



POLITECNICO
MILANO 1863



SLIDES
PRE-LAB

Fondamenti di Comunicazioni e Internet

Antonio Capone, Matteo Cesana,
Guido Maier, Francesco Musumeci



POLITECNICO
MILANO 1863

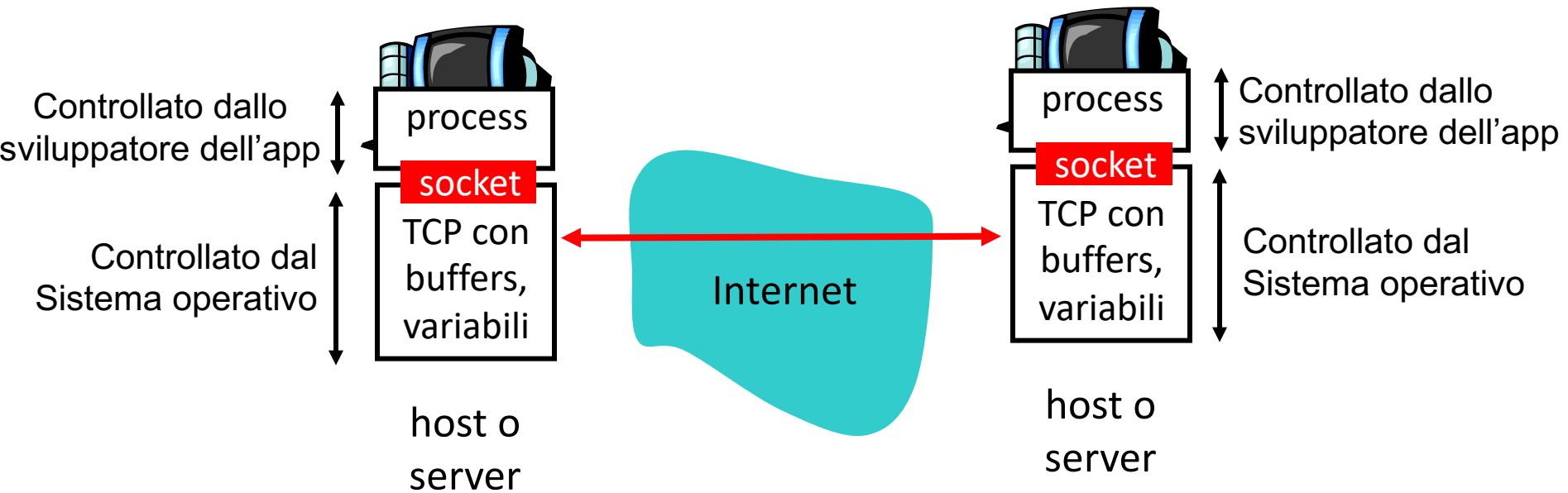


Programmazione Socket: TCP

Antonio Capone, Matteo Cesana,
Guido Maier, Francesco Musumeci

Programmazione socket con TCP

Servizio TCP: trasferimento affidabile di **bytes** da un processo all'altro



Programmazione socket con TCP

1. Attivazione Processo Server:

Il processo server deve essere eseguito **per primo** e deve aver **creato un socket (porta)** per accogliere le richieste del client

Welcome socket: assomiglia al socket UDP e può ricevere da tutti i client

2. Il Client deve contattare il Server:

Creando un socket TCP e specificando indirizzo IP e numero di porta del processo server

Punto di vista dell'applicazione

TCP fornisce trasferimento affidabile e ordinato di bytes ("pipe") tra client e server

Programmazione socket con TCP

3. Instaurazione della connessione:

Il Client TCP **instaura una connessione** tra il proprio socket e il Welcome socket del Server TCP

4. Inizio della trasmissione:

Quando contattato dal client, il **server TCP crea un nuovo socket per la comunicazione tra processo server e client:**

- Così da poter **comunicare con diversi client**
- **Usando i numeri di porta sorgente per distinguere i client**

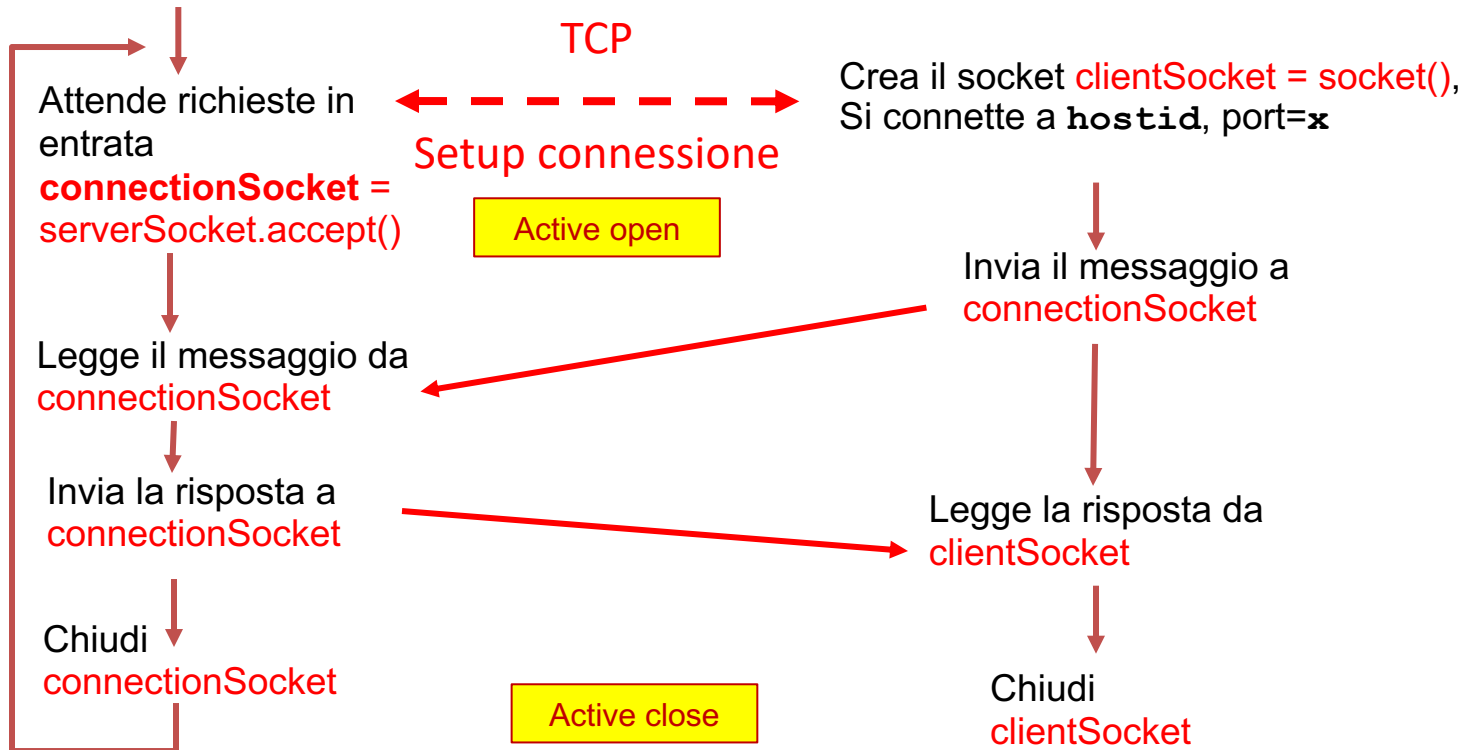
Interazione tra socket Client/server: TCP

Server (running su `hostid`)

Crea Welcome Socket sulla porta `x`, per una data richiesta in entrata:

Passive open

`serverSocket = socket()`



Stream jargon

Uno **stream** (flusso) è una sequenza di caratteri che scorrono in input/output da un processo.

Un **input stream** viene associato ad una sorgente di input per un processo (es: una tastiera o un socket)

Un **output stream** viene associato ad una sorgente d'uscita (es: un monitor o un socket)

