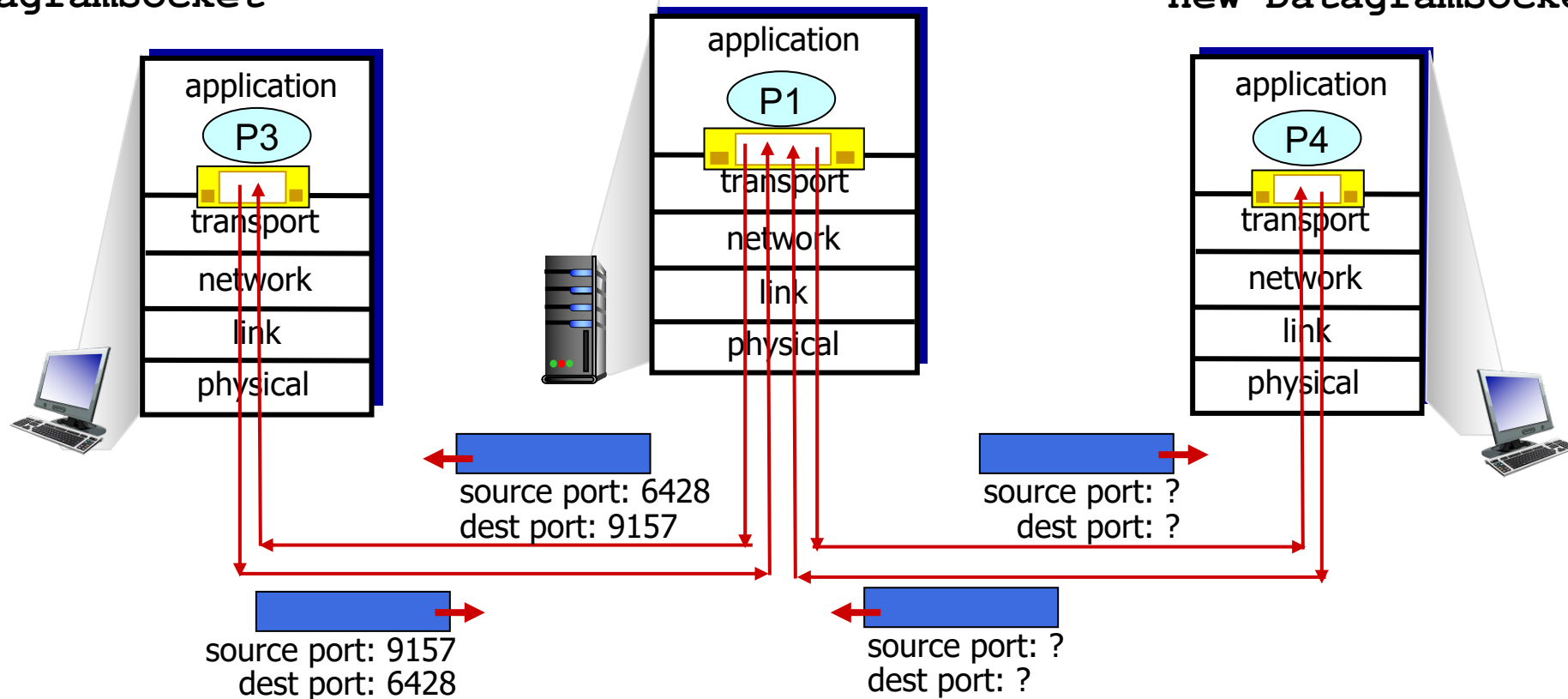
Datagrammi IP/UDP con la *stessa porta di dest.*, ma diversi indirizzi IP di origine e/o numeri di porta di origine verranno indirizzati allo *stesso socket* dell'host ricevente

Demultiplexing senza connessione: un esempio

```
DatagramSocket mySocket2 =  
new DatagramSocket  
(9157);
```

```
DatagramSocket  
serverSocket = new  
DatagramSocket  
(6428);
```

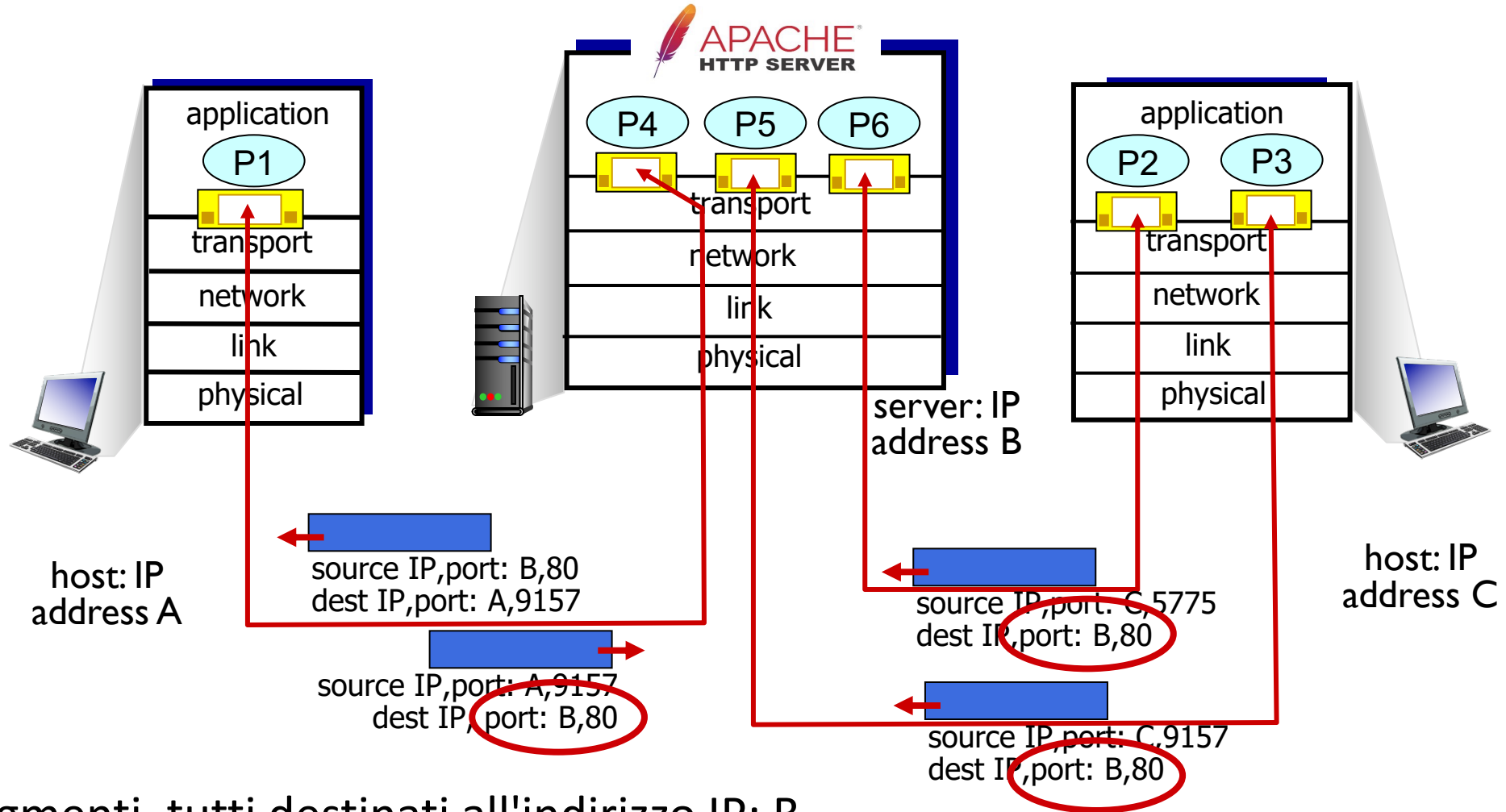
```
DatagramSocket mySocket1 =  
new DatagramSocket (5775);
```



Demultiplexing orientato alla connessione

- Socket TCP identificato da una **4-tuple**:
 - indirizzo IP di origine
 - numero di porta di origine
 - indirizzo IP di destinazione
 - numero di porta di destinazione
- demux: il ricevitore utilizza *tutti e quattro i valori* per indirizzare il segmento al socket appropriato
- il server può avere più socket TCP simultanei:
 - ogni socket identificato dalla tupla di 4 elementi
 - ogni socket associato a una diversa connessione TCP

Demultiplexing orientato alla connessione: esempio



Tre segmenti, tutti destinati all'indirizzo IP: B,
porta dest: 80 sono demultiplati in socket differenti

Riepilogo

- Multiplexing, demultiplexing: in base ai valori del campo di intestazione del datagramma (rete) e del segmento (trasporto)
- **UDP:** demultiplexing utilizzando (soltanto) il numero di porta di destinazione
- **TCP:** demultiplexing utilizzando 4 tuple: indirizzi IP di origine e destinazione e numeri di porta
- Il multiplexing/demultiplexing avviene a *tutti i* livelli, lo rivedremo