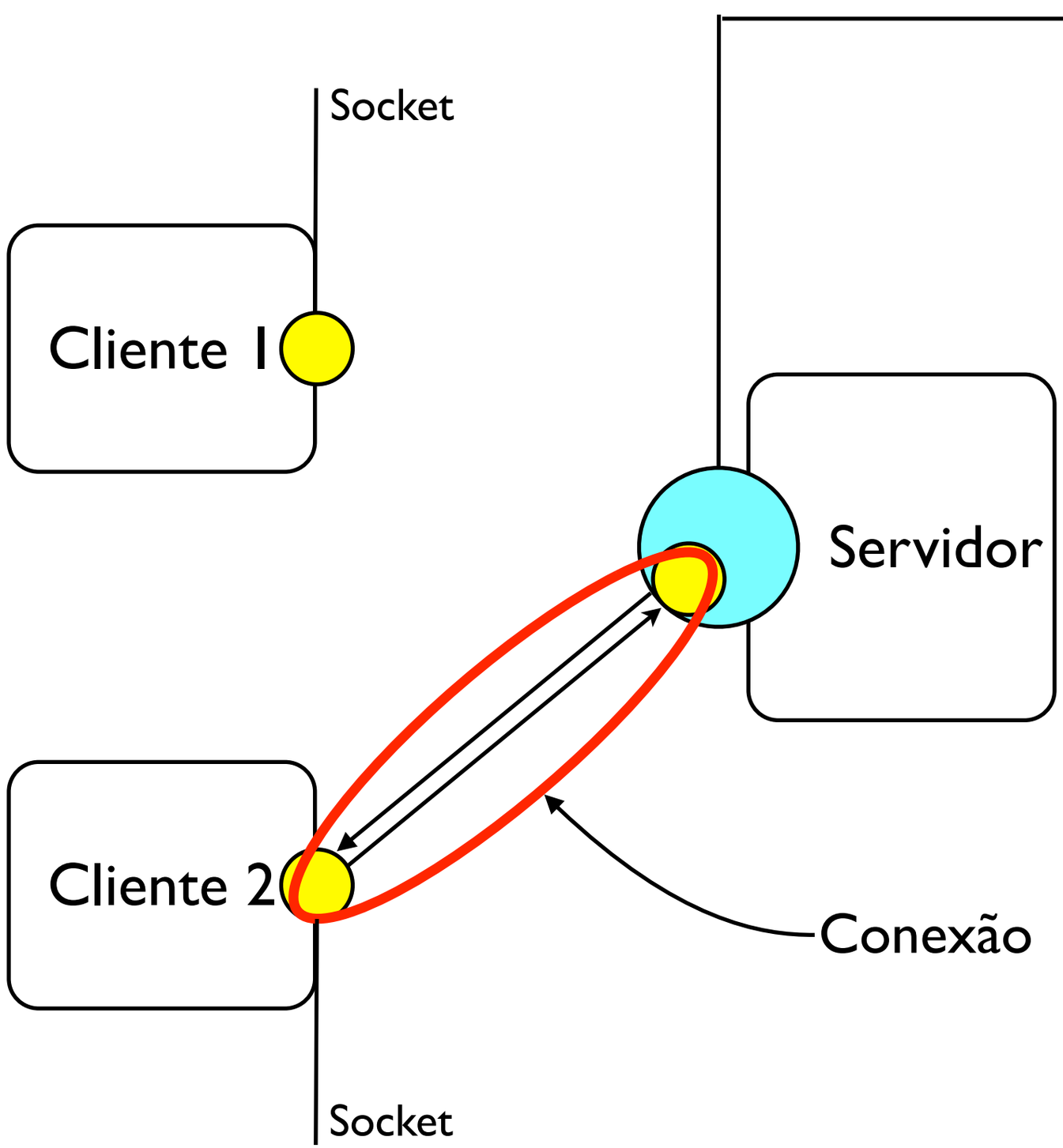
Sistemas Distribuídos

Universidade do Minho

2011/2012



Server Socket TCP/IP:

- endereço (ip);
- porto 16 bits-> distinguem serviços na mesma máquina;
  - 0–1023 são standard ex: http 80
  - 1024–49151
  - 49152-65535 dinâmicos

- Servidor fica à espera de ligações;
- quando o cliente se liga é estabelecida uma conexão, bidireccional;
- Socket representa um extremo de uma conexão.

Aula 7: Socket JAVA

Cliente

Esqueleto:

socket()
connect()

JAVA:

```
Socket socket = new Socket(remotehost,port);
```

while ()

write()
read()

```
out.write(...);
out.flush();
```

close()

```
socket.shutdownInput();
socket.shutdownOutput();
socket.close();
```

Servidor

Esqueleto:

socket()
bind()

JAVA:

```
ServerSocket sSock = new
ServerSocket(porto);
```

listen()
while ()
accept()

```
while (true){ //para aceitar
conexões indefinidamente
Socket sock = sSock.accept()
//fica à escuta e bloqueia até que
uma conexão seja estabelecida
BufferedReader in=new BufferedReader(new
InputStreamReader(sock.getInputStream()))
BufferedWriter out = new
BufferedWriter(new
OutputStreamWriter(sock.getOutputStream()
))
...
while(...){
```

while ()

read()
write()

```
in.readLine();
out.write(...);
out.flush();
}
```

close()

```
sock.shutdownInput();
sock.shutdownOutput();
sock.close();
```

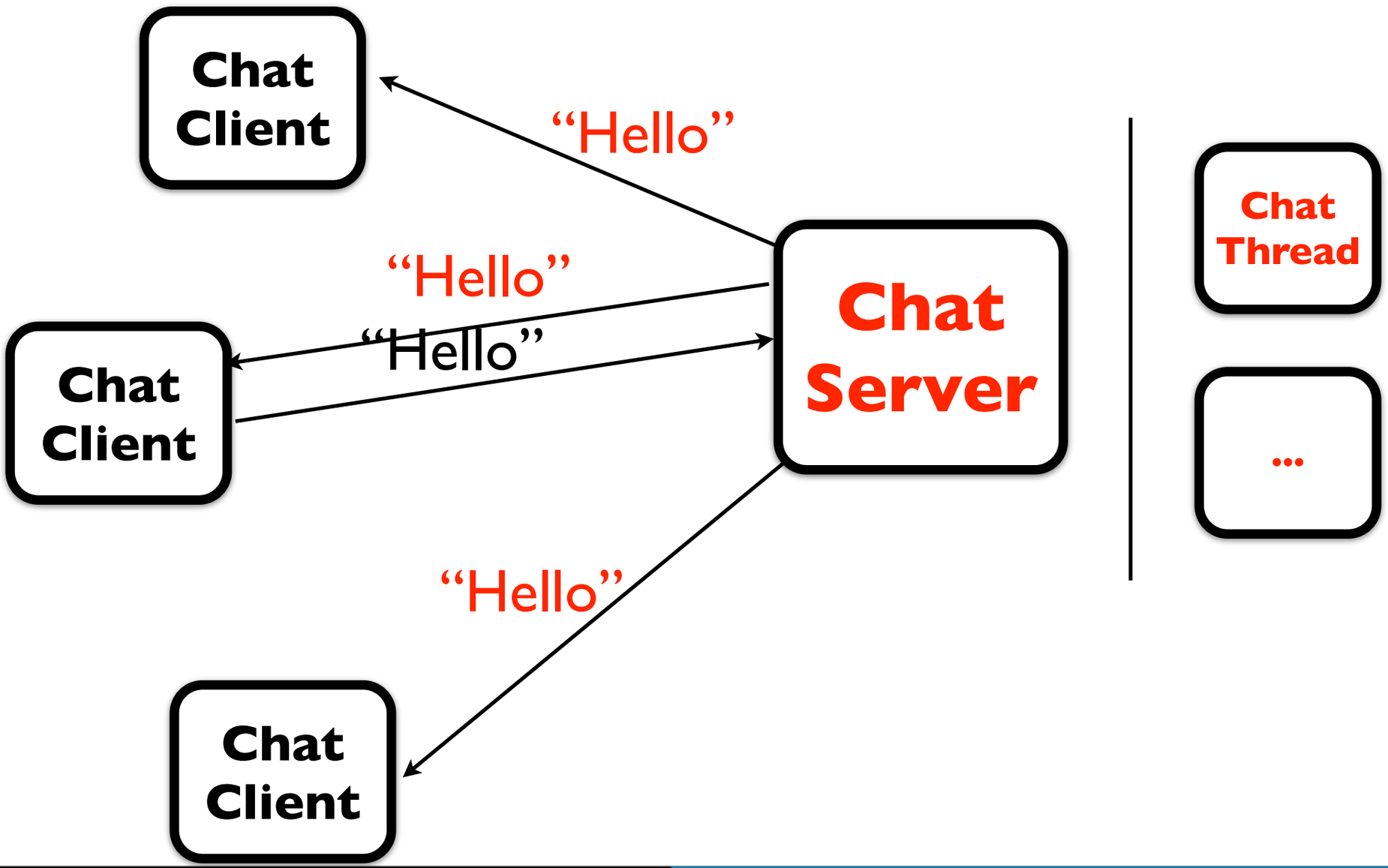
Aula 9: ex 1

- Reimplemente o servidor de Banco do exercício anterior de modo a que este aceite a conexão simultânea de múltiplos clientes.

- Quem não tiver acabado o servidor Banco da aula anterior pode fazer download do código aqui:

http://dl.dropbox.com/u/2534164/SD%20Banco.zip

- Implemente um servidor de conversação que aceite a conexão de múltiplos clientes. Cada mensagem enviada por um cliente é difundida por todos os clientes ligados.



Aula 9: ex 3

- Acrescente a funcionalidade de registo de nicks e do envio de mensagens privadas ao serviço de conversação desenvolvido no exercício anterior.