

Socket

Generalità

In generale un'applicazione di rete è costituita da un insieme di programmi che vengono eseguiti su due o più computer contemporaneamente.

I processi hanno la necessità di scambiare messaggi con gli altri processi della medesima applicazione, sia che essi appartengano alla stessa rete locale oppure che siano remoti e quindi dislocati dall'altra parte del globo: per comunicare tra loro questi processi devono mettersi in "contatto" tramite i loro indirizzi e utilizzare i servizi offerti dal livello di applicazione.

TCP e UDP

TCP e UDP svolgono funzioni diverse, cioè offrono servizi diversi allo strato applicativo:

- TCP è un protocollo di trasporto orientato alla connessione ed è affidabile dato che consente il controllo dell'integrità dell'informazione contenuta nei pacchetti e dispone di un sistema per segnalare l'errore al mittente.
- UDP è un protocollo di trasporto non orientato alla connessione e non affidabile.

Socket

Affinché un processo, presente su un determinato host, invii un messaggio a un qualsiasi altro host, il processo mittente deve identificare il processo destinatario in modo univoco.

Generalmente ogni PC ha una sola porta fisica di connessione al network, se più applicazioni necessitano di utilizzare la rete, devono essere riconosciute in qualche modo.

Questo metodo di riconoscimento viene effettuato tramite le cosiddette porte logiche identificate da un numero detto **numero di porta**. Il numero di porta è di 16 bit e può quindi assumere valori compresi tra 0 e 65535.

Socket

Le porte da 0 a 1023 sono dette "well-known ports" e sono riservate per i servizi di rete più comuni, come ad esempio http (80), ftp (21), telnet (23), ecc.

Le porte da 1024 a 49151 sono dette "registered ports" e sono riservate per i servizi di rete meno comuni, ma possono essere utilizzate da chiunque.

Le porte da 49152 a 65535 sono dette "dynamic and/or private ports" e sono riservate per i servizi di rete temporanei, come ad esempio le connessioni ftp passive o le connessioni telnet.

Socket

Il solo numero di porta non è sufficiente per identificare univocamente un processo, infatti, se più processi utilizzassero la stessa porta, non sarebbe possibile distinguere i messaggi inviati da un processo da quelli inviati da un altro. Per questo motivo, oltre al numero di porta, è necessario utilizzare anche l'indirizzo IP.

È bene sottolineare il fatto che i numeri di porta logica sono relativi soltanto al protocollo considerato: una determinata porta per il protocollo TCP è diversa dallo stesso numero di porta per il protocollo UDP (si tratta effettivamente di porte diverse) anche se in genere viene utilizzato lo stesso numero di porta per un servizio che gestisce entrambi i protocolli.

Un socket consente quindi di comunicare attraverso la rete usando la pila TCP/IP.

Socket

Esiste una struttura chiamata "association" che contiene tutte le informazioni necessarie per identificare univocamente una connessione, cioè:

- l'indirizzo IP del mittente
- il numero di porta del mittente
- l'indirizzo IP del destinatario
- il numero di porta del destinatario

Socket

Il modello client-server è organizzato in due moduli, chiamati appunto server e client, operanti su macchine diverse:

- il server svolge le operazioni necessarie per realizzare un servizio
- il client, generalmente tramite un'interfaccia utente, acquisisce i dati, li elabora e li invia al server richiedendo un servizio.

Quindi in rete sono presenti calcolatori su cui girano processi server che erogano servizi e sono in attesa di ricevere richieste di connessione da parte di processi client interessati a usufruire di tali servizi.

Socket

Il concetto di socket è stato sviluppato come estensione diretta del paradigma UNIX di I/O su file, che si basa sulla sequenza di operazioni open-read-write-close:

- open: permette di accedere ad un file
- read/write: accedono ai contenuti del file
- close: terminazione dell'utilizzo del file

L'utilizzo dei socket avviene pressoché con la stessa modalità ma aggiungendo a questa struttura l'insieme dei parametri necessari a realizzare la connessione tra macchine remote, cioè richiedendo:

- gli indirizzi
- il protocollo e numero di porta
- il tipo del protocollo

Socket

Ogni sistema operativo mette a disposizione nelle API i meccanismi per realizzare l'interfacciamento tra diversi protocolli: le socket API sono delle specifiche API di protocollo che hanno origine con Berkeley BSD UNIX e che oggi sono disponibili in qualunque sistema operativo.

Elenchiamo le funzioni presenti sia in C che in Java che verranno utilizzate in seguito:

- `socket()`: crea un socket
- `bind()`: assegna un indirizzo di rete ad un socket
- `listen()`: mette in ascolto un socket
- `accept()`: accetta una connessione in ingresso
- `connect()`: stabilisce una connessione con un altro socket

- `send()`: invia dati su un socket
- `recv()`: riceve dati da un socket
- `close()`: chiude un socket

Famiglie di socket

Le famiglie di socket sono un insieme di protocolli che possono essere utilizzati per realizzare una connessione. Le famiglie di socket più comuni sono:

- AF_INET: per la comunicazione in Internet
- AF_UNIX: per la comunicazione in locale

Tipi di socket

I socket sono fondamentalmente di tre tipi e per ciascuna tipologia abbiamo una diversa modalità di connessione:

- **Stream socket:** sono socket orientati alla connessione e sono utilizzati per la comunicazione affidabile e bidirezionale. La connessione è stabilita tramite il protocollo TCP e la comunicazione è full-duplex. I dati vengono inviati in modo sequenziale e non vengono per nessun motivo duplicati o persi.
- **Datagram socket:** sono socket non orientati alla connessione e sono utilizzati per la comunicazione non affidabile e bidirezionale. La connessione è stabilita tramite il protocollo UDP e la comunicazione è half-duplex. I dati vengono inviati in modo non sequenziale e possono essere duplicati o persi.

- **Raw socket:** sono socket non orientati alla connessione e sono utilizzati per la comunicazione non affidabile e bidirezionale. Vengono utilizzati per inviare pacchetti IP personalizzati.