



## L'interfaccia di programmazione socket

Mario BALDI  
staff.polito.it/mario.baldi

Silvano GAI  
sgai@cisco.com

Fulvio RISSO  
fulvio.risso@polito.it

© M. Baldi: see page 2

## Nota di Copyright

Questo insieme di trasparenze (detto nel seguito slide) è protetto dalle leggi sul copyright e dalle disposizioni dei trattati internazionali. Il titolo ed i copyright relativi alle slide (ivi inclusi, ma non limitatamente, ogni immagine, fotografia, animazione, video, audio, musica e testo) sono di proprietà degli autori indicati a pag. 1.

Le slide possono essere riprodotte ed utilizzate liberamente dagli istituti di ricerca, scolastici ed universitari afferenti al Ministero della Pubblica Istruzione e al Ministero dell'Università e Ricerca Scientifica e Tecnologica, per scopi istituzionali, non a fine di lucro. In tal caso non è richiesta alcuna autorizzazione.

Ogni altra utilizzazione o riproduzione (ivi incluse, ma non limitatamente, le riproduzioni su supporti magnetici, su reti di calcolatori e stampate) in toto o in parte è vietata, se non esplicitamente autorizzata per iscritto, a priori, da parte degli autori.

L'informazione contenuta in queste slide è ritenuta essere accurata alla data della pubblicazione. Essa è fornita per scopi meramente didattici e non per essere utilizzata in progetti di impianti, prodotti, reti, ecc. In ogni caso essa è soggetta a cambiamenti senza preavviso. Gli autori non assumono alcuna responsabilità per il contenuto di queste slide (ivi incluse, ma non limitatamente, la correttezza, completezza, applicabilità, aggiornamento dell'informazione).

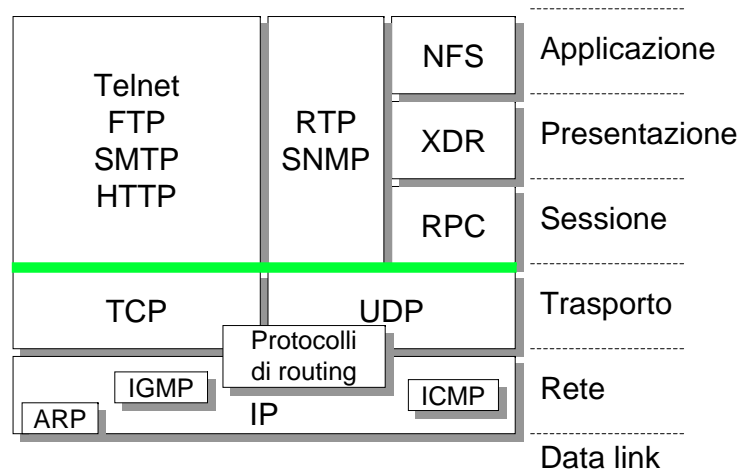
In ogni caso non può essere dichiarata conformità all'informazione contenuta in queste slide.

In ogni caso questa nota di copyright non deve mai essere rimossa e deve essere riportata anche in utilizzi parziali.

socket - 2

© M. Baldi: see page 2

## I protocolli TCP e UDP



socket - 3

© M. Baldi: see page 2

## I socket

- sono l'interfaccia base per le comunicazioni basate su TCP/IP
- paradigma Client - Server
- sono i punti estremi della comunicazione
- una coppia di socket connessi fornisce un servizio tipo pipe
- identificato da un socket descriptor
  - analogo ad un file descriptor
- Tipi:
  - stream: connessione di tipo affidabile (TCP)
  - datagram: comunicazioni di tipo datagram (UDP)
  - raw: per la manipolazione diretta del pacchetto IP
    - realizzazione di un protocollo di livello 4

socket - 4

© M. Baldi: see page 2

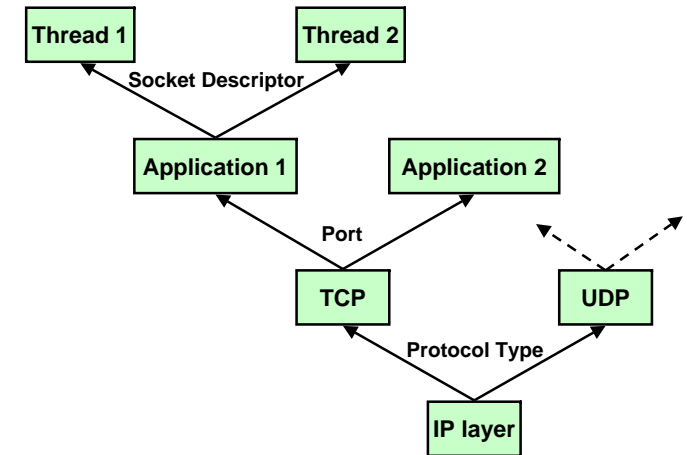


## Socket descriptor

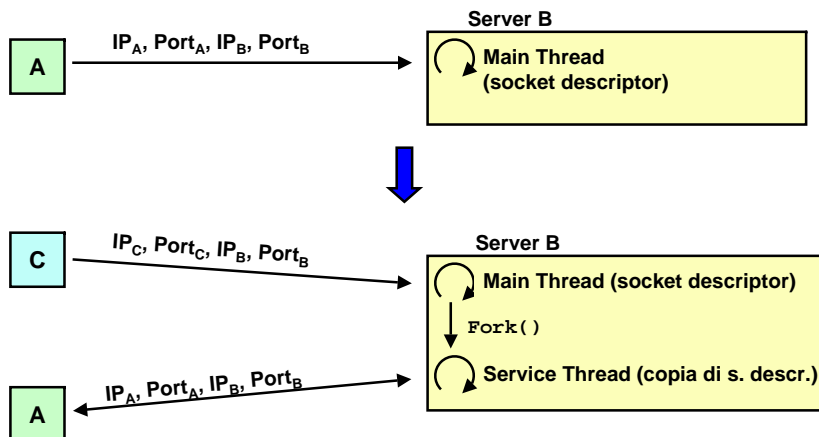
- in UNIX è un normale file descriptor riferito ad un socket anziché ad un file
- può essere usato normalmente per la lettura o la scrittura
- si possono usare tutte le system call operanti su file



## Socket descriptor



## Connessioni multiple con socket TCP



## Operazioni con i socket: lato client

- **socket()**: crea un socket descriptor
- **bind()**: collega il descriptor ad una porta locale
- **connect()**: collega il socket locale con uno remoto instaurando la connessione
  - se il socket è di tipo stream viene aperta una connessione TCP
- **read()**, **write()**, ...
- **sendto()**, **recvfrom()**



## Operazioni con i socket: lato server

- **socket()**: crea un socket descriptor
- **bind()**: collega il descriptor ad una porta locale
- **listen()**: mette il processo in attesa di richieste di connessioni
- **accept()**: è bloccante e crea un nuovo socket all'arrivo di una nuova connessione
- Utilizzo
  - un processo server esegue **listen()** e si blocca in attesa di richieste di connessione
  - all'arrivo di una connessione il server esegue **accept()** per creare un nuovo descriptor
  - il server clona se stesso (**fork()**) o crea un thread per la gestione della nuova connessione
  - il processo/thread originale esegue nuovamente **listen()**