

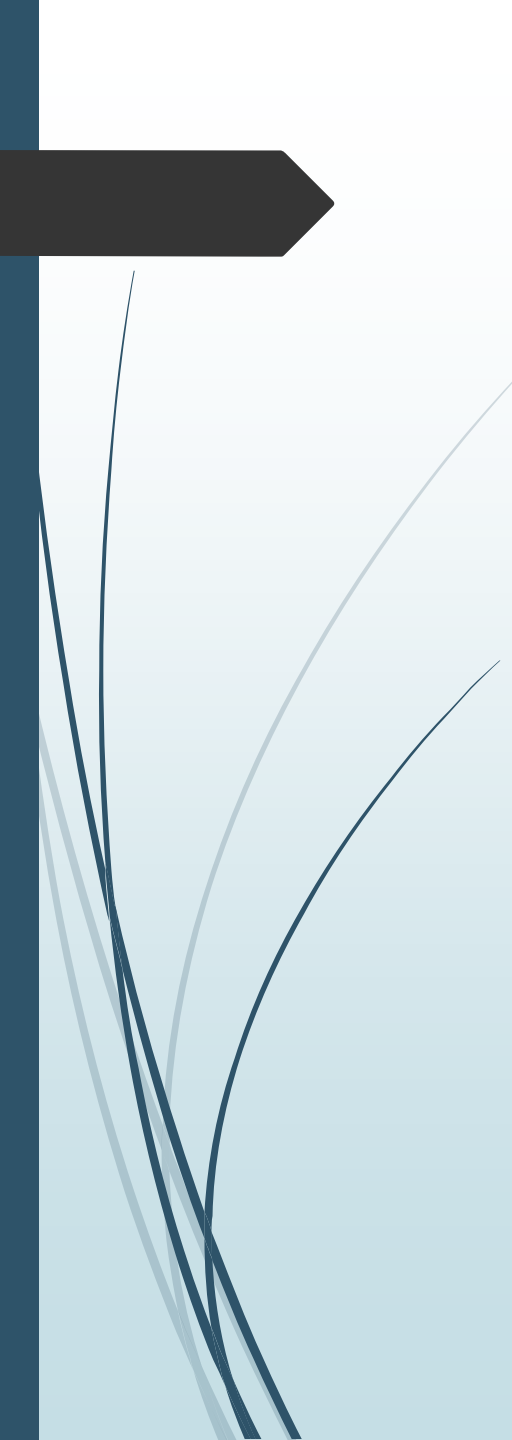


Introduzione ai Socket Java



Socket Java

- In una rete di ampie dimensioni, ciascuna sottorete (es. LAN, WAN) è connessa ad altre sottoreti tramite router. Internet è un insieme di reti connesse tra loro.
- Essenzialmente, in una rete alcune macchine possono fornire dei servizi ad altre macchine connesse. Spesso tali servizi si basano sul modello client/server.
- **Client:** macchina o processo che richiede un servizio ad un'altra macchina o processo attraverso un **protocollo**.
- **Server:** fornisce un servizio al client.
- **Protocollo:** un set di regole che descrive come trasmettere dati.

- 
- Il modello client/server consente a due processi di condividere risorse e di cooperare per il raggiungimento di un obiettivo.
 - La comunicazione dei sistemi che usano il modello client/server può avvenire attraverso le **socket** (presa o spinotto). Esse forniscono un'astrazione rispetto a meccanismi di più basso livello per la comunicazione in rete.
 - Una socket può essere rappresentata come il punto di accesso per il canale di comunicazione di due parti che risiedono su host diversi (o sullo stesso host).
 - Una socket in uso è solitamente legata ad un indirizzo ed ad una porta.
 - La libreria Java che fornisce il supporto per le socket è `java.net`



Il dominio Internet : indirizzi IP e le porte

- Un **indirizzo IP** (Internet Protocol) è costituito da quattro numeri interi di 8 bit (con valori da 0 a 255) separati da punti (es. 151.97.253.200) e consente di individuare univocamente la posizione di una sottorete e di un host al suo interno.
- Per usare le socket, oltre a conoscere l'indirizzo IP dell'host a cui connettersi, bisogna disporre dell'informazione sufficiente per collegarsi al processo server corretto.
- Per questo motivo esistono i **numeri di porta** (port number) che permettono di associare un servizio (un processo server che risponde alle richieste) ad un ben determinato numero.



Il dominio Internet : indirizzi IP e le porte

- Le connessioni avvengono sempre specificando un indirizzo IP ed un numero di porta.
- I numeri di porta sono interi di 16 bit (da 0 a 65535). Quelli da 0 a 1023 sono riservati per i servizi standard (es. FTP porta 21, HTTP porta 80, etc.), mentre i numeri da 1024 a 65535 sono disponibili per i processi utente.



Tipi di Socket

- Esistono due modi principali per comunicare in rete:
 - **connectionless**
 - **connection oriented**
- In corrispondenza a tali due modi di comunicazione si hanno i seguenti tipi di socket, rispettivamente:
 - **Socket a datagrammi (Datagram)**
 - **Stream socket**



Socket a datagrammi (Datagram)

- **Socket a datagrammi (Datagram)** trasferiscono messaggi di dimensione variabile, preservando i confini, **ma** senza garantire ordine o arrivo dei pacchetti. Supportate nel dominio Internet dal protocollo UDP (User Datagram Protocol).
- Non instaurano una connessione tra client e server, e non verificano l'arrivo del dato o il ri-invio. Hanno il vantaggio di trasferire velocemente i dati.
- La classe Java che usa questo protocollo è DatagramSocket. I dati sono impacchettati tramite la classe DatagramPacket.



Socket a datagrammi (Datagram)

- Usare le datagram socket consiste nell'implementazione di un programma che effettua i seguenti passi fondamentali:
- **Client:**
- (a) crea socket; (b) manda richiesta sulla socket con indirizzo, porta e messaggio; (c) riceve dati dalla socket; (d) chiude la socket.
- **Server:**
- (a) crea socket, per ascoltare richieste in arrivo; (b) riceve dati dalla socket; (c) invia dati sulla socket al client che ne ha fatto richiesta; (d) chiude la socket.



Socket a datagrammi (Datagram)

- La classe Java DatagramSocket usa il protocollo UDP per trasmettere dati attraverso le socket. Essa permette ad un client di connettersi ad una determinata porta di un host per leggere e scrivere dati impacchettati attraverso la classe DatagramPacket.
- La comunicazione è:
 - **Bidirezionale**
 - **Non affidabile**
- con ordine di consegna dei messaggi non garantito



Socket a datagrammi (Datagram)

- Nel caso delle socket DATAGRAM, il protocollo di comunicazione sottostante è il protocollo UDP (+ IP)
- UDP è un protocollo di trasporto, livello 4 OSI e fornisce l'astrazione porta
- IP è un protocollo di rete, livello 3 OSI e identifica univocamente ogni macchina collegata alla rete Internet
- La comunicazione tra Client e Server tramite datagram segue uno schema **simmetrico**.
- Questa considerazione ha portato al progetto di un unico tipo di socket per il Client e per il Server

Socket a datagrammi (Datagram)

- La classe Java con cui sono rappresentati i pacchetti UDP da inviare e ricevere sulle socket di tipo datagram è la classe
- DatagramPacket.
- Un oggetto istanza di DatagramPacket si costruisce inserendo nella chiamata al suo costruttore (**lato client**):
 1. il contenuto del messaggio (i primi length bytes dell'array buf)
 2. l'indirizzo IP del destinatario
 3. il numero di porta su cui il destinatario è in ascolto
- Se il pacchetto deve essere ricevuto (**lato server**) basta definire una istanza di DatagramPacket per il contenuto:

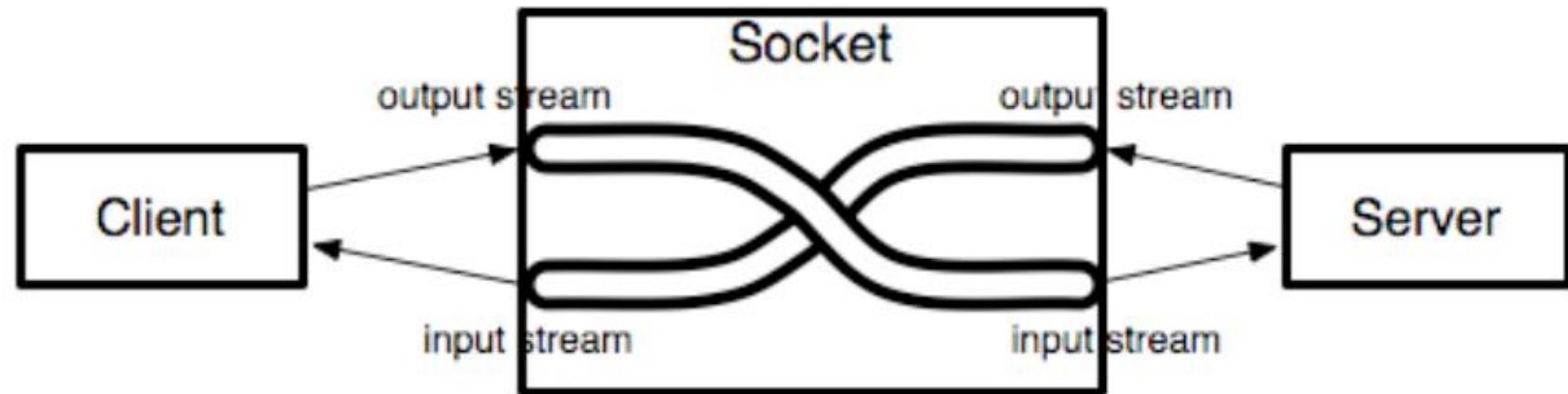


Stream socket

- **Stream socket** forniscono stream di dati affidabili ed ordinati. Nel dominio Internet sono supportati dal protocollo TCP (Transfer Control Protocol).
- Permettono di creare una connessione tra client e server con 2 canali (uno per ciascuna direzione).
- Le classi Java che usano questo protocollo sono Socket e ServerSocket.
- Le socket STREAM sono i terminali di un canale di comunicazione virtuale con connessione creato tra il Client e il Server. La comunicazione avviene in modo bidirezionale e affidabile.

Stream socket

- Client e server instaurano una connessione che consiste di due flussi di dati, uno per l'input ed uno per l'output.





Stream socket

- Le API invocate da client e server, per instaurare una connessione, sono diverse e quindi esistono classi di libreria Java diverse.
- Dal lato client vi è il processo che vuole instaurare una connessione per inviare e ricevere dati. La classe Java che usa il client è Socket.
- Dal lato server vi è un processo che attende una richiesta di connessione, la classe Java che usa è ServerSocket.
- La connessione è individuata da 4 elementi:
 - indirizzo IP del client,
 - porta del client,
 - indirizzo IP del server,
 - porta del server.



Client di uno stream socket

- La classe Socket consente di creare una socket con connessione, stream (TCP) per il collegamento di un Client a un Server. Socket attiva
- La creazione della socket produce in modo atomico anche la connessione al server corrispondente o lancia l'opportuna eccezione.
- La creazione di una socket a stream se va a buon fine produce una connessione bidirezionale(stream) tra i due processi interagenti e impegna risorse tra i nodi e tra i processi La connessione permette poi una comunicazione bidirezionale (full duplex).



Server di uno stream socket

- La classe `ServerSocket` definisce una socket capace di accettare richieste di connessione provenienti dai Client
- *Socket passiva*
- Consente di gestire:
 - più richieste di connessione pendenti allo stesso tempo (definisce anche la lunghezza della coda in cui vengono messe le richieste di connessione non ancora accettate dal server)
 - più connessioni aperte contemporaneamente
- Una volta stabilita la connessione (ottenuta dal server tramite il metodo `accept`) la trasmissione dei dati avviene attraverso un normale oggetto `Socket` del server (restituito dalla `accept`)