Università degli Studi della Calabria

Corso di Laurea in Ingegneria Gestionale

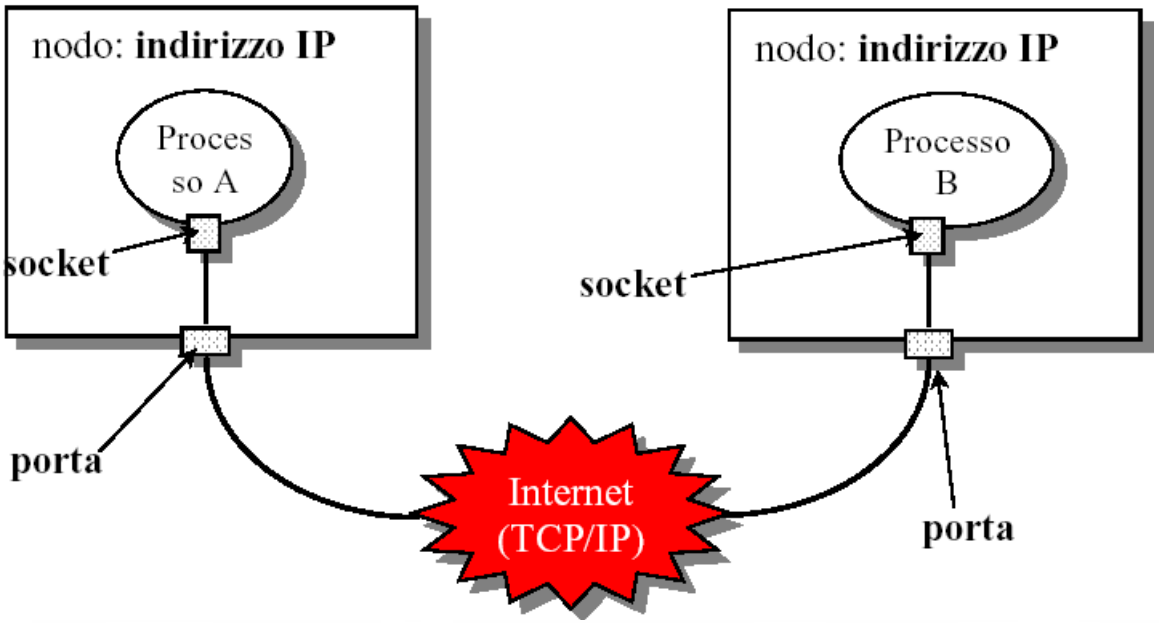
A.A. 2007/2008

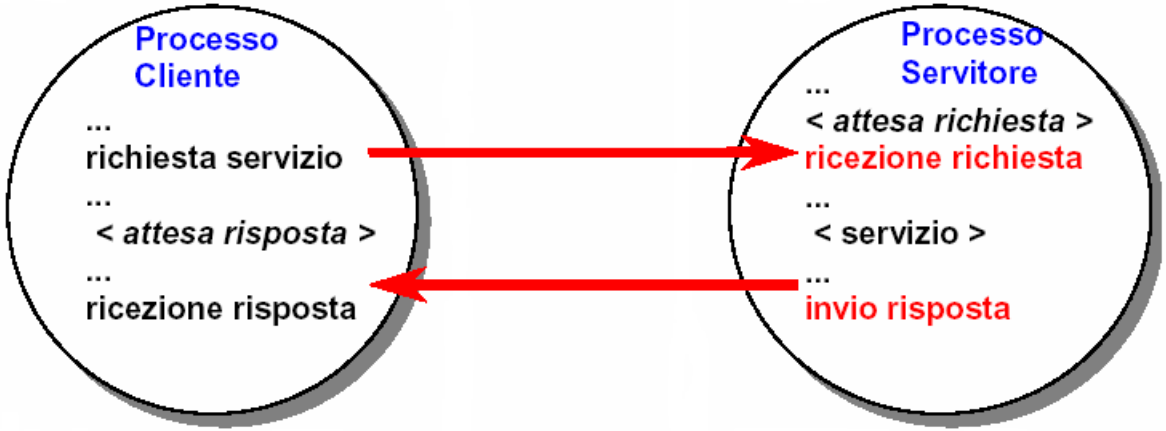
Corso di Sistemi di Elaborazione in Rete

Lucidi delle Esercitazioni

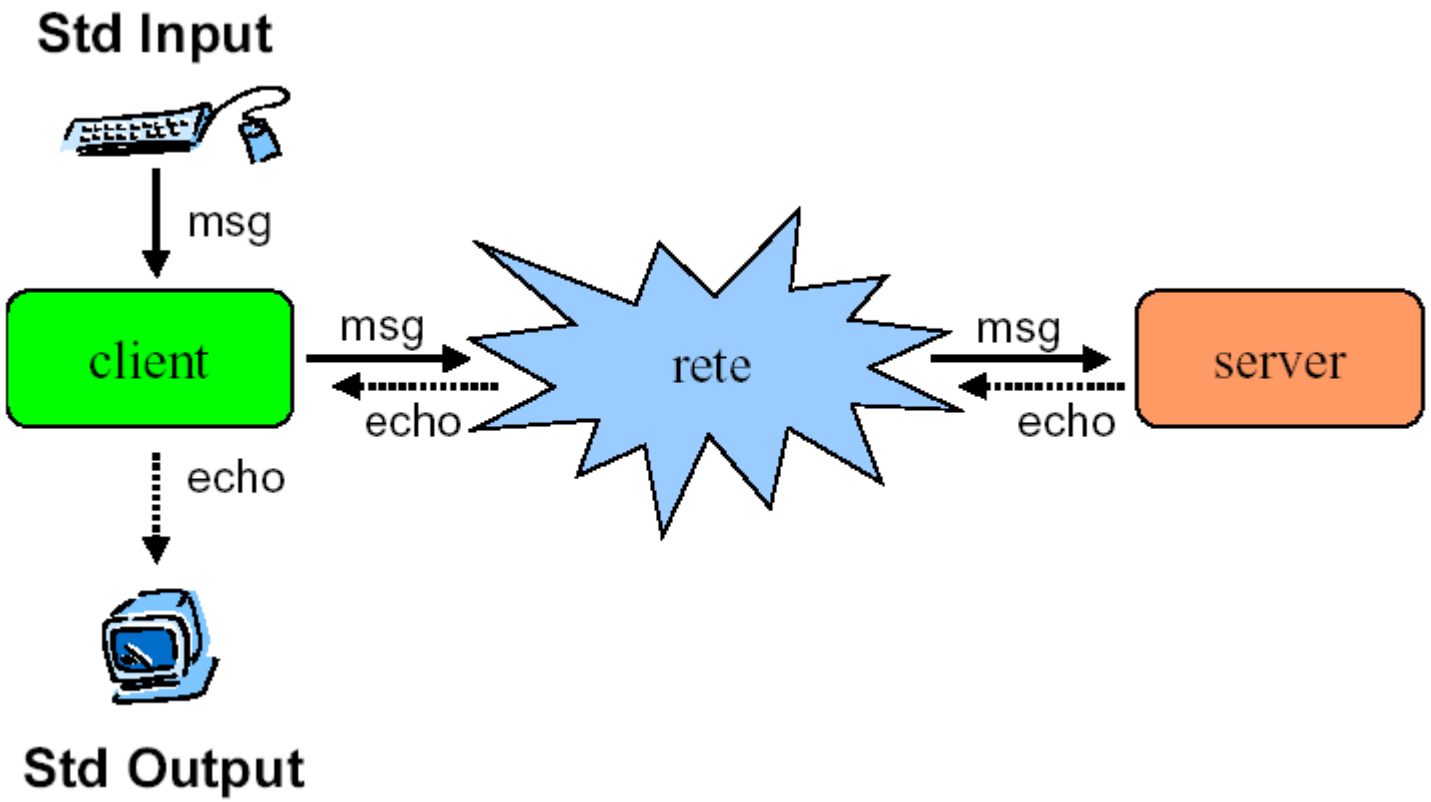
Agostino Forestiero

Socket TCP


nodo: indirizzo IP
Proces so A
socket
porta
nodo: indirizzo IP
Processo B
socket
porta
Internet (TCP/IP)


Processo Cliente
...
richiesta servizio
...
< attesa risposta >
...
ricezione risposta
Processo Servitore
...
< attesa richiesta >
ricezione richiesta
...
< servizio >
...
invio risposta

Echo Application



Echo Client (1)

```
import java.net.*;
import java.io.*;

public class EchoClient {
    public static void main(String[] args) throws IOException {

      /* Lanciando il programma senza argomenti si ottiene il local loopback IP address,
         per testarlo in locale (client e server sulla stessa macchina), altrimenti
         si possono passare da linea di comando l'indirizzo o il nome della macchina remota */

      InetAddress addr;
      if (args.length == 0) addr = InetAddress.getByName(null);
      else addr = InetAddress.getByName(args[0]);
      Socket socket=null;
      BufferedReader in=null, stdIn=null;
      PrintWriter out=null;
      try {
    // creazione socket
    socket = new Socket(addr, EchoServer.PORT);
    System.out.println("EchoClient: started");
    System.out.println("Client Socket: "+ socket);
```

Echo Client (2)

```
// creazione stream di input da socket
    InputStreamReader isr = new InputStreamReader( socket.getInputStream());
    in = new BufferedReader(isr);

// creazione stream di output su socket
    OutputStreamWriter osw = new OutputStreamWriter(socket.getOutputStream());
    BufferedWriter bw = new BufferedWriter(osw);
    out = new PrintWriter(bw, true);

// creazione stream di input da tastiera
    stdIn = new BufferedReader(new InputStreamReader(System.in));
    String userInput;
```

Echo Client (3)

```
// ciclo di lettura da tastiera, invio al server e stampa risposta
    while (true){
            userInput = stdIn.readLine();
            out.println(userInput);
            if (userInput.equals("END")) break;
            System.out.println("Echo: " + in.readLine());
     }
  }catch (UnknownHostException e) {
    System.err.println("Don't know about host "+ addr);
    System.exit(1);
  } catch (IOException e) {
      System.err.println("Couldn't get I/O for the connection to: " + addr);
      System.exit(1);
  }
  System.out.println("EchoClient: closing...");
  out.close();
  in.close();
  stdIn.close();
  socket.close();
  }
} //EchoClient
```

Echo Server (1)

```
public class EchoServer {
public static final int PORT = 1050; // porta al di fuori del range 1-1024 !

public static void main(String[] args) throws IOException {
  ServerSocket serverSocket = new ServerSocket(PORT);
  System.out.println("EchoServer: started ");
  System.out.println("Server Socket: " + serverSocket);
  Socket clientSocket=null;
  BufferedReader in=null;
  PrintWriter out=null;
  try {
   // bloccante finchè non avviene una connessione
   clientSocket = serverSocket.accept();
   System.out.println("Connection accepted: "+ clientSocket);
   // creazione stream di input da clientSocket
    InputStreamReader isr = new InputStreamReader(clientSocket.getInputStream());
    in = new BufferedReader(isr);
   // creazione stream di output su clientSocket
    OutputStreamWriter osw = new OutputStreamWriter(clientSocket.getOutputStream());
    BufferedWriter bw = new BufferedWriter(osw);
    out = new PrintWriter(bw, true);
```

Echo Server (2)

```
//ciclo di ricezione dal client e invio di risposta
  while (true) {
    String str = in.readLine();
    if (str.equals("END")) break;
    System.out.println("Echoing: " + str);
    out.println(str);
  }
 }
 catch (IOException e) {
  System.err.println("Accept failed");
  System.exit(1);
 }
 // chiusura di stream e socket
 System.out.println("EchoServer: closing...");
 out.close();
 in.close();
 clientSocket.close();
 serverSocket.close();
 }
} // EchoServer
```