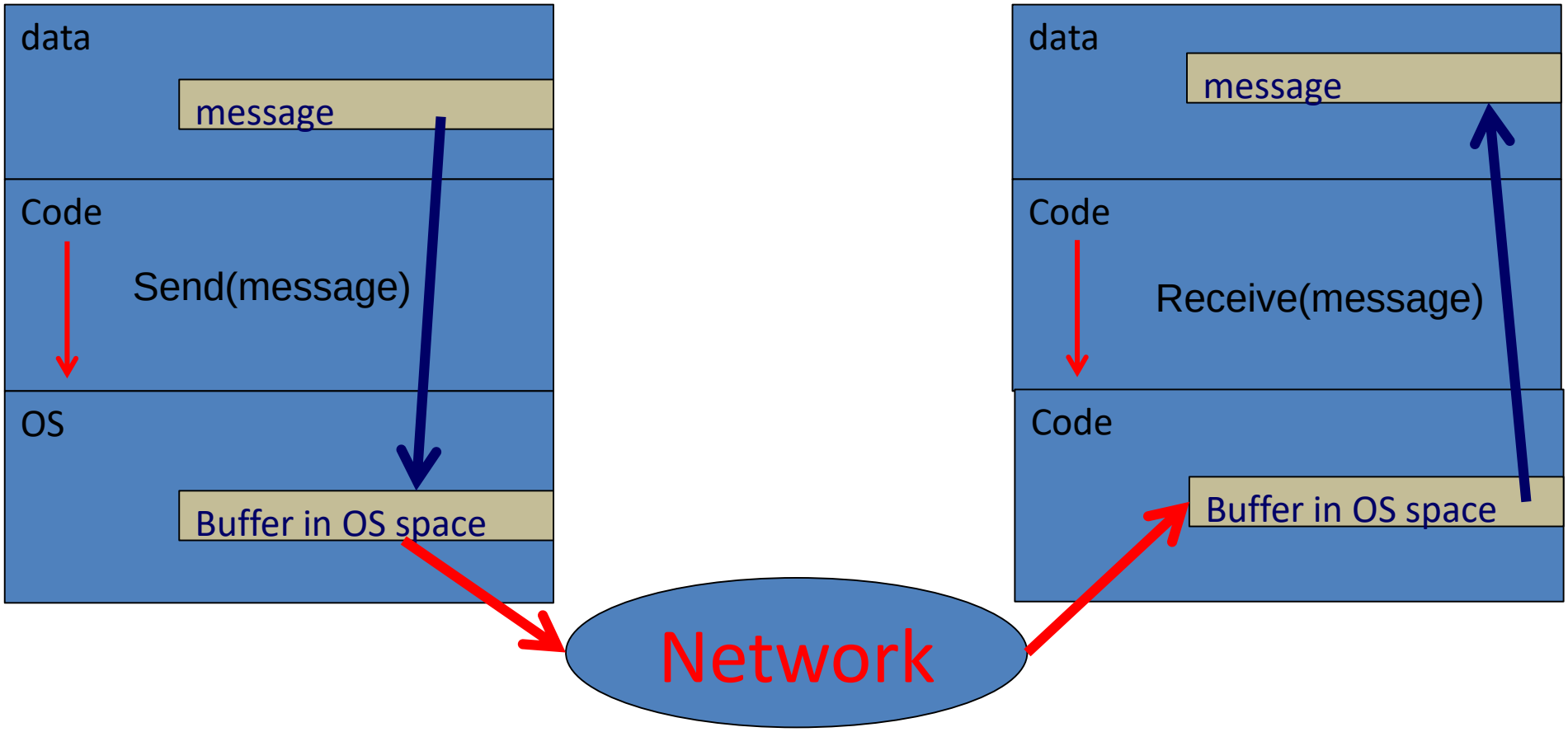
FSO
05 de Dezembro 2018

- Comunicação por mensagens
  - Redes de computadores e seus protocolos
  - Comunicação usando *socket streams* (TCP/IP)
  - Interação servidor – cliente: servidores sequenciais e concorrentes

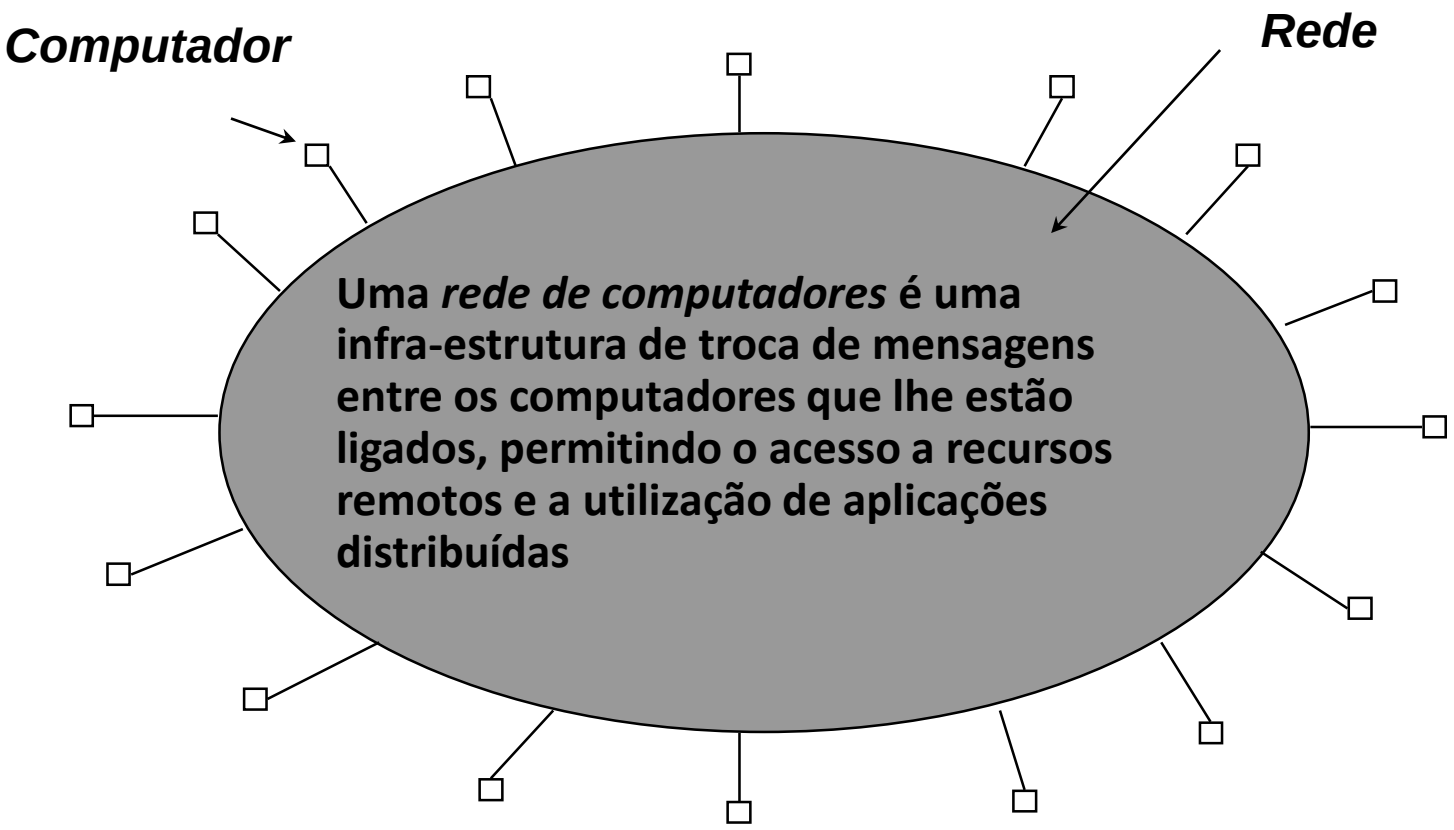
Comunicação entre processos em máquinas distintas



- O SO pode enviar e receber bytes através da rede
- O emissor tem de especificar: o endereço da máquina onde os bytes são entregues, e uma port / mailbox na máquina remota

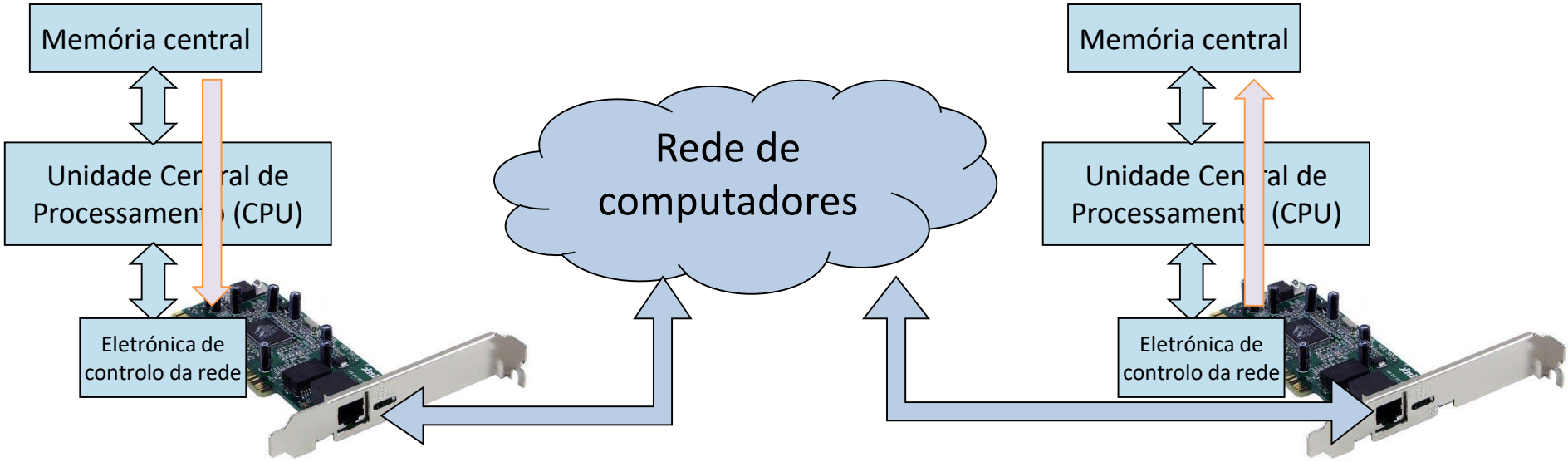
Rede de computadores

Um conjunto de computadores que estão interligados com o objetivo de trocar informação e partilhar recursos.



Numa transferência intervêm dois computadores e a rede

- Dois computadores
  - Nó emissor: produz uma sequência de bytes (mensagem).
  - Nó recetor ou destinatário: recebe essa sequência de bytes.
- Rede
  - Meios de interligação: cabos, atmosfera, …
  - Equipamentos dedicados a assegurar que a mensagem é transportada do nó emissor ao nó recetor.



Internet e o encaminhamento de pacotes

- A Internet é uma interligação de redes locais de acordo com normas próprias.
- Na Internet todos os computadores (ou nós) têm um endereço único (normalmente um número com 32 bits) chamado endereço IP (Internet Protocol).
- Quando uma computador emite um pacote destinado a outro, noutra LAN, entrega-o ao *router* mais próximo (que está na sua rede local).
- Os *routers* propagam o pacote até este chegar ao *router* da rede local do computador de destino, que entrega o pacote ao destinatário.

Endereços dos nós

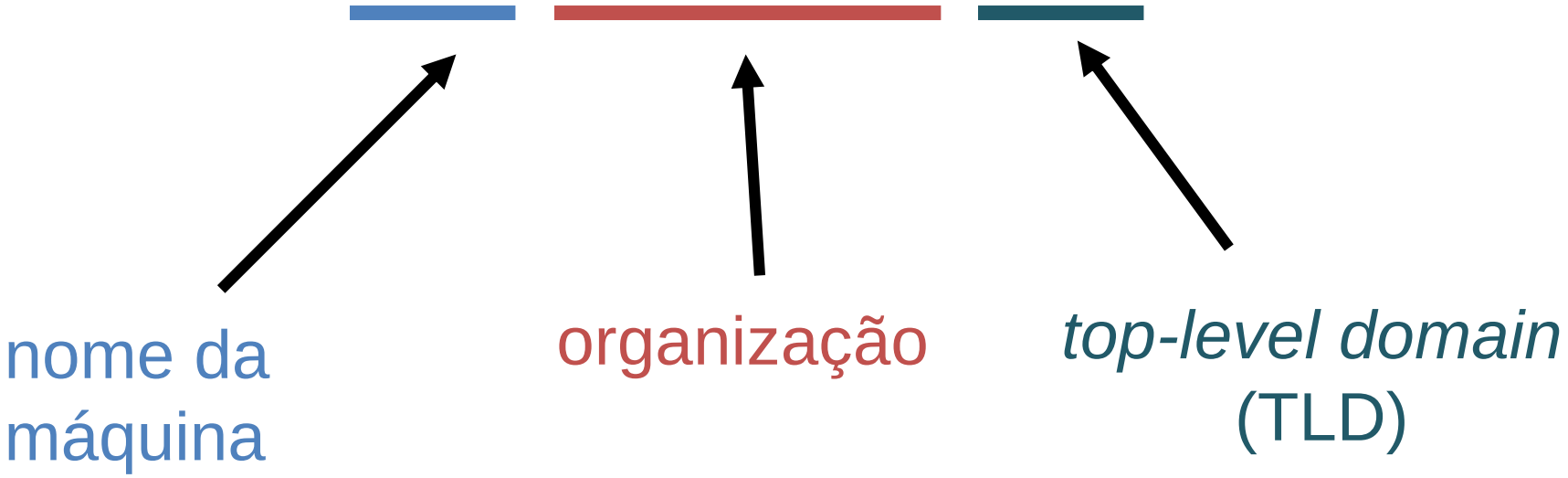
Endereço IP do nó (numérico / *machine friendly*)

- Constituído por 4 bytes (32 bits)
  - representado por 4 números entre 0 e 255, separados por pontos.
- Identifica univocamente o computador na Internet.
- Usado no encaminhamento das mensagens na rede.
- **Exemplo:** 193.136.122.33

Endereço simbólico do nó (cadeia de caracteres / *user friendly*)

- São aqueles nomes que o utilizador usa.
- **Exemplo:** www.google.com

Nome do nó
asc.di.fct.unl.pt



Domain Name System (DNS)

- Serviço de conversão de nomes “user friendly” em nomes “machine friendly”.
  - Este serviço reside num nó da LAN (ou do fornecedor de Internet).
- Invocado quando um nó tenta obter o endereço IP correspondente a um dado nome simbólico.

TLDs do Domain Name System [1]

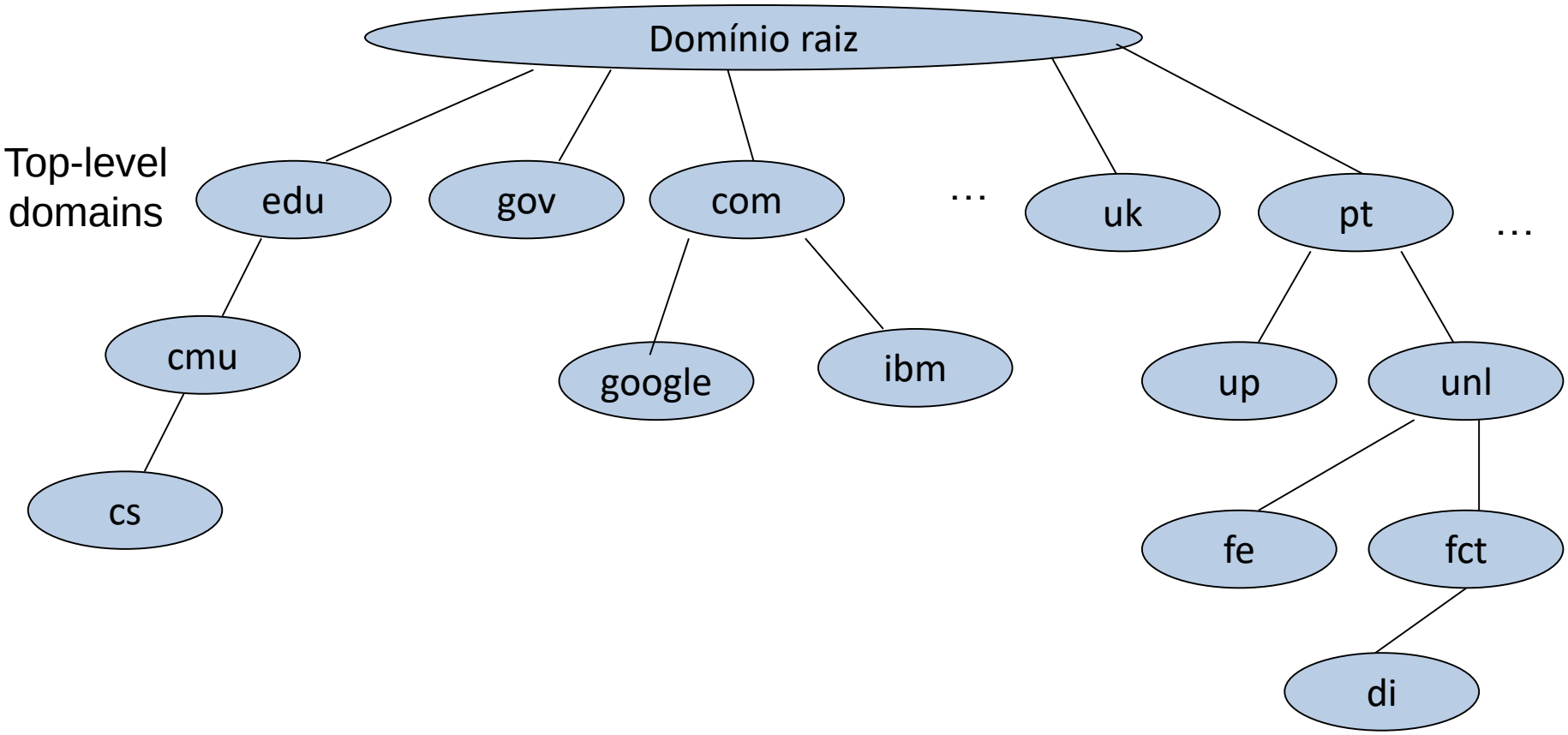
Top-Level Domain	Utilização
biz	Negócios
com	Comercial (EUA)
edu	Educação (EUA)
info	Informação
gov	Governo (EUA)
mil	Militar (EUA)
net	Rede
org	Sem fins lucrativos

TLDs do Domain Name System [2]

As organizações com sede fora dos Estados Unidos usam um top-level domain que é o código do país (com 2 letras).

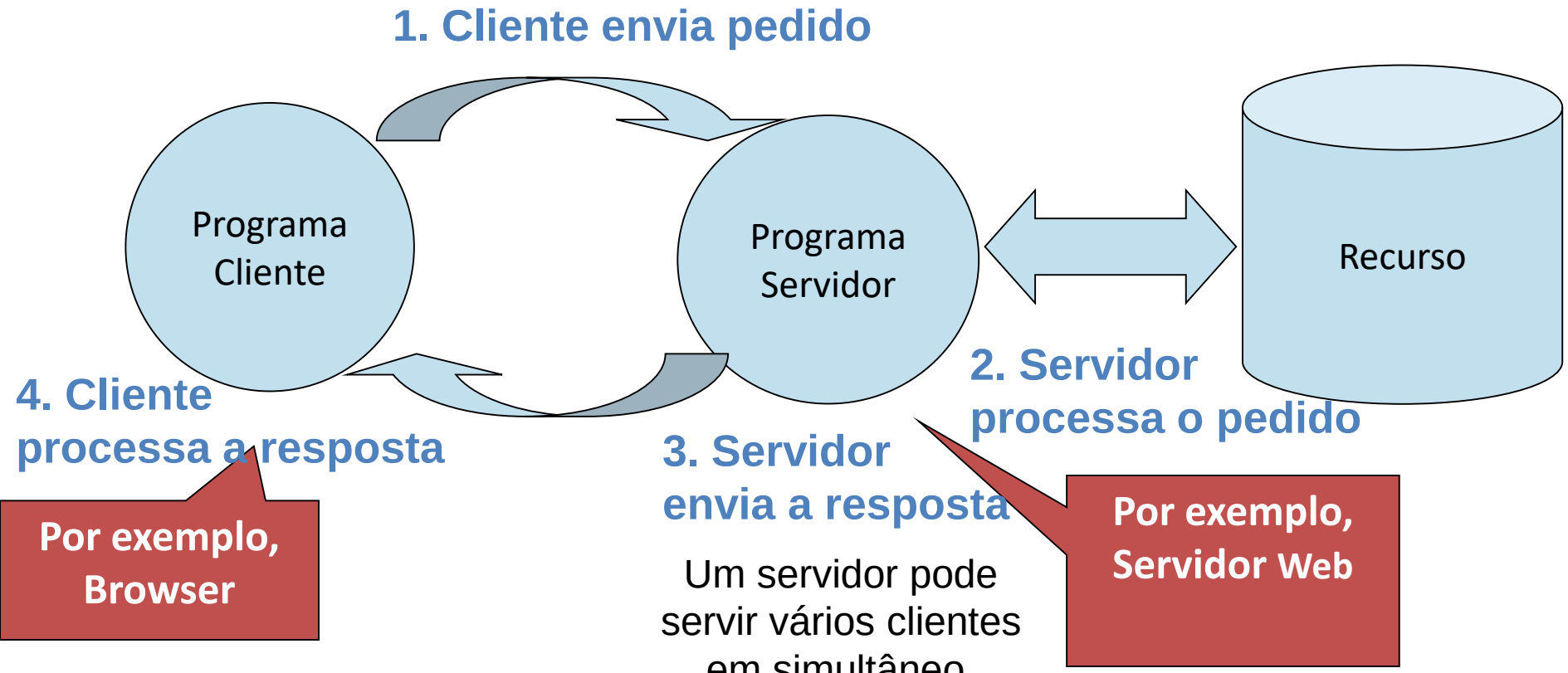
País	TLD
pt	Portugal
uk	Reino Unido
es	Espanha
fr	França
nz	Nova Zelândia
cn	China
…	…

Árvore de domínios do DNS



Relação cliente-servidor

- Na maioria das situações, as aplicações que usam a rede são constituídas por dois programas.



Exemplos de servidores

- Servidor Web
  - Recursos: ficheiros e programas.
  - Serviços: obter ficheiros e gerar páginas dinâmicas executando programas a pedido do cliente.
- Servidor de Mail
  - Recurso: ficheiros com mensagens.
  - Serviços: armazena mensagens ou encaminha-as para outros servidores de Mail.
- Servidor Dropbox
  - Recursos: ficheiros.
  - Serviços: leitura e escrita de ficheiros.

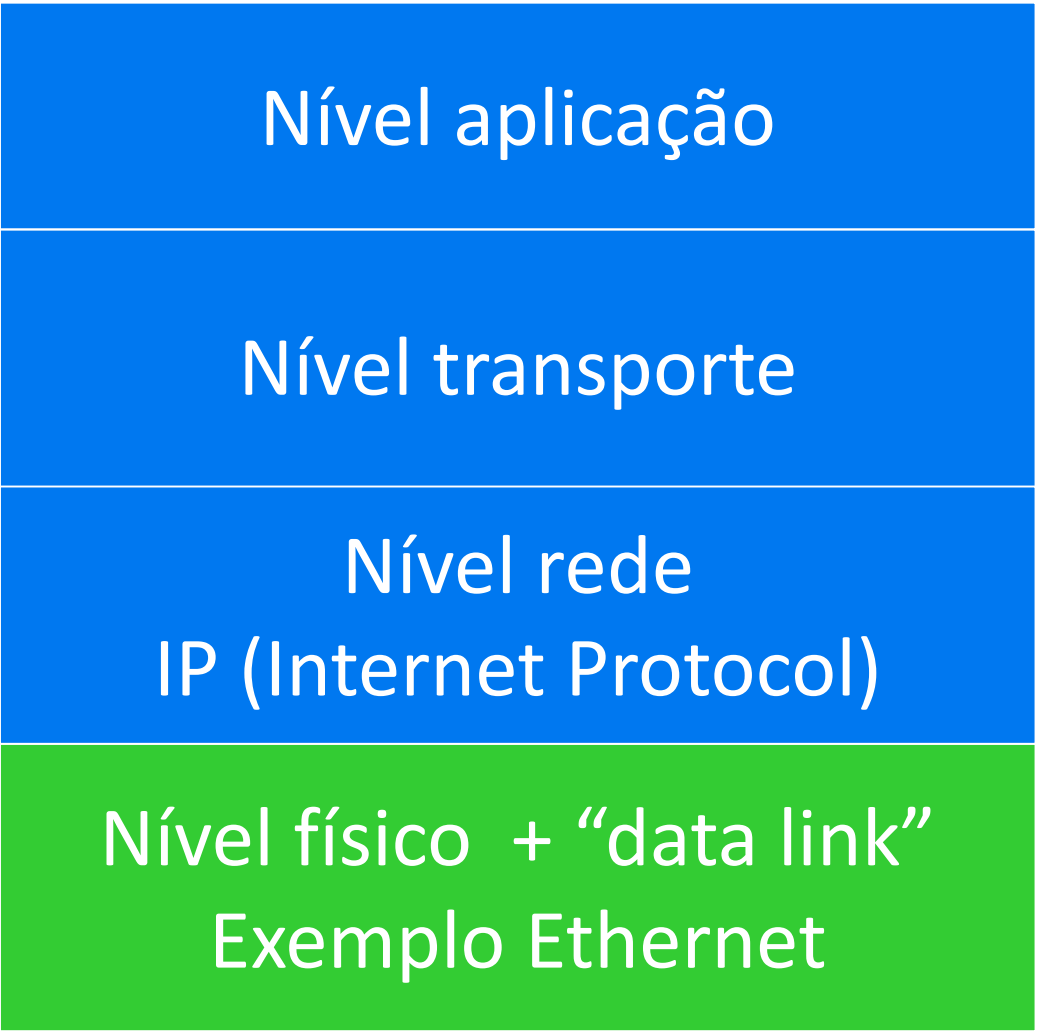
Exemplos de clientes

- Cliente Web
  - Firefox, Internet Explorer, Safari, etc.
- Ciente de Mail
  - eM Client, Mozilla Thunderbird, Opera, Mail, Live Mail, etc.

Protocolos de aplicação

- Conjunto de regras que definem como é que os bytes que circulam na rede são interpretados.
  - O cliente e o servidor têm de acordar previamente os formatos do pedido e da resposta;
  - Existem normas para imensos serviços na Internet.
  - **Exemplos:**
    - Protocolos Internet para consultar o DNS e obter o endereço IP para um nome;
    - Protocolos de aplicações de email (SMTP) ou de Web (HTTP).

A pilha de protocolos TCP/IP [1]

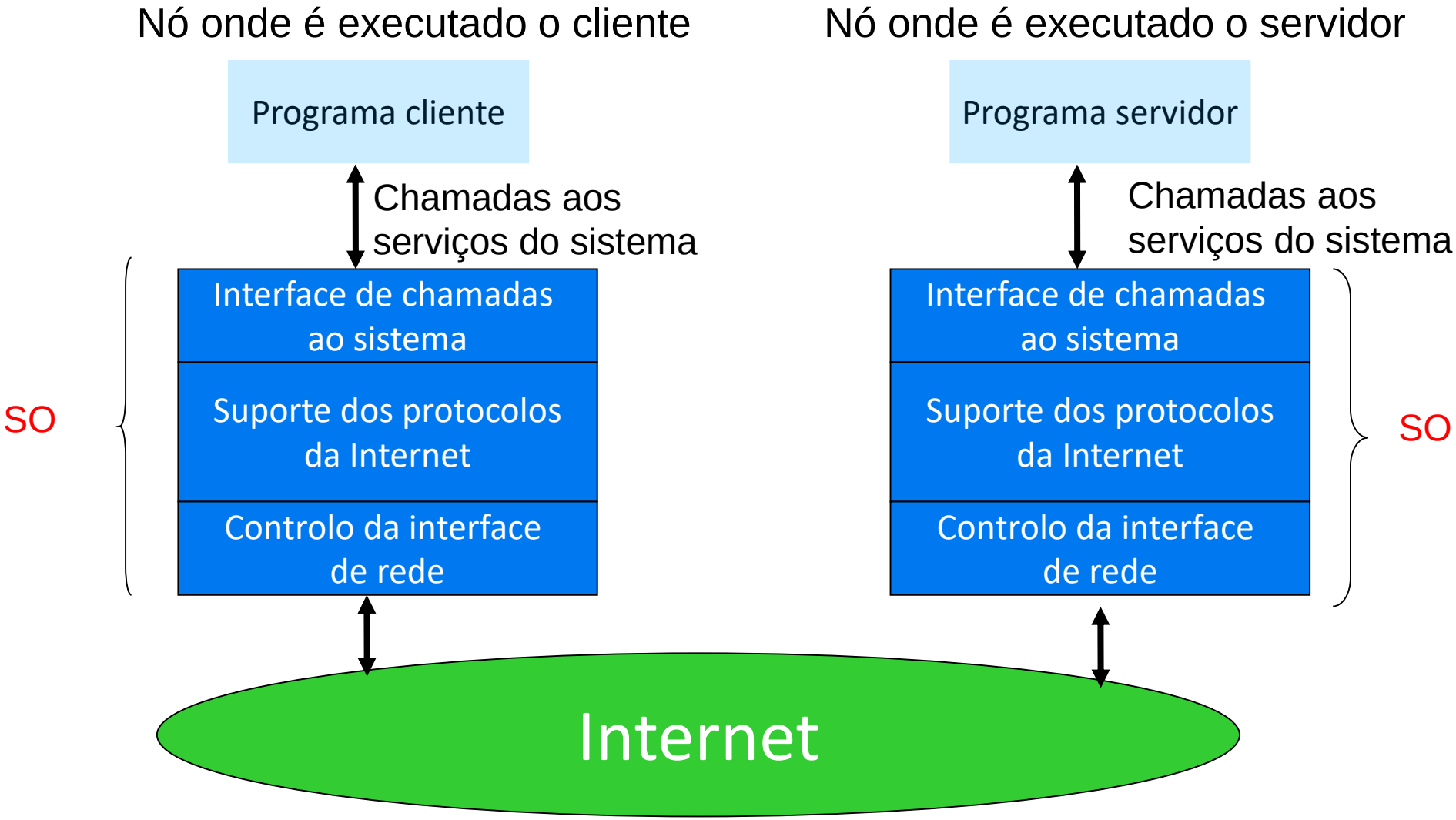


HTTP, FTP, SMTP, …
Cliente/servidor

UDP (Unreliable Datagram Protocol)
TCP (Transmission Contol Protocol)

Define a forma de designar os nós e o mecanismo de entrega de pacotes; não fiável

Aplicação cliente-servidor na Internet



O nível transporte

- A camada de *Transporte*:
  - Disponibiliza todos serviços necessários à comunicação "*end-to-end*" (e.g. entre dois *hosts / end-points* na periferia da rede) incluindo a noção de *porta (ver à frente)*
  - Se necessário, divide a mensagem que recebe do nível da aplicação (e.g. uma página html) no número de pacotes IP necessários.

A noção de porta nos protocolos de transporte TCP/IP

- Endereço de portos/portas *“Port Addresses” (port numbers)*
  - A camada de transporte usa-os para identificar a aplicação a quem é dirigida uma mensagem
  - Analogia: número do apartamento num prédio (num determinado endereço)
  - Exemplo: a porta 80 é geralmente usado for serviços Web por omissão

Portas normalizadas para alguns serviços

Protocolo de aplicação	Porta	Utilização
ftp	20	Transferência de ficheiros
ssh	22	Login remoto seguro
smtp	25	Envio e recepção de email
http	80	Web
pop3	110	Protocolo de correio alternativo

Nível transporte: o protocolo TCP [1]

- Objectivo: transferência de dados entre dois computadores de forma a simplificar o trabalho dos programadores de aplicações
  - Modelo oferecido é de um canal de bytes; não há perda de bytes nem chegadas fora de ordem
- Numa primeira fase é estabelecido um "*handshake*" (aperto de mão), isto é, uma ligação ou conexão entre os dois computadores
- Esta fase termina com o estabelecimento da conexão que fica conhecida e é monitorizada pelos dois computadores
- TCP é usado por: HTTP, FTP, SMTP, Telnet, …

Nível transporte: o protocolo TCP [2]

- **TCP – Transmission Control Protocol** [RFC 793] é o serviço de “transmissão fiável de dados, orientado à conexão” usado na Internet
  - Transferência de sequências de bytes de forma ordenada e fiável: *acknowledgements* (confirmações de recepção) e retransmissões se necessário
  - Controlo de fluxo: o emissor não afoga o receptor
  - Controlo da saturação: o emissor abranda o ritmo de emissão quando a rede está saturada

Nível transporte: o protocolo TCP [2]

- **TCP – Transmission Control Protocol** [RFC 793] é o serviço de “transmissão fiável de dados, orientado à conexão” usado na Internet
  - Transferência de sequências de bytes de forma ordenada e fiável: *acknowledgements* (confirmações de recepção) e retransmissões se necessário
  - Controlo de fluxo: o emissor não afoga o receptor
  - Controlo da saturação: o emissor abranda o ritmo de emissão quando a rede está saturada

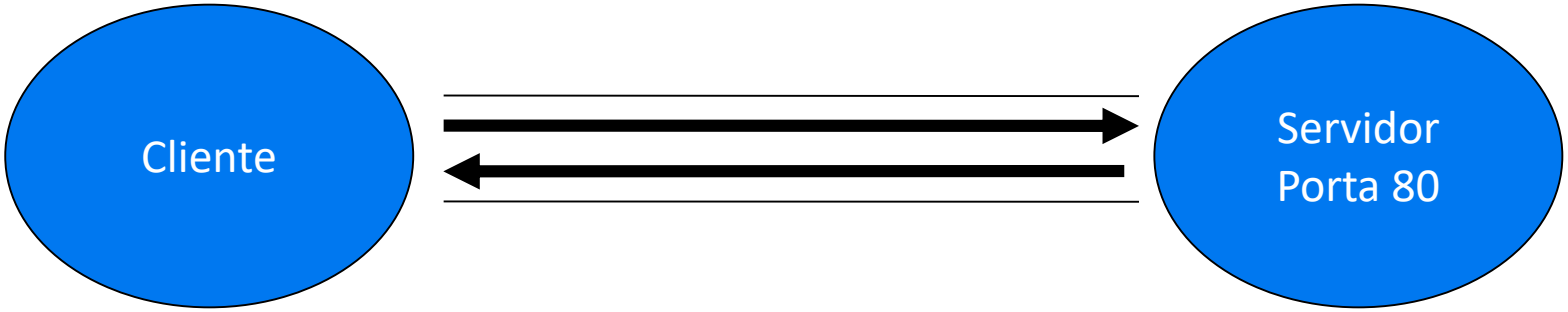
A noção de conexão [1]

- Clientes e servidores que usam o protocolo TCP comunicam enviando e recebendo sequências (streams) de bytes através de canais (connections):
  - É Ponto-a-Ponto porque dois programas são interligados
  - É Full-Duplex, porque os dados podem fluir nos dois sentidos (cliente / servidor e servidor / cliente)
  - É fiável porque (a menos de algo catastrófico) a sequência de bytes enviada pelo emissor é recebido pelo receptor, sem perdas e pela mesma ordem da emissão

A noção de conexão [2]

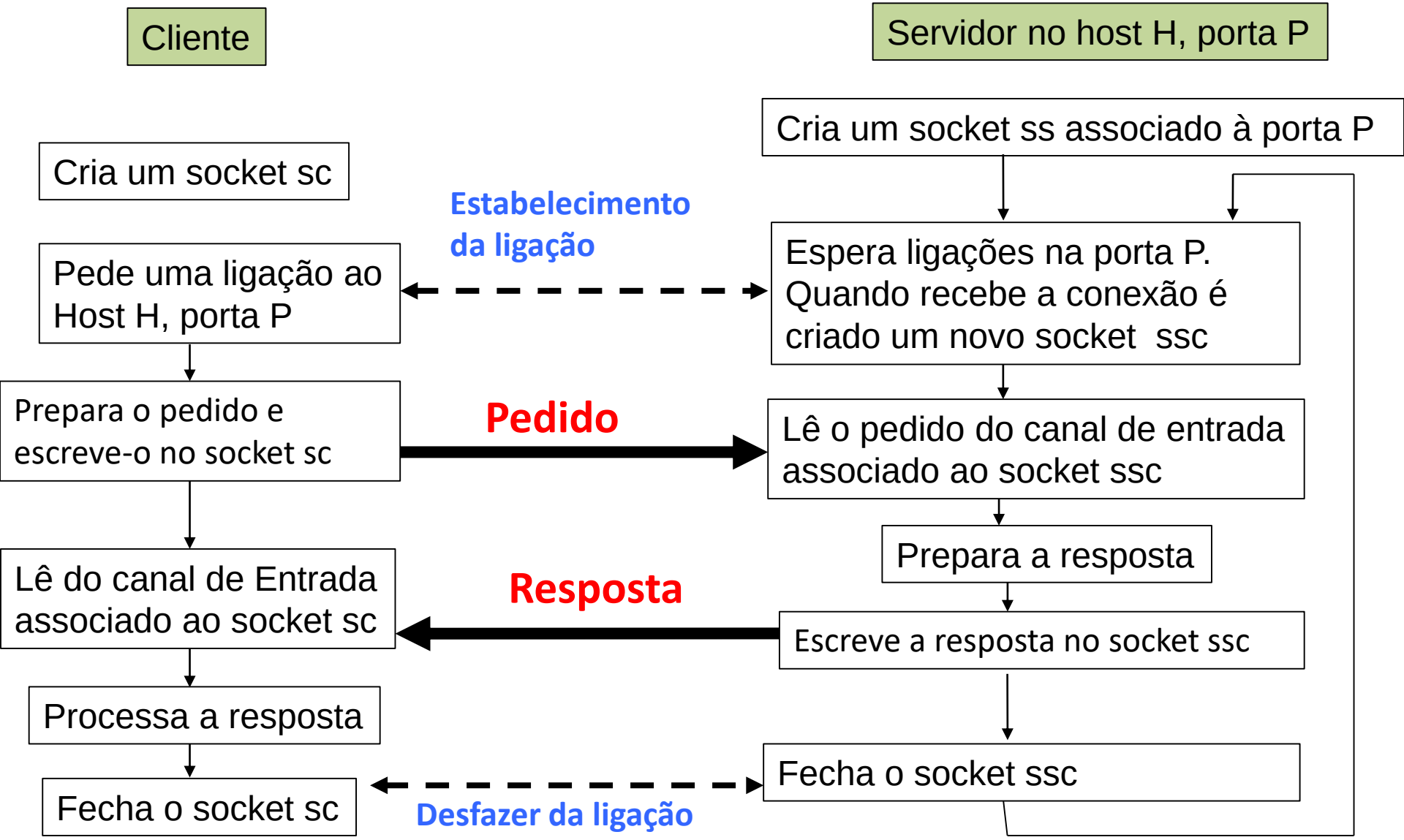
Endereço do nó do cliente: 193.136.122.33

Endereço do nó do servidor: 208.216.181.15



Porta do servidor: 80

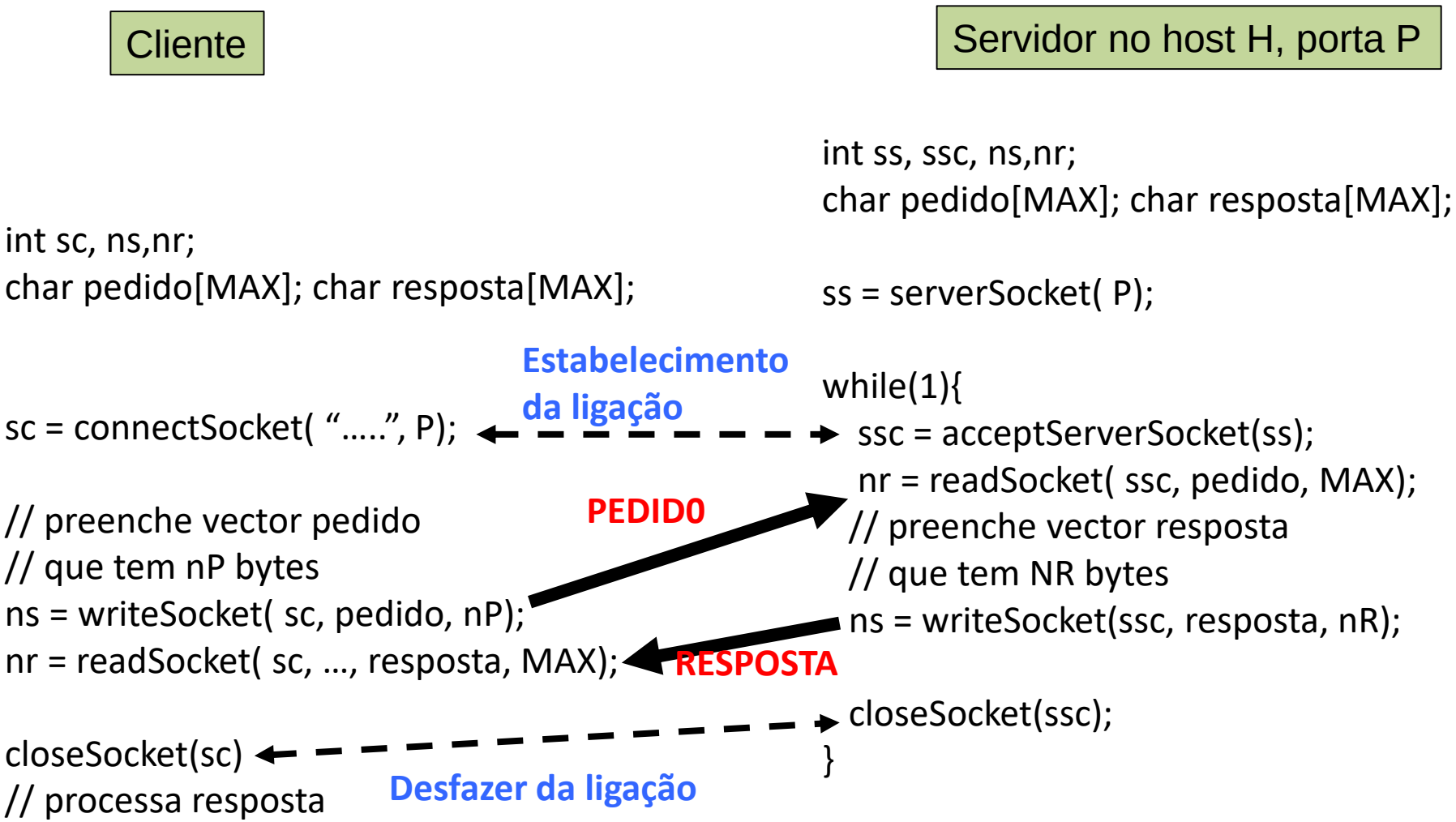
Cliente/Servidor com Sockets TCP



Uma biblioteca de sockets TCP

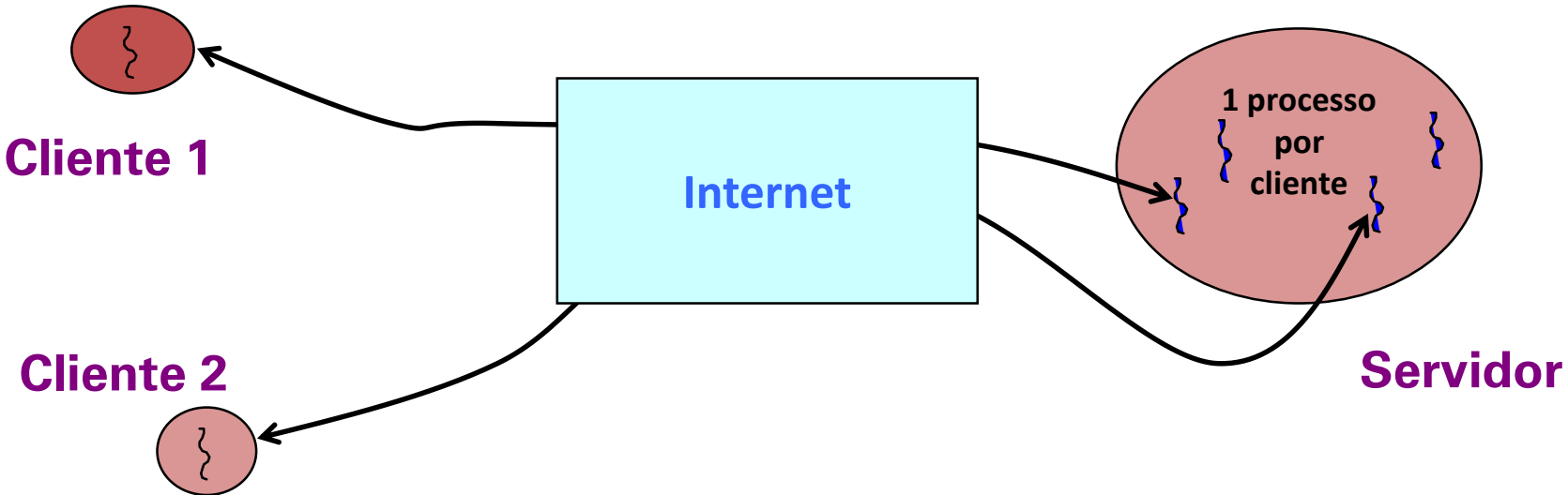
Operação	Parâmetros de entrada	Retorno
serverSocket (só no servidor)	Server Port	> 0 canal de entrada/saída (E/S) associado ao socket; -1 se erro
acceptServerSocket (só no servidor)	Canal retornado pela operação serverSocket	> 0 canal de E/S que permite ler e escrever bytes da/na conexão com o cliente
connectSocket (só no cliente)	String com o nome da máquina onde está o servidor, inteiro com o nº da porta do servidor	> 0 canal de E/S que permite ler e escrever bytes na/da conexão com o servidor
writeSocket (cliente e servidor)	canal de E/S, endereço inicial dos bytes a escrever, número de bytes a escrever	> 0, nº de bytes enviados < 0 erro
readSocket (cliente e servidor)	canal de E/S, endereço inicial do buffer para onde vão os bytes a receber, nº máximo de bytes a receber	> 0 nº de bytes recebidos < 0, erro == 0, o outro extremo da conexão fechou o canal de escrita
closeSocket (cliente e servidor)	Canal de E/S associado ao socket	0 OK; -1 erro

Cliente/Servidor com sockets TCP

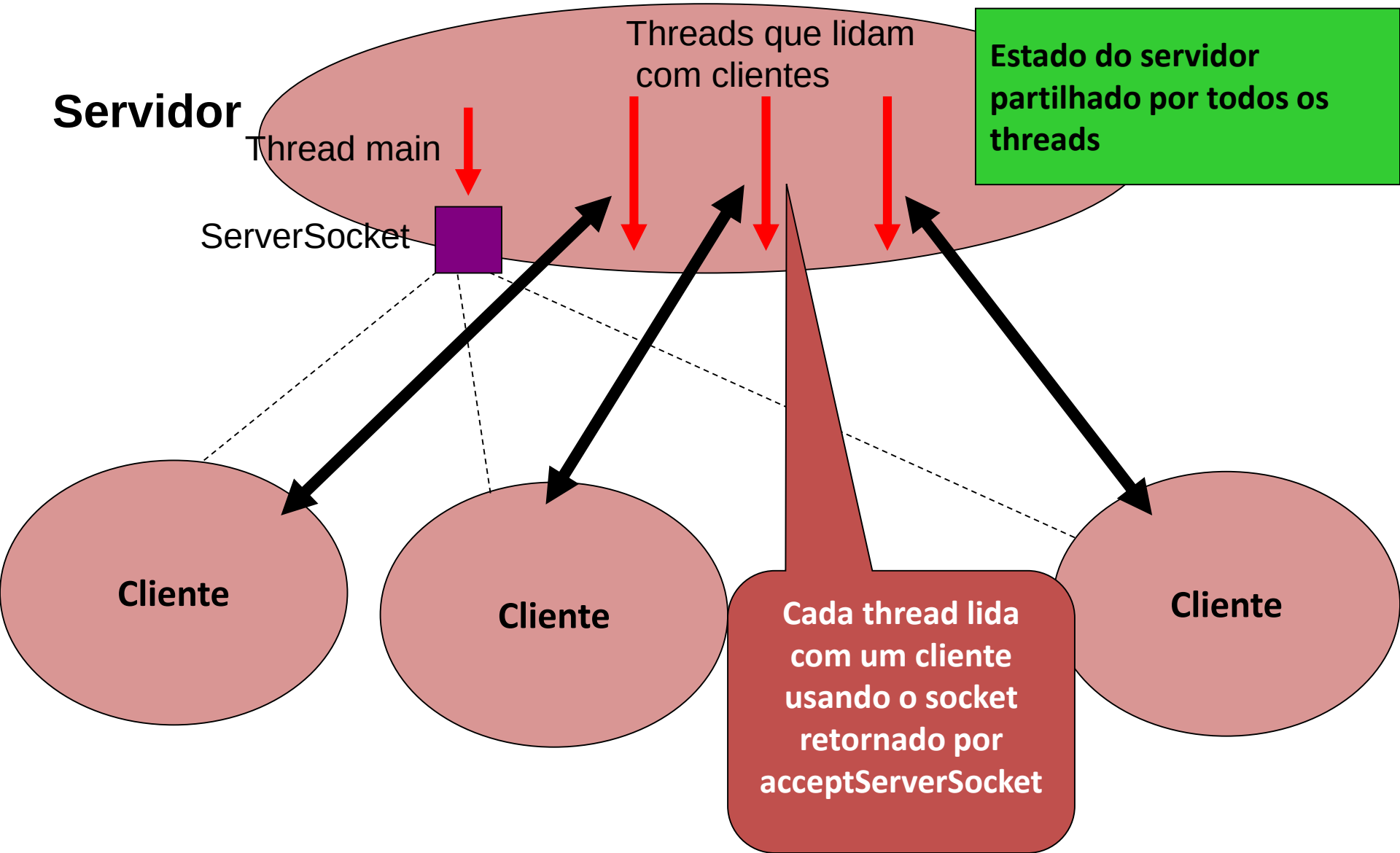


Servidores que atendem vários pedidos ao mesmo tempo

- Os servidores têm de ser capazes de dialogar com vários clientes em simultâneo
  - Os servidores que tratam vários pedidos ao meso tempo chamam-se servidores concorrentes.
- Quando é pedida uma nova conexão, o servidor aceita-a e lança um novo processo para responder ao cliente.



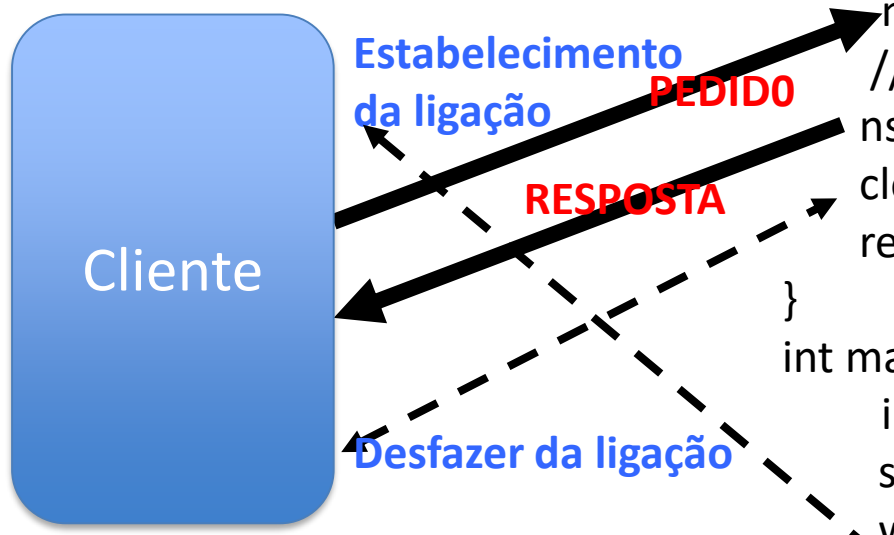
Servidor com múltiplos threads



Servidor com múltiplos threads

Servidor no host H, porta P

Cliente não alterado




```
void * handleClient( void *arg){
        int ssc, ns, nr;
        char pedido[MAX]; char resposta[MAX];
        ssc = (int)arg;
        nr = readSocket( ssc, pedido, MAX);
        // preenche vector resposta  que tem NR bytes
        ns = writeSocket(ssc, resposta, nR);
        closeSocket(ssc);
        return NULL;
}
int main( ){
        int ss, ssc;
        ss = serverSocket( P);
        while(1){
            ssc = acceptServerSocket(ss);
            pthread_create( …,…, handleClient,
              (void *) ssc);
        }
}
```