Lo strato di Trasporto
Introduzione

Corso di
Reti di Calcolatori
AA. 2023-2024

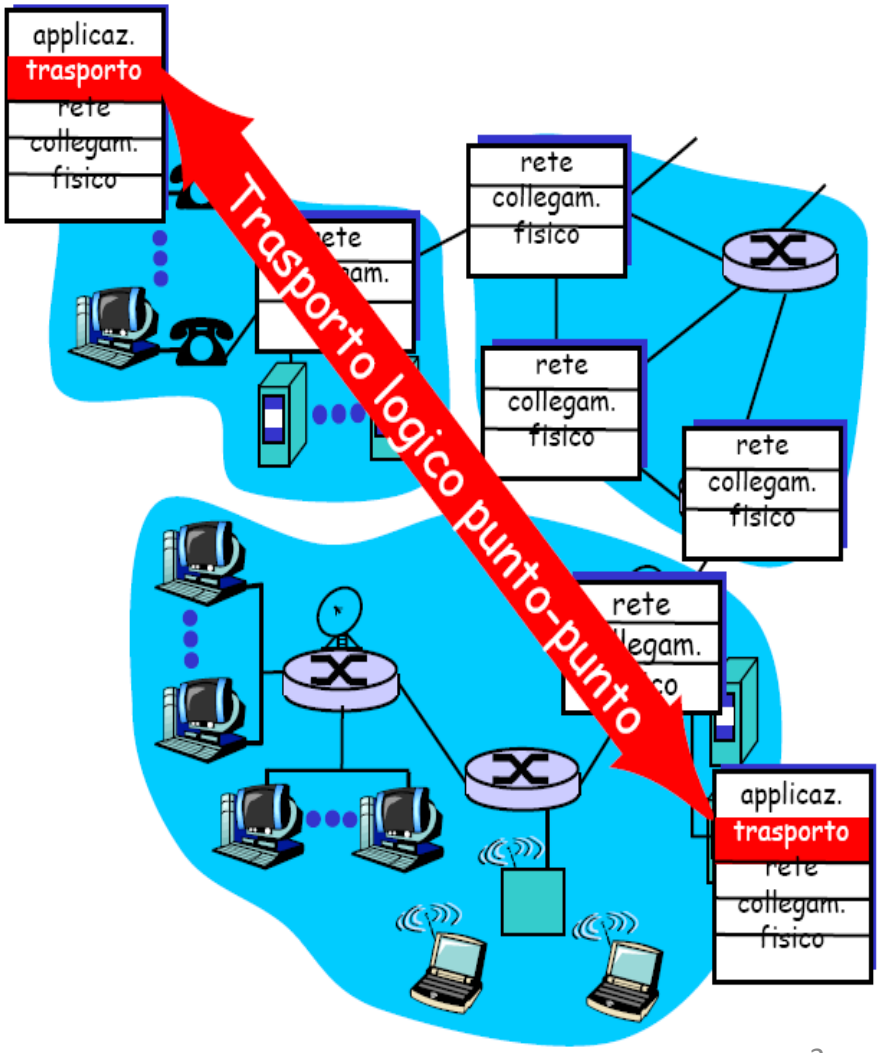
Federica Paganelli

Obiettivo

- Realizza una comunicazione logica fra processi residenti in host system diversi
  - Logico: i processi si comportano come se gli host fossero direttamente collegati, non si preoccupano dei dettagli dell'infrastruttura fisica usata per la comunicazione
- Offre servizi allo strato di applicazione
  - Un'applicazione interagisce con i protocolli di trasporto per trasmettere o ricevere dati. L'applicazione sceglie lo stile di trasporto: i) sequenza di messaggi singoli o ii) una sequenza continua di byte. Il programma applicativo passa i dati nella forma richiesta al livello di trasporto per la consegna
- Utilizza i servizi dello strato di rete.
  - Il livello di rete si occupa della comunicazione tra host
  - Il protocollo di rete consegna il datagramma all'host destinatario (non al processo)

Servizi di Trasporto

- Forniscono la comunicazione logica tra processi applicativi di host differenti
- I protocolli di trasporto vengono eseguiti nei sistemi terminali



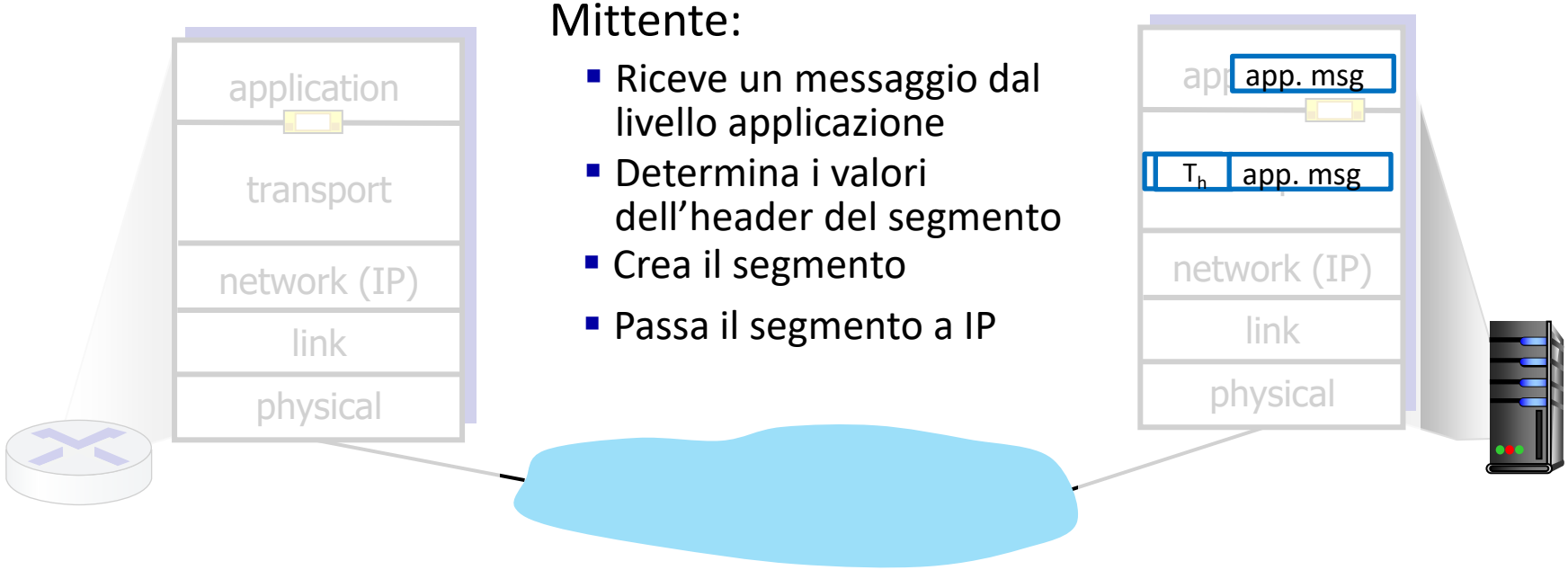
Lo strato di Trasporto

- Servizio privo di connessione
  - In un servizio privo di connessione il processo mittente consegna i messaggi al livello di trasporto uno per uno
  - Il livello di trasporto tratta ogni messaggio come entità singola senza mantenere alcuna relazione fra di essi
    - I messaggi possono non essere consegnati o non arrivare in ordine
- Servizio orientato alla connessione
  - In un servizio orientato alla connessione client e server stabiliscono una connessione (logica)

Lo strato di Trasporto

- Il Protocollo TCP [RFC 793]
  - Gestione della connessione
  - Consegna affidabile (priva di errori, completezza e ordine)
  - Flow Control
  - Controllo di congestione
- Il Protocollo UDP [RFC 768]
  - Senza connessione
  - Non affidabile, consegne senza ordine
  - Estensione “senza fronzoli” del servizio di consegna “host to host” di IP
- Servizi offerti:
  - Multiplexing/demultiplexing
  - Controllo degli errori (header + dati)

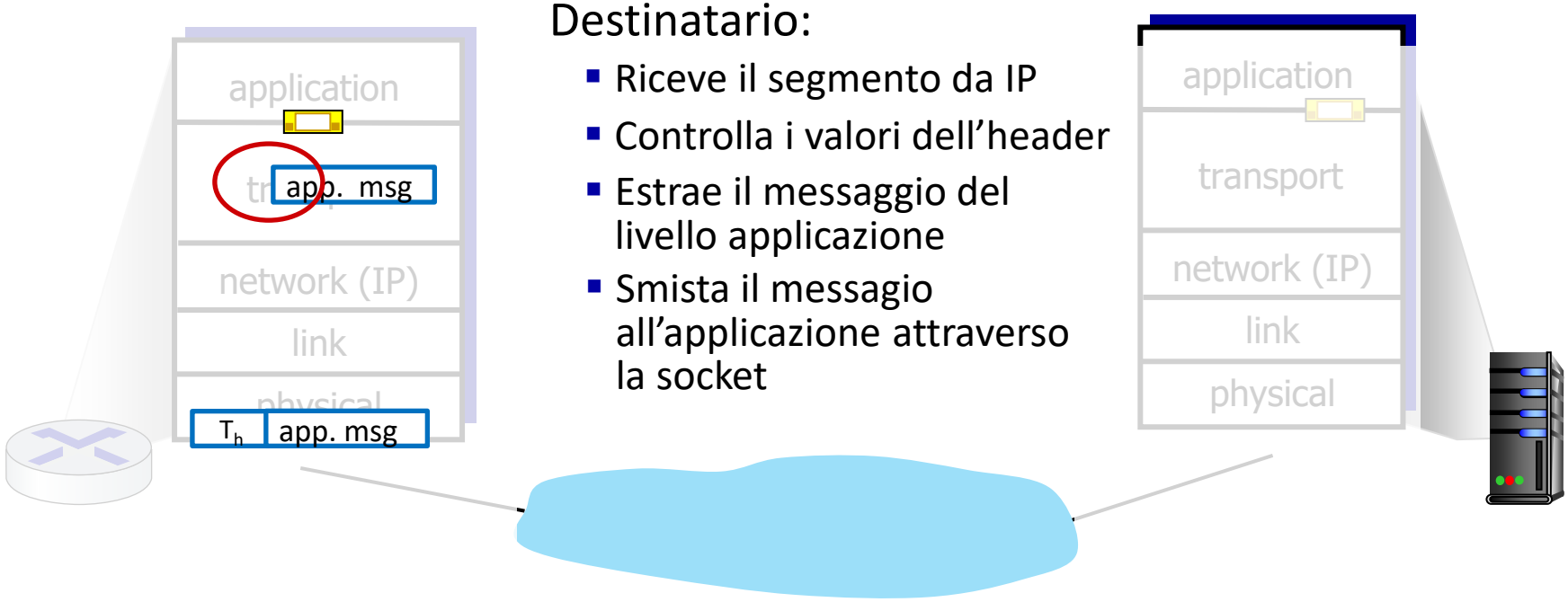
Azioni del livello di Trasporto



Mittente:

- Riceve un messaggio dal livello applicazione
- Determina i valori dell'header del segmento
- Crea il segmento
- Passa il segmento a IP

Azioni del livello di Trasporto



Destinatario:

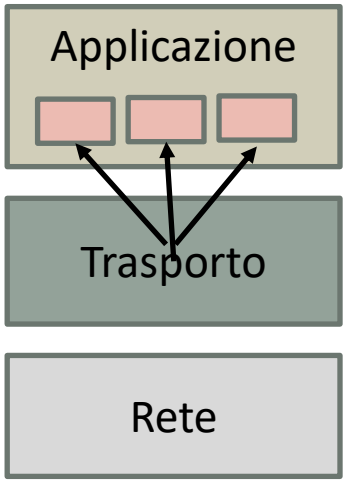
- Riceve il segmento da IP
- Controlla i valori dell'header
- Estrae il messaggio del livello applicazione
- Smista il messagio all'applicazione attraverso la socket

Concetto di "Multiplexing/Demultiplexing"

- Demultiplexing
- Lo strato Trasporto provvede allo "smistamento" dei pacchetti fra la rete e le applicazioni (processi)

  Esempio: utente che: scarica pagine Web & trasferisce file con FTP & ha 1 sessione Telnet aperta

  - ha 3 processi applicazione che utilizzano TCP
  - quando il livello trasporto riceve dati (da sotto) deve "dirigerli" a uno di questi processi



Concetto di “Multiplexing/Demultiplexing”

- Multiplexing
- Lo strato trasporto provvede all'“accorpamento” dei flussi dati dai processi verso la rete
- “Imbusta” i dati ricevuti (dall’alto) con un preambolo
- Le operazioni di multiplexing e demultiplexing si basano sui socket address dei processi.
  - Il socket address è identificato dalla combinazione indirizzo IP e numero di porta

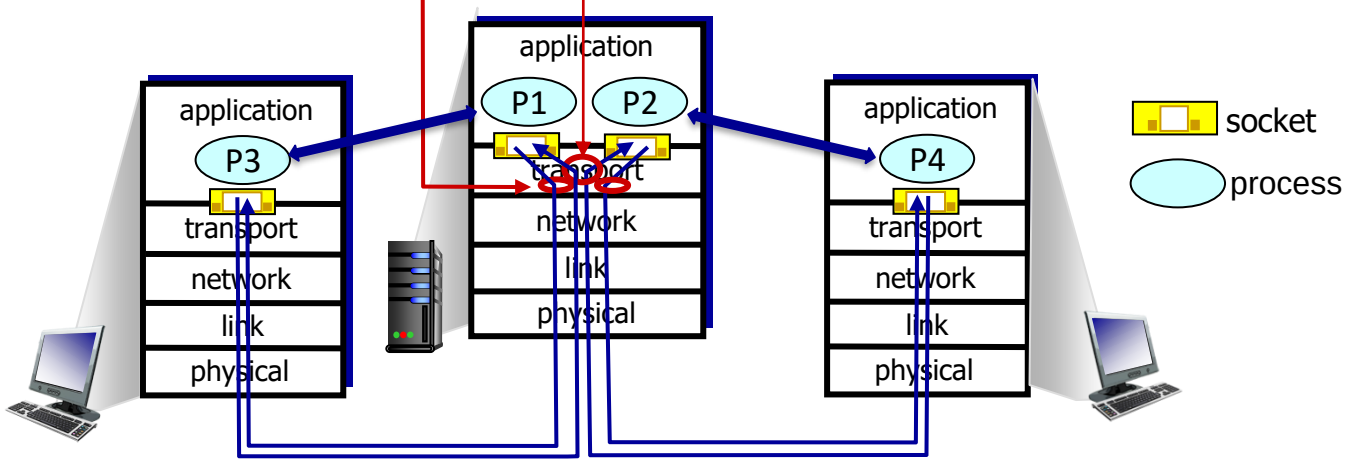
Multiplexing/demultiplexing

multiplexing lato mittente:

Gestisce dati da più socket, aggiunge l'header di trasporto

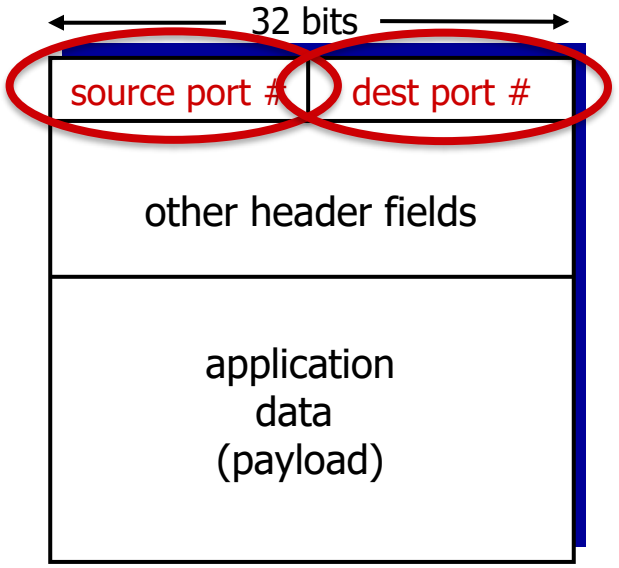
demultiplexing lato destinario:

Usa le informazioni nell'header per recapitare i segmenti alla socket corretta



Come funzione il demultiplexing

- L'host riceve il datagramma IP
  - ogni datagramma ha indirizzo IP sorgente e indirizzo IP destinatario
  - Ogni datagramma trasporta un segmento di livello trasporto
  - Ogni segmento contiene nell'header un numero di porta sorgente e un numero di porta destinazione



TCP/UDP segment format

Concetto di porta

- Ogni comunicazione di trasporto (TCP o UDP) è identificata in maniera univoca grazie alle coppie numero IP/porta degli host.
  - La "porta" è un numero (unsigned int – 16 bit [0-65535]) che viene assegnato a un processo, o più precisamente a un punto di demultiplexing dei protocolli TCP o UDP.
  - L'indirizzo IP è un indirizzo di 32 bit presente nello stack TCP/IP
- Range di porte e utilizzi in RFC 6335
  - *System Ports* (Well Known Ports): da 0 a 1023, assegnate da IANA, identificano processi server
  - *User Ports* (o Registered Ports): da 1024 a 49151, assegnate da IANA
  - *Dynamic Ports* (Private o Ephemeral Ports), non assegnate da IANA
- Il sistema operativo assegna dinamicamente le porte ai processi che ne fanno richiesta

Esempi di Well Known Ports

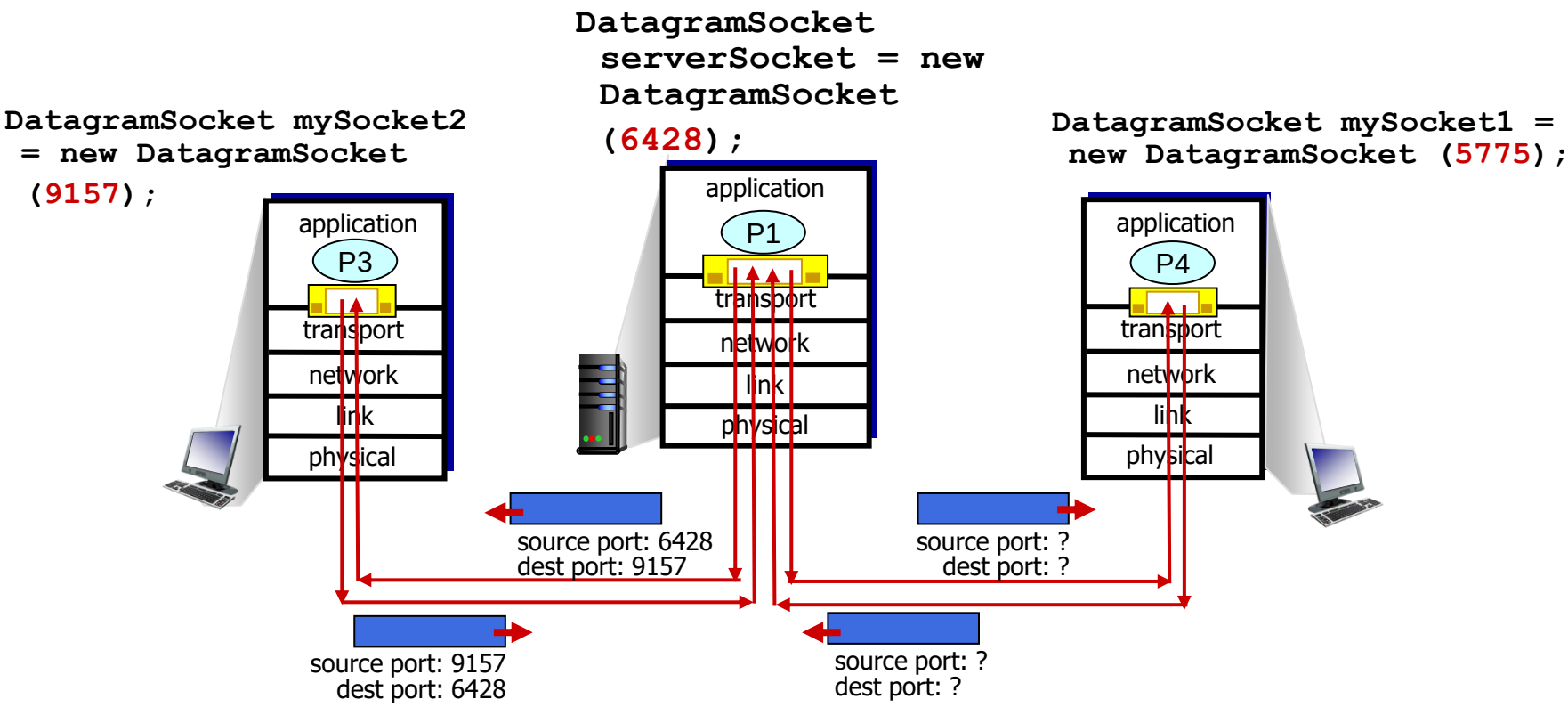
http://www.iana.org/assignments/port-numbers

SERVICE NAME	PORT/PROTOCOL	DESCRIPTION
ftp-data	20/tcp	File Transfer [Default Data]
ftp	21/tcp	File Transfer [Control]
ssh	22/tcp	SSH Remote Login Protocol
telnet	23/tcp	Telnet
smtp	25/tcp	Simple Mail Transfer
domain	53/tcp	Domain Name Server
www-http	80/tcp	World Wide Web HTTP
pop3	110/tcp	Post Office Protocol - Version 3
nntp	119/tcp	Network News Transfer Protocol
imap3	220/tcp	Interactive Mail Access Protocol v3
mysql	3306/tcp	MySQL server
tftp	69/udp	Trivial File Transfer Protocol
Domain	53/udp	Domain Name Server
RIP	520/udp	Routing Information Protocol
snmp	161/udp	Simple Network Manag. Prot.

Demultiplexing senza connessione - UDP

- es. creazione automatica di socket *DatagramSocket mySocket = new DatagramSocket()*
- Socket UDP identificata dalla coppia (IP, porta)
- Lo strato di trasporto dell'host ricevente consegna il segmento UDP alla **socket identificata da IP e porta destinazione**
- I datagrammi con IP e/o porta mittente differenti ma stessi IP e porta destinatari vengono consegnati alla stessa socket

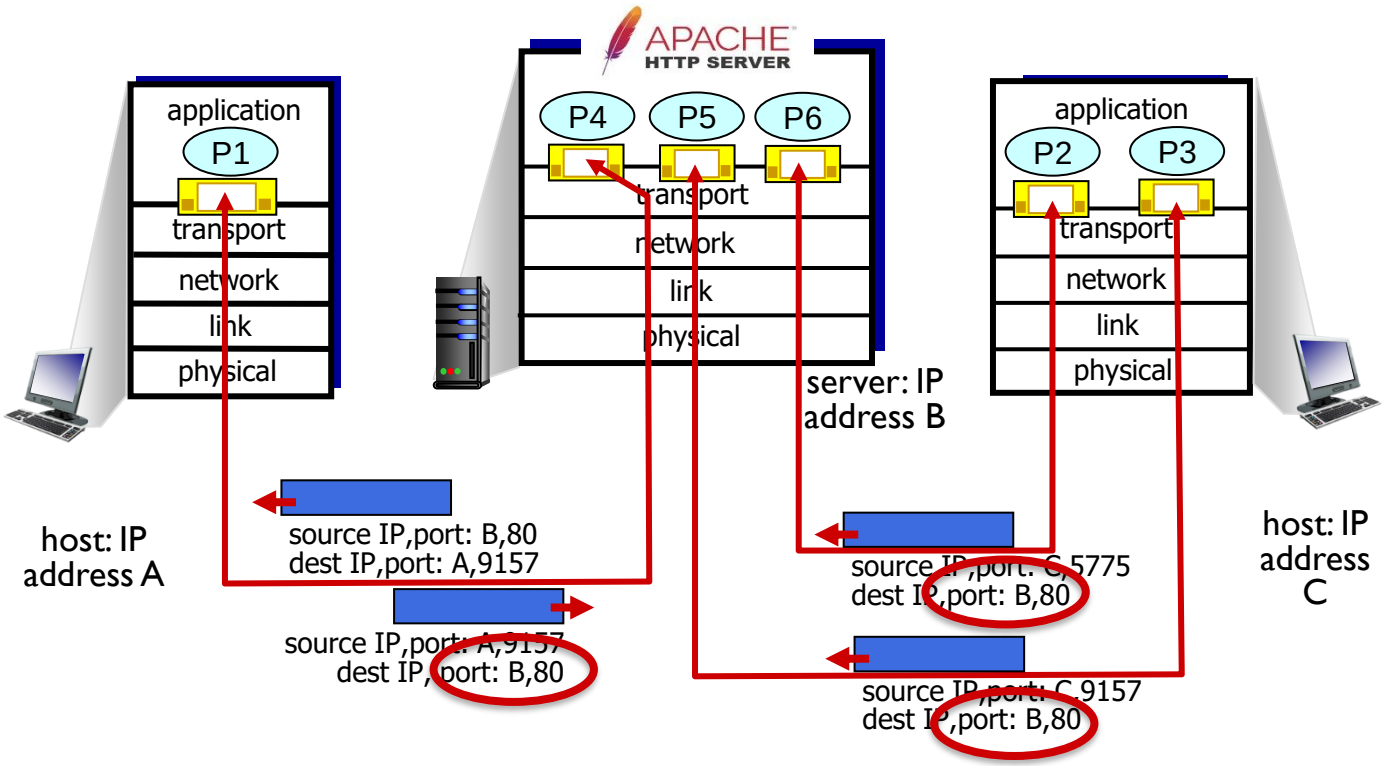
Connectionless demultiplexing: esempio



Demultiplexing orientato alla connessione

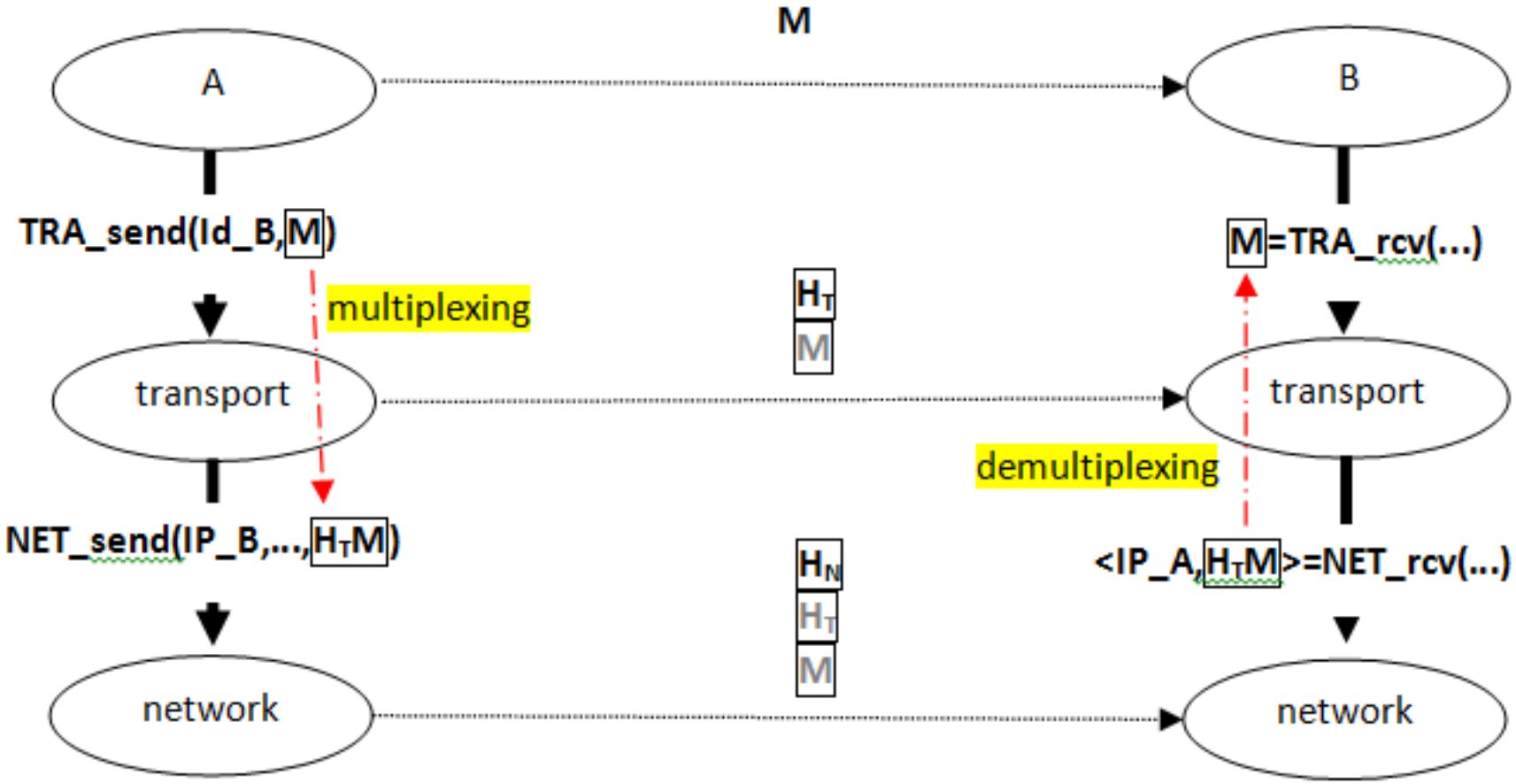
- La socket TCP connessa è identificata da 4 parametri:
  - Indirizzo IP di origine
  - Numero di porta di origine
  - Indirizzo IP di destinazione
  - Numero di porta di destinazione
- L'host ricevente usa i 4 parametri per inviare il segmento alla socket appropriata
- Un host server può supportare più socket contemporanee
- Es. server Web: socket differenti per ogni client

Connection-oriented demultiplexing: esempio

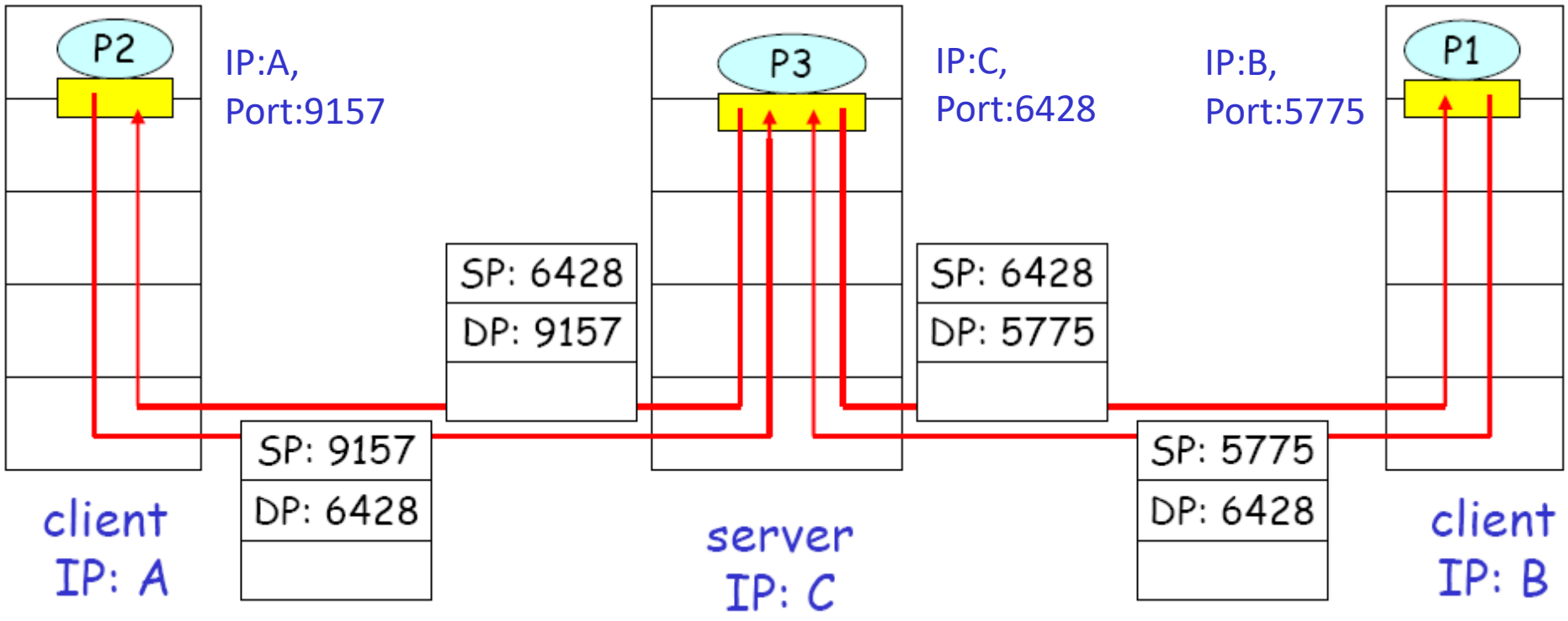


Tre segmenti con stessa destinazione IP address: B, dest port: 80 sono inviate a socket *distinte*

Interfaccia tra i livelli dello stack



Demultiplexing senza connessione – UDP (2)



Concetto di "Multiplexing/Demultiplexing"

Demultiplexing nell'host ricevente:

consegnare i segmenti ricevuti alla socket appropriata

Multiplexing nell'host mittente:

raccogliere i dati da varie socket, incapsularli con l'intestazione (utilizzati poi per il demultiplexing)