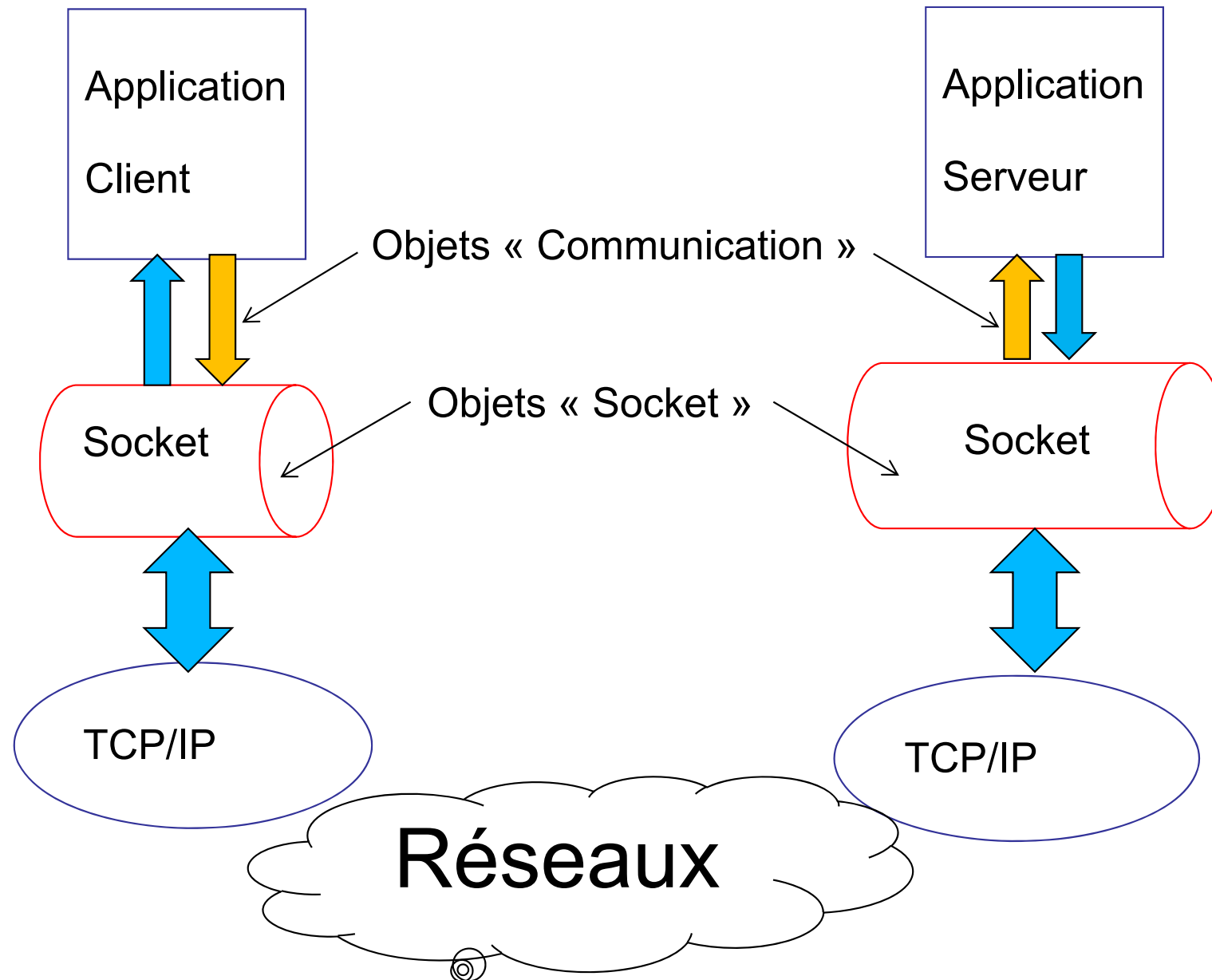


Interface Socket - Java

Interface Socket – En Java

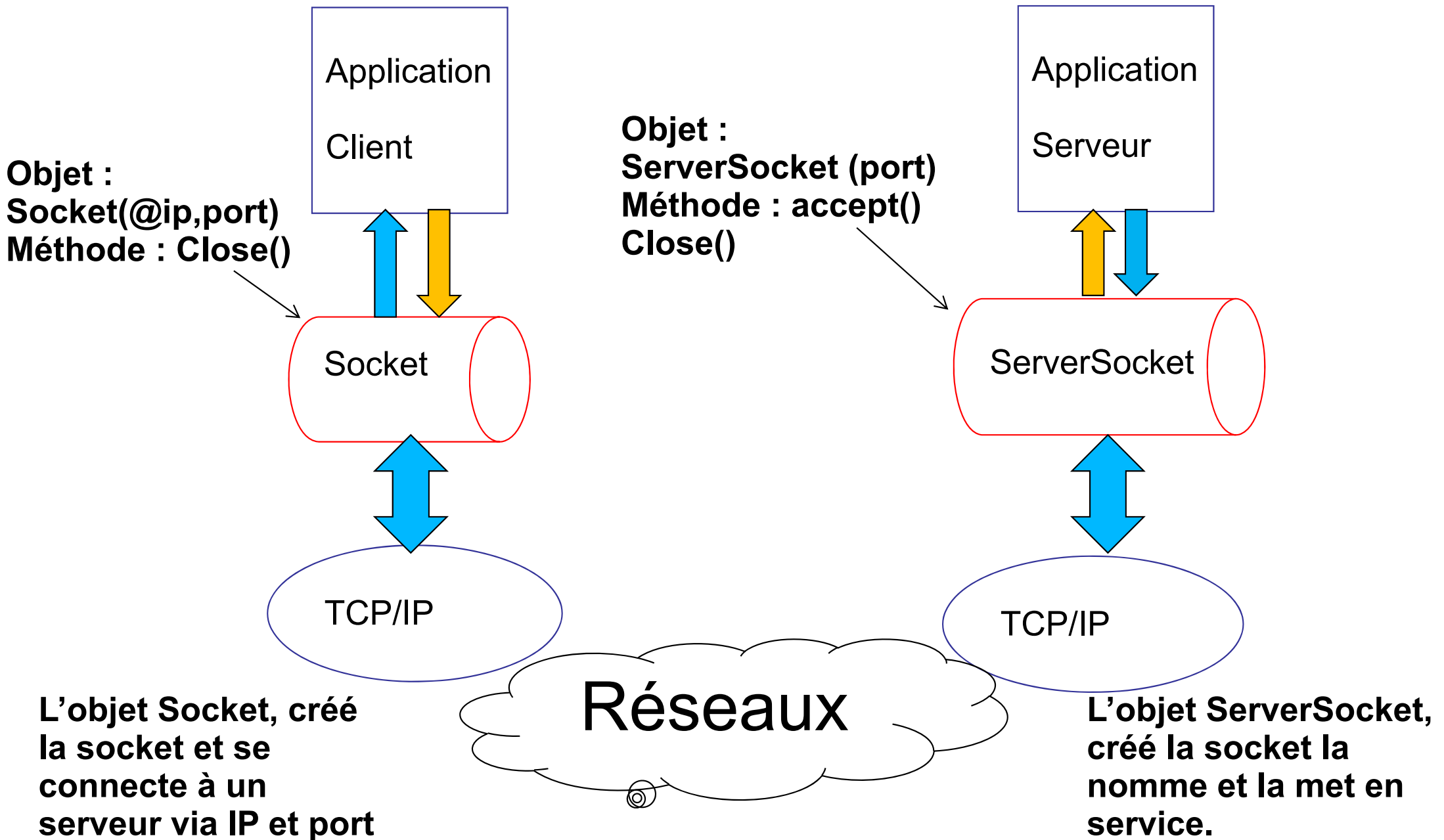


Interface Socket – En Java

Le bibliothèques :

Import java.net.*	(interface réseau)
Import java.net.InetAddress;	(gestion des adresses IP)
Import java.net.ServerSocket;	(socket coté serveur)
Import java.net.Socket;	(socket coté client)
Import java.net.UnknownHostException;	(gestion des erreurs)
...	

Interface Socket – En Java

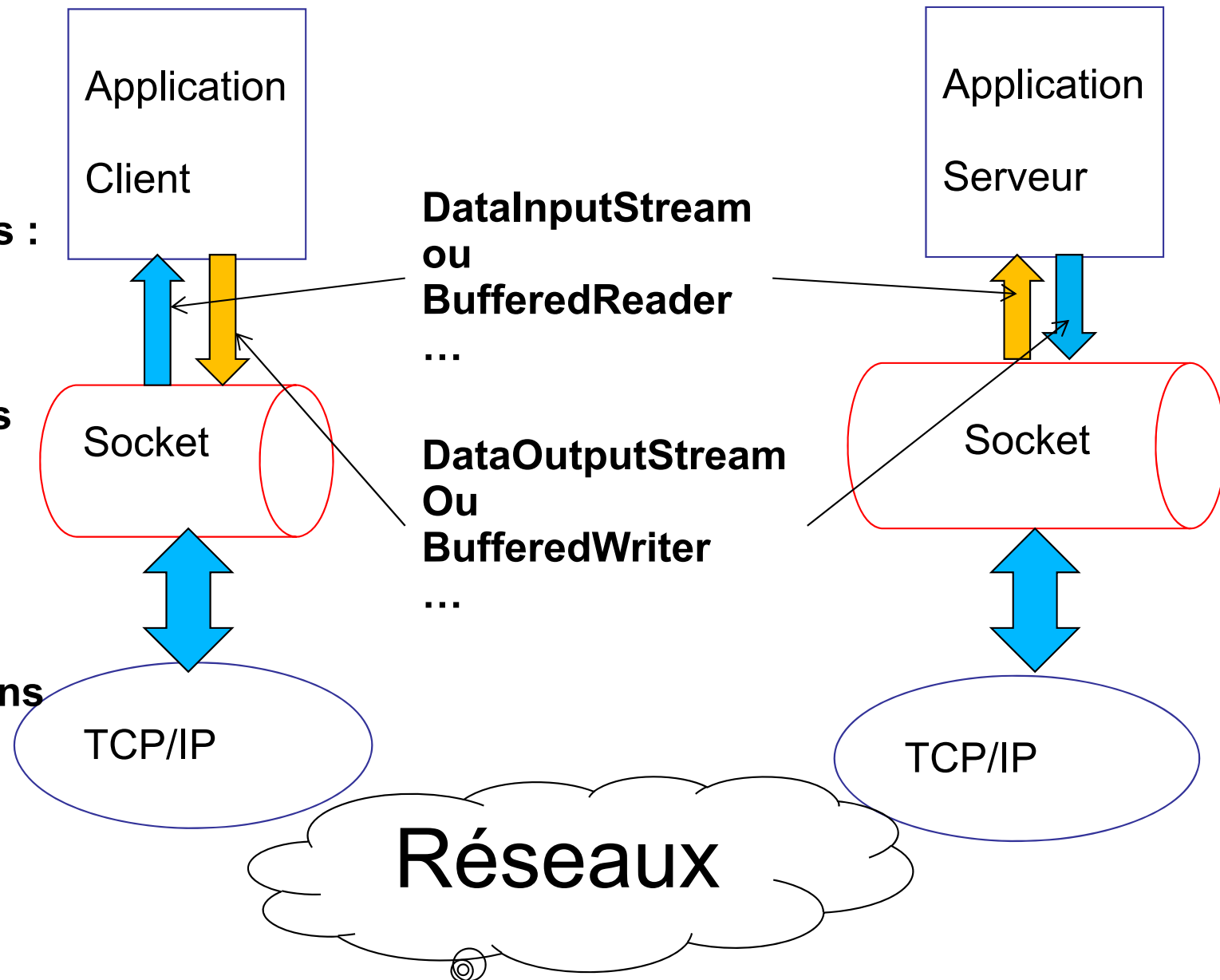


Interface Socket – En Java

Les objets Socket() possèdent deux méthodes :
getInputStream()
getOutputStream()

Que l'on va associer à des objets d'E/S :
DataInputStream
DataOutputStream

Des méthodes :
read() et write() permettent l'utilisation des sockets



Interface Socket – En Java - Résumé

ServerSocket // coté serveur

/** socket du serveur pour les requêtes clients, on précise en paramètre le numéro de port et le serveur se lance et attend une connexion client ***/**

Les méthodes : `accept()` , `close()`,

Socket // coté client

/** socket pour communiquer avec un serveur , on précise l'adresse IP et le numéro de port et la connexion s'établit si le serveur est actif ***/**

Les méthodes : `close()`

DataInputStream (`sock.getInputStream()`)

ou BufferedReader (`InputStreamReader (sock.getInputStream())`)

/** réception des informations du client ***/**

Les méthodes : `read()`, `readLine()`

DataOutputStream (`sock.getOutputStream()`)

ou BufferedWriter (`OutputStreamWriter (soc.getOutputStream())`)

/** envoi d'informations vers le client ***/**

Les méthodes : `write()`, `writeLine()`

Comparaison C - Java

Serveur C

```
main()
{ struct sockaddr_in local;
  struct sockaddr_in distant;

  local.sin_family = AF_INET;
  local.sin_port = htons(PORT);
  local.sin_addr.s_addr = INADDR_ANY;
  bzero(&(local.sin_zero),8);

  lg = sizeof(struct sockaddr_in);
  sock=socket(AF_INET, SOCK_STREAM,0);
  bind(sock, (struct sockaddr *)&local, lg);
  listen(sock, 5);

  while(1)
  {  socket2=accept(sock, (struct sockaddr *)...)

    read(socket2,mess,80);
    write(socket2,mess,80);
  }
  close(socket2);
}
```

Serveur java

```
...
sockserv = new ServerSocket (Port);

while (true)
{ Socket sockcli = sockserv.accept();
  in = new DataInputStream ((sockcli..
  out = new DataOutputStream (sockcli.
in.lecture(mess,0,80);
out.ecriture(mess);
  sockcli.close();
}
```

Comparison C - Java

Client Java

...

```
sockcli = new Socket (url, Port);  
in = new DataInputStream (sockcli..;  
out = new DataOutputStream (sockcli..;
```

```
while(boucle)  
{ out.ecriture(mess);  
  mess=in.lecture(in,0,80);  
}  
sockcli.close();
```

Client C

```
main()  
{ int port,sock;  
  struct sockaddr_in  serv_addr;  
  struct hostent      *server;  
  
  server = gethostbyname(SERV);  
  sock = socket(AF_INET, SOCK_STREAM, 0);  
  
  bzero(&serv_addr, sizeof(serv_addr));  
  serv_addr.sin_family = AF_INET;  
  bcopy(server->h_addr, &..);  
  serv_addr.sin_port = htons(port);  
  connect(sock, (struct sockaddr ..;  
  
  while (strcmp(mess,"fin",3)!=0)  
      { write(sock,mess,80);  
        read (sock,mess,80);  
      }  
  close (sock);  
}
```


Interface Socket – En Java

Coté serveur – Exemple de code

```
ServerSocket sockserv=null;  
DataInputStream in ;  
DataOutputStream out;  
  
sockserv = new ServerSocket (Port); //création socket serveur  
Socket sockcli = sockserv.accept(); //attente requête client  
  
// creation des flux de communication  
in = new DataInputStream (sockcli.getInputStream());  
out = new DataOutputStream (sockcli.getOutputStream());  
  
// reception et emission message  
in.read(message,0,80);  
out.write("Recu :"+message);  
  
sockserv.close();      // fermeture port de communication
```

Interface Socket – En Java

Coté client – Exemple de code

```
Socket sockcli = null;  
DataInputStream in;  
DataOutputStream out;  
  
sockcli = new Socket ("127.0.0.1" , Port);           //connexion au serveur  
  
// creation des objets de communication  
in = new DataInputStream (sockcli.getInputStream());  
out = new DataOutputStream (sockcli.getOutputStream());  
  
// emission et réception message  
out.write("coucou");  
in.read(message,0,80);  
  
sockcli.close(); // fermeture socket
```