Programmazione in Rete

a.a. 2005/2006

http://www.di.uniba.it/~lisi/courses/prog-rete/prog-rete0506.htm

dott.ssa Francesca A. Lisi
lisi@di.uniba.it

Orario di ricevimento: mercoledì ore 10-12

dib

Sommario della lezione di oggi: Lo strato di applicazione (5/5)

- Principi dei protocolli dello strato di applicazione
- Trasferimento di file & il protocollo FTP
- World Wide Web & HTTP
- Posta elettronica & SMTP
- DNS: il servizio directory di Internet
- Condivisione di file
- Programmazione delle socket

Programmazione delle socket

Scopo: comprendere come le applicazioni client/server comunicano attraverso le socket

Socket API

- introdotte in BSD4.1 UNIX, 1981
- esplicitamente create, usate, rilasciate dalle applicazioni
- paradigma client/server
- due tipi di servizio di trasporto via socket API:
  - affidabile, byte stream-oriented (cfr. TCP)
  - inaffidabile, datagram-oriented (cfr. UDP)

socket

una interfaccia
- *host-local*,
- *application-created/owned*,
- *OS-controlled*

(una "porta") in cui un processo applicativo può sia inviare che ricevere messaggi da/a un altro processo applicativo (remoto o locale)

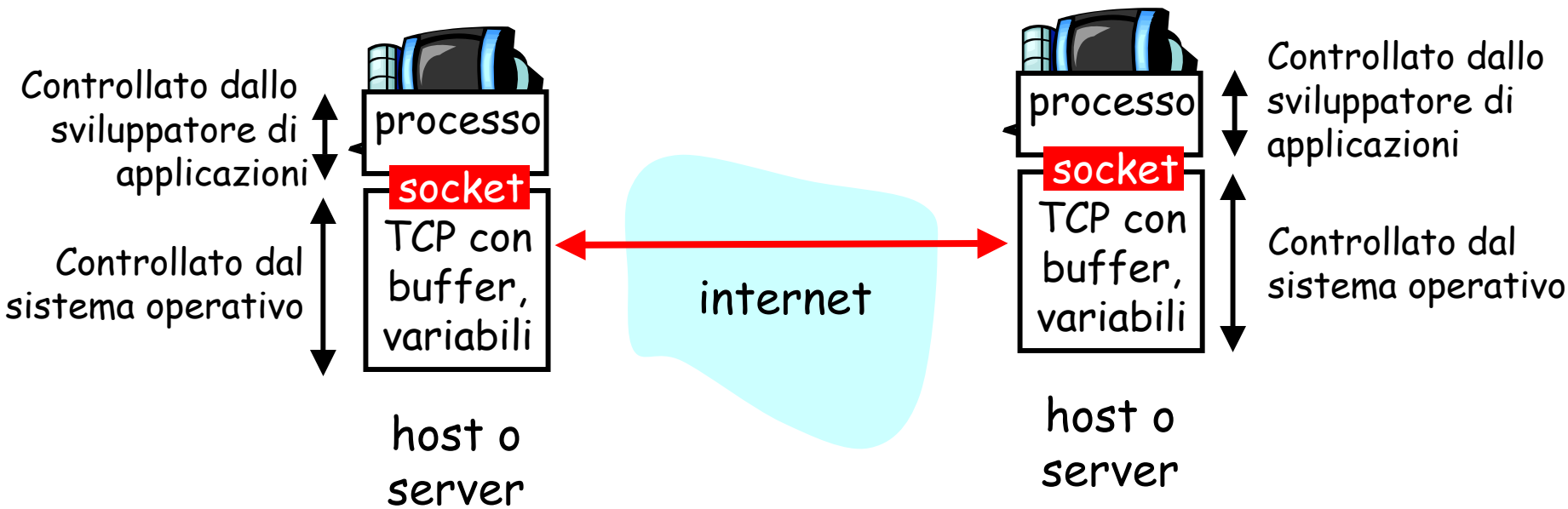
Programmazione delle socket: un esempio di applicazione client/server

- il client legge una linea dal suo *ingresso standard* (tastiera) e la invia attraverso la sua socket al server
- il server legge una linea dalla sua socket
- il server converte la linea in maiuscolo
- il server invia al client la linea modificata facendola passare attraverso la sua socket
- il client legge la linea (modificata) dalla sua socket e la stampa sulla sua *uscita standard* (monitor)

Programmazione delle socket con TCP

Socket: una porta fra il processo applicativo ed il protocollo TCP per il trasporto end-to-end

Servizio TCP: trasferimento affidabile di byte da un processo ad un altro



Programmazione delle socket con TCP

Il client deve contattare il server

- Il processo server deve innanzitutto essere in esecuzione
- il server deve aver creato una socket che accolga il contatto da parte del client

Il client contatta il server:

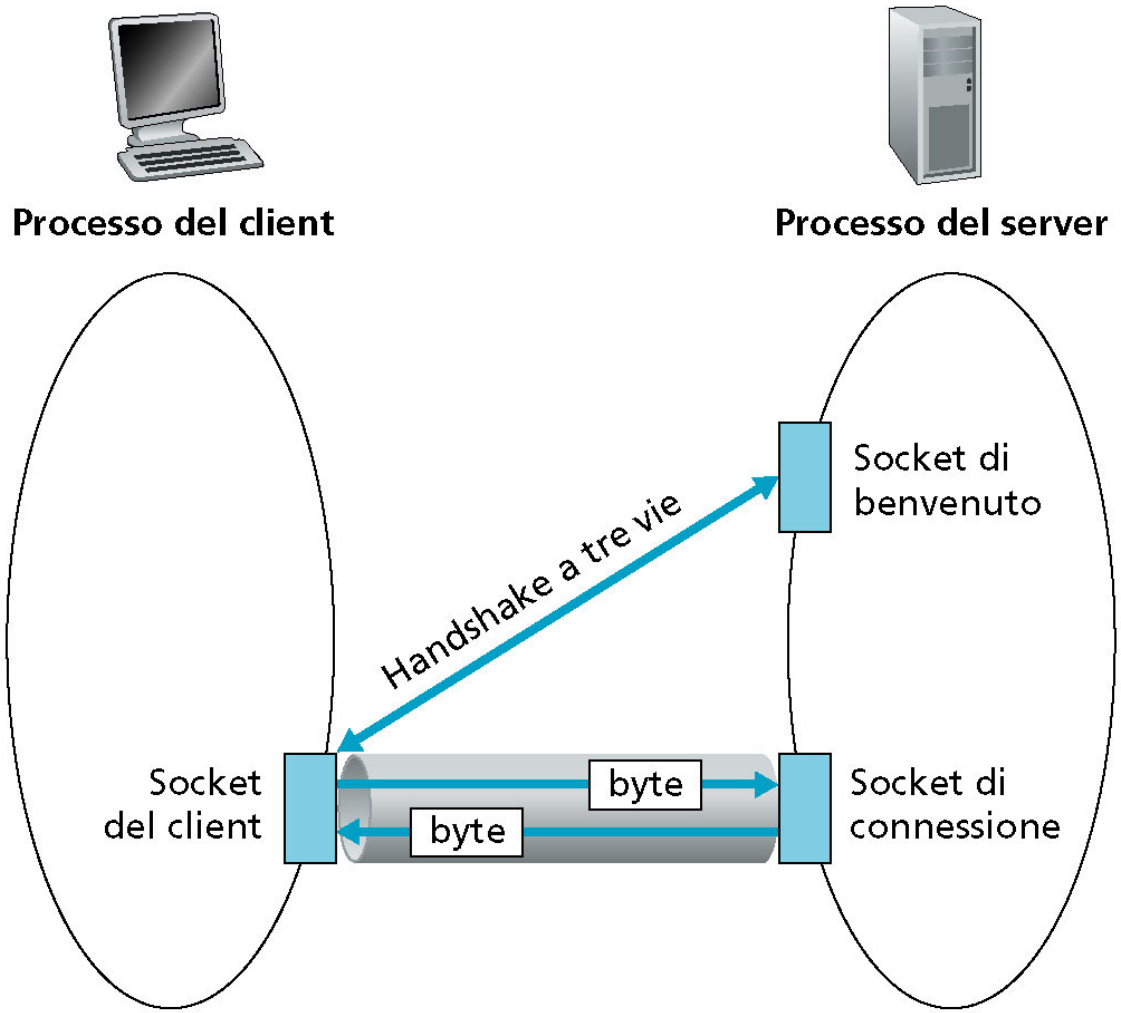
- creando una socket TCP locale al client
- specificando indirizzo IP e numero di porta del processo server

- Quando il client crea la socket: il client TCP stabilisce una connessione al server TCP
- Se contattato dal client, il server TCP crea una nuova socket per il processo server per comunicare con il client
  - consente al server di parlare con client multipli

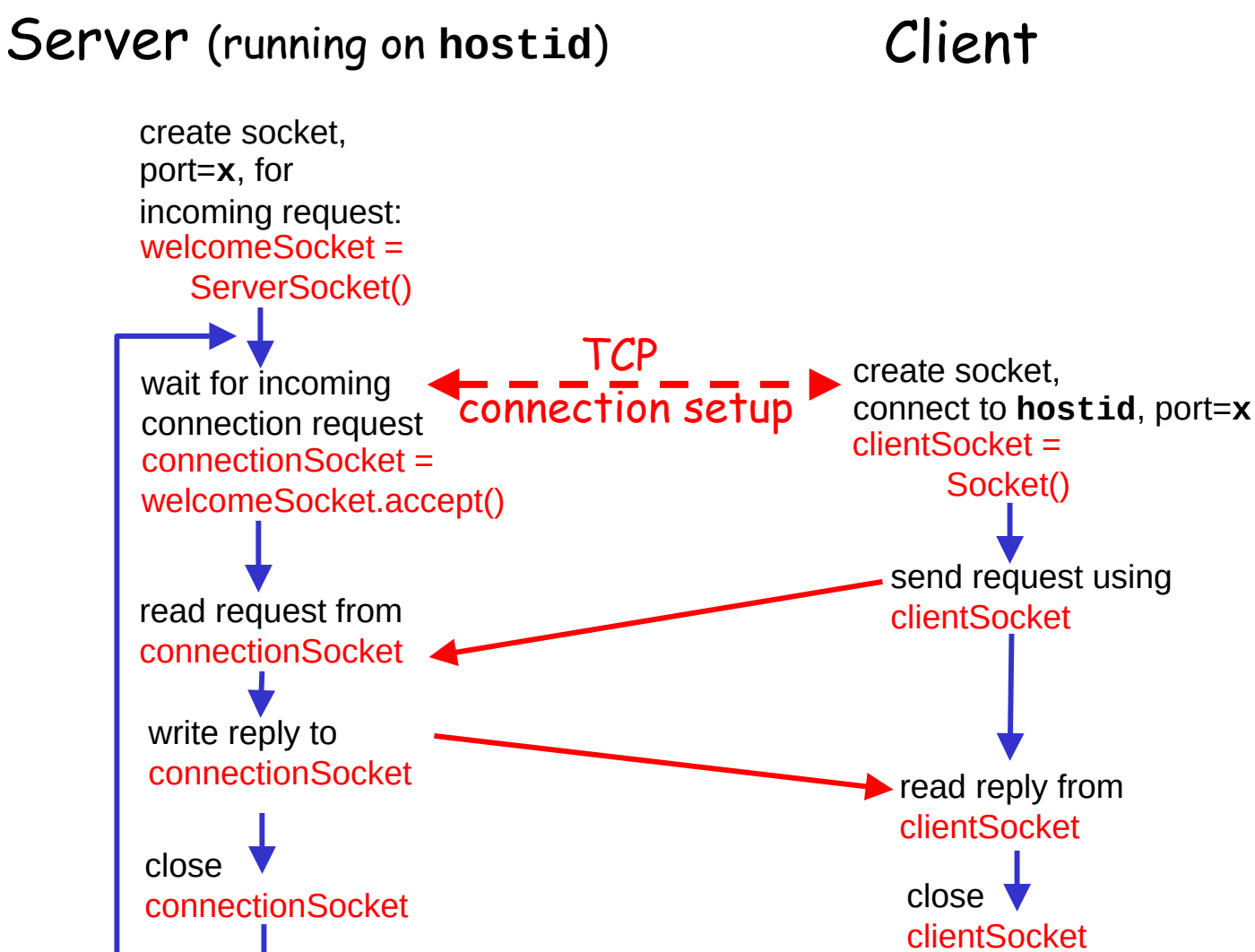
Punto di vista applicativo

Il TCP fornisce un trasferimento affidabile, in-order di byte ("pipe") fra client e server

Programmazione delle socket con TCP: interazione client/server

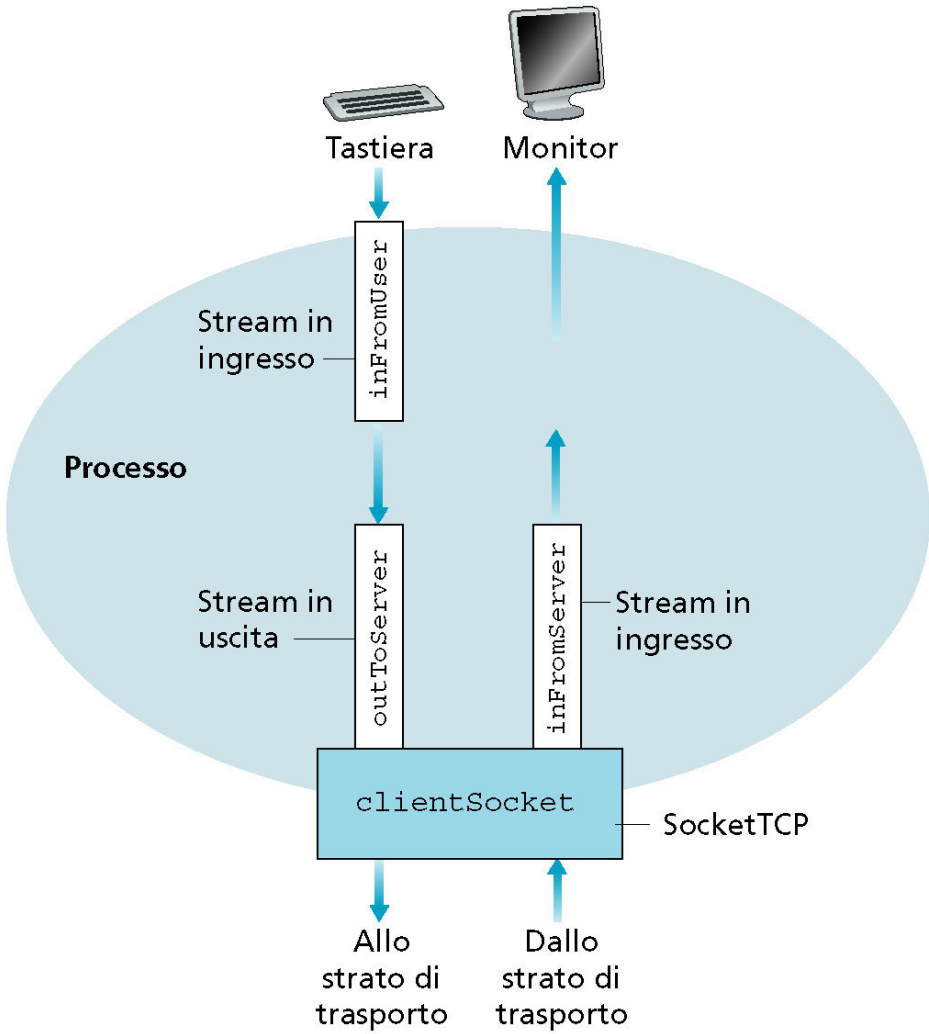


Programmazione delle socket con TCP: interazione client/server (cont.)



Programmazione delle socket con TCP: lato client

- il client legge una riga (stream in ingresso **`inFromUser`**) dallo standard input (tastiera) e la invia al server in byte (stream in uscita **`outToServer`**) attraverso la sua socket
- il client legge la linea modificata dalla sua socket (stream in ingresso **`inFromServer`**) e la manda in stampa sullo standard output (monitor)
- N.B. Un flusso (*stream*) è una sequenza di caratteri che fluisce verso/da un processo.



Programmazione delle socket con UDP

UDP: nessuna "connessione" fra client e server

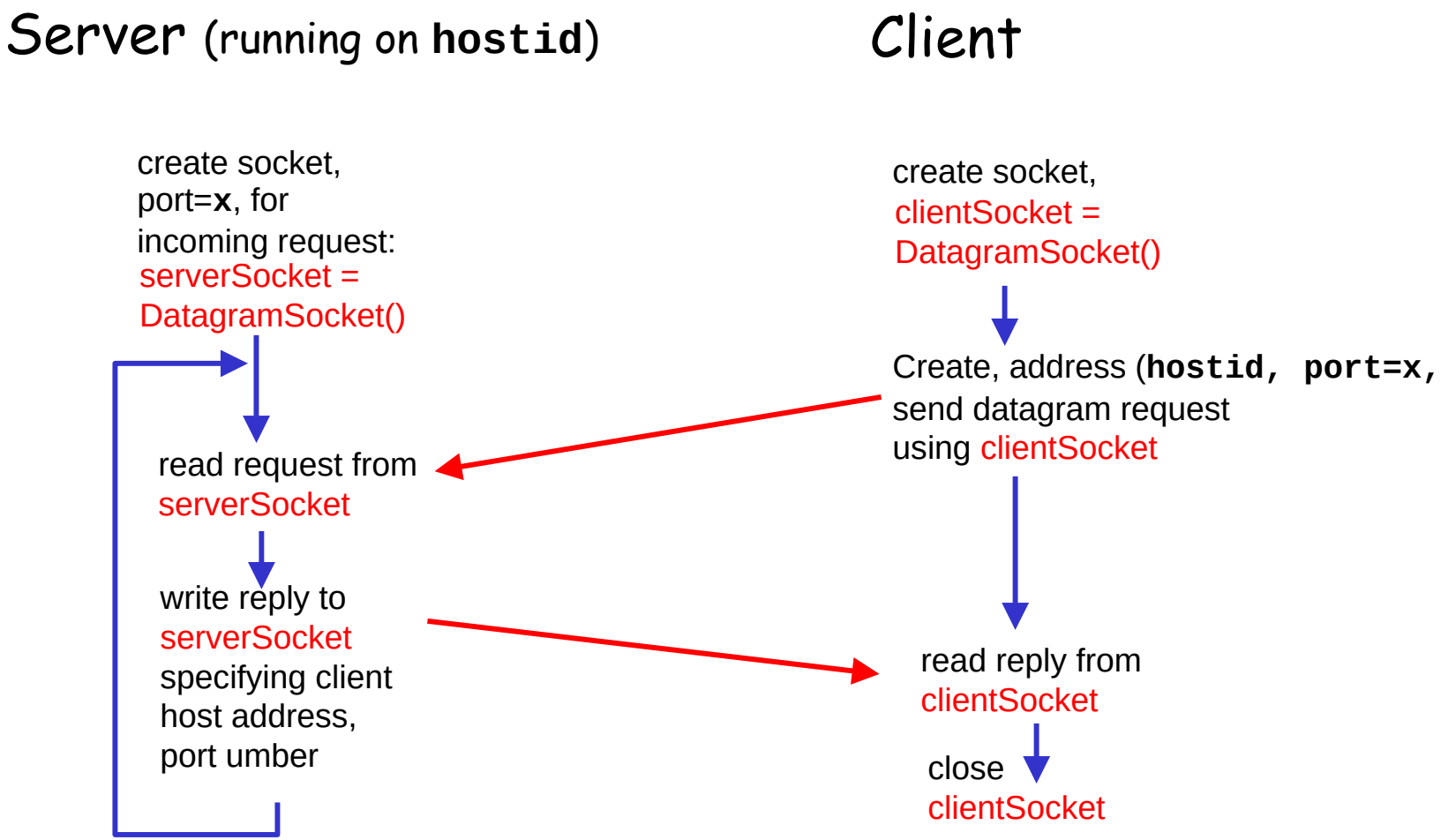
- no handshaking
- il mittente attacca esplicitamente l'indirizzo IP e il numero di porta del destinatario
- il server deve estrarre l'indirizzo IP e numero di porta del mittente dal datagramma ricevuto

UDP: i dati trasmessi potrebbero essere ricevuti non in ordine, o andare persi

application viewpoint

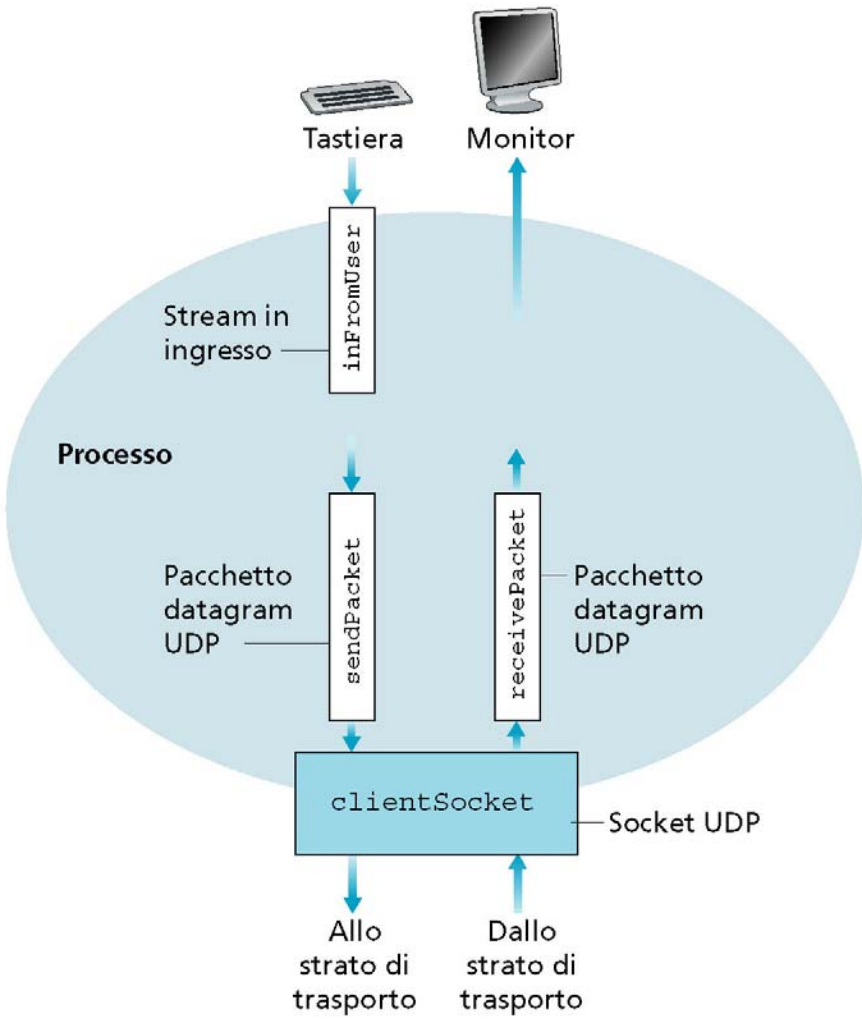
UDP fornisce trasferimento inaffidabile di gruppi di byte ("datagrammi") fra client e server

Programmazione delle socket con UDP: interazione client/server

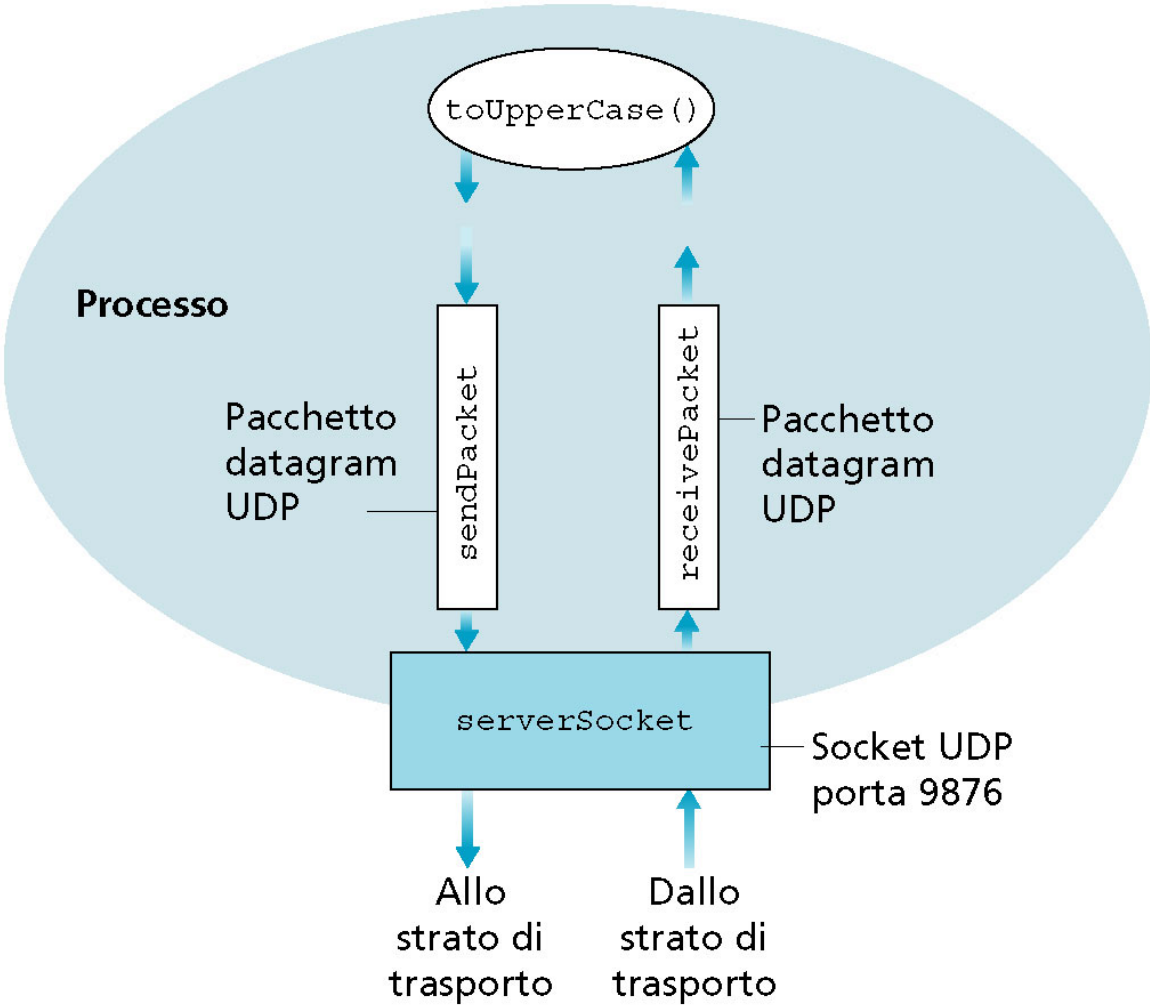


Programmazione delle socket con UDP: lato client

- il client legge una riga (stream in ingresso **`inFromUser`**) dallo standard input (tastiera) e la invia in pacchetti datagram UDP (**`sendPacket`**) al server attraverso la socket
- il client legge il pacchetto datagram UDP contenente la linea modificata (**`receivePacket`**) attraverso la sua socket e la manda in stampa sullo standard output (monitor)



Programmazione delle socket con UDP: lato server



Lo strato di applicazione: Conclusioni

Abbiamo saputo di più su:

- gli aspetti concettuali ed implementativi delle applicazioni di rete
  - paradigma client-server
  - Paradigma peer-to-peer
  - programmazione delle socket
- le principali applicazioni per Internet
  - file transfer, WWW, posta elettronica, servizio directory
- i protocolli sottostanti tali applicazioni
  - FTP, HTTP, SMTP, POP3, DNS
- i modelli di servizio (il COSA) offerti da TCP e UDP
- Nelle prossime lezioni sullo strato di trasporto sapremo il COME ed il PERCHE' di TCP ed UDP