

Desenvolvimento de Aplicações de Rede

Programação com Sockets em Python

DCA0130 - Redes de Computadores

Período Letivo Suplementar Excepcional (2020.5)

Prof. Carlos M. D. Viegas

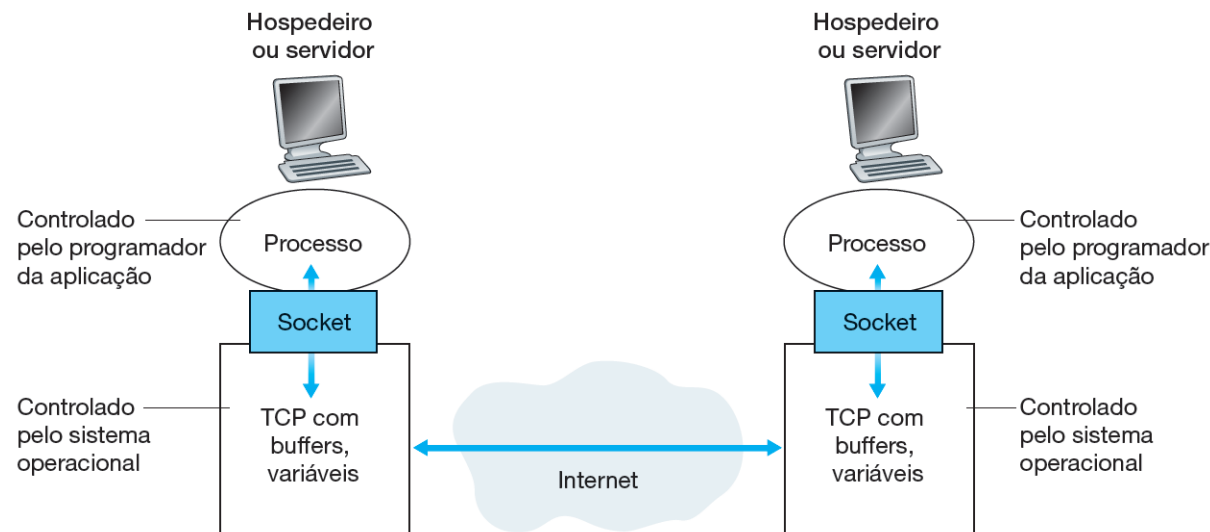


Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte



Programação com sockets

- API Sockets
 - Utilizada para construir aplicações no modelo cliente/servidor
 - Apareceu no BSD 4.1 UNIX em 1981
 - Dois tipos de serviço de transporte via API Sockets
 - Datagrama não confiável (UDP)
 - Fluxo de bytes, confiável (TCP)



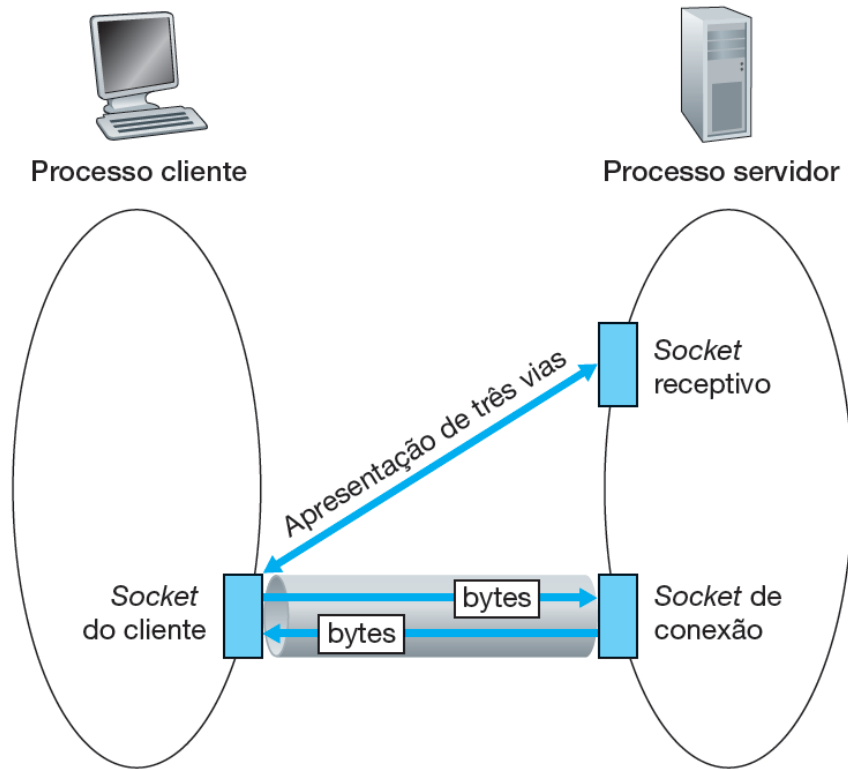
SOCKET

É uma interface local (uma “porta”) no host, criada pela aplicação, e controlada pelo SO, pela qual um processo de aplicação pode **enviar e/ou receber** mensagens de/para outro processo de aplicação (remoto ou local)

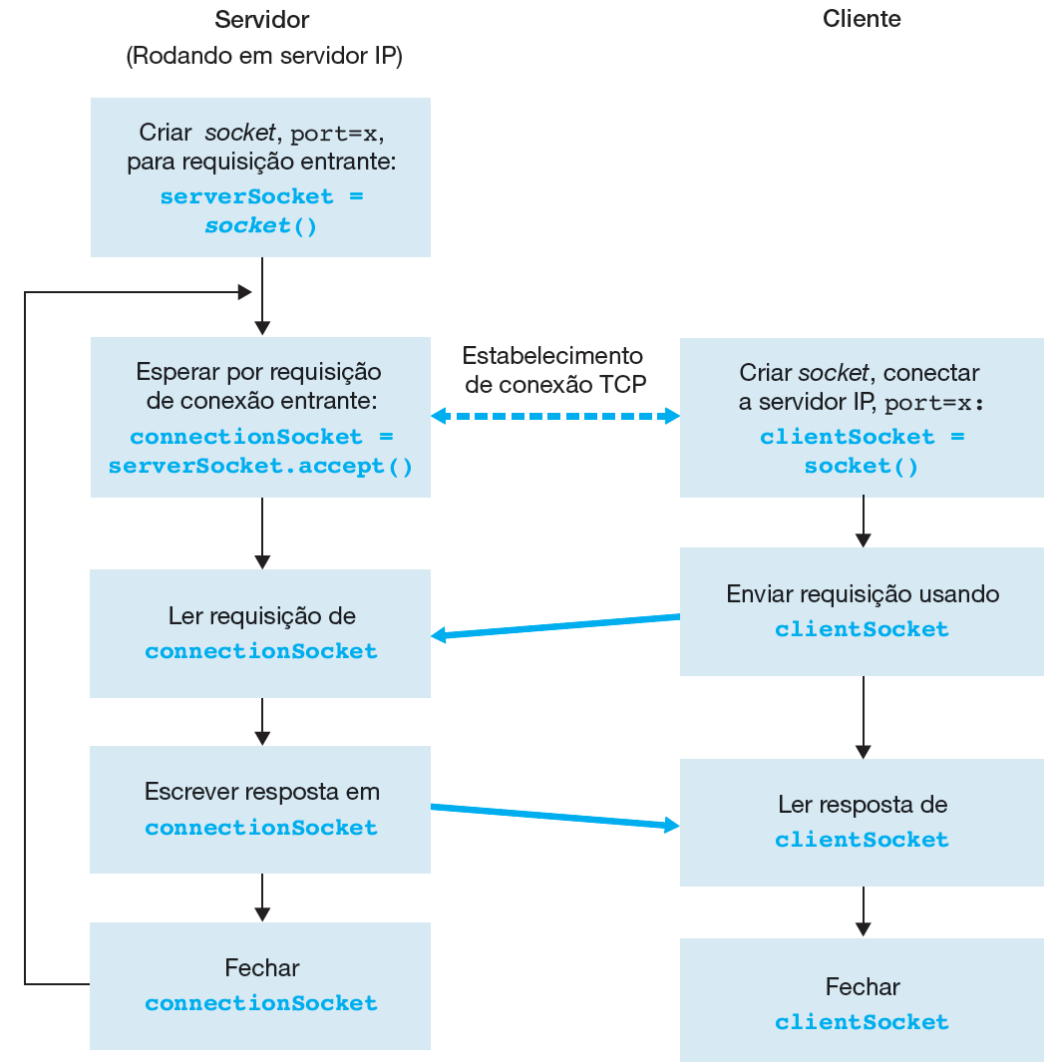
Programação com sockets usando TCP

- Serviço TCP
 - Transferência confiável de bytes de um processo para outro
- Servidor deve estar em execução e aceitando novas conexões
 - Fica em modo de “escuta”
 - A cada conexão um novo socket é criado
 - É possível ter múltiplos sockets com diferentes clientes
 - Endereços IP e portas de origem e destino são usados para distinguir os clientes
- Cliente deve contatar servidor
 - Cria um socket com o servidor
 - Deve-se indicar o endereço IP e porta de “escuta” do servidor

Programação com sockets usando TCP

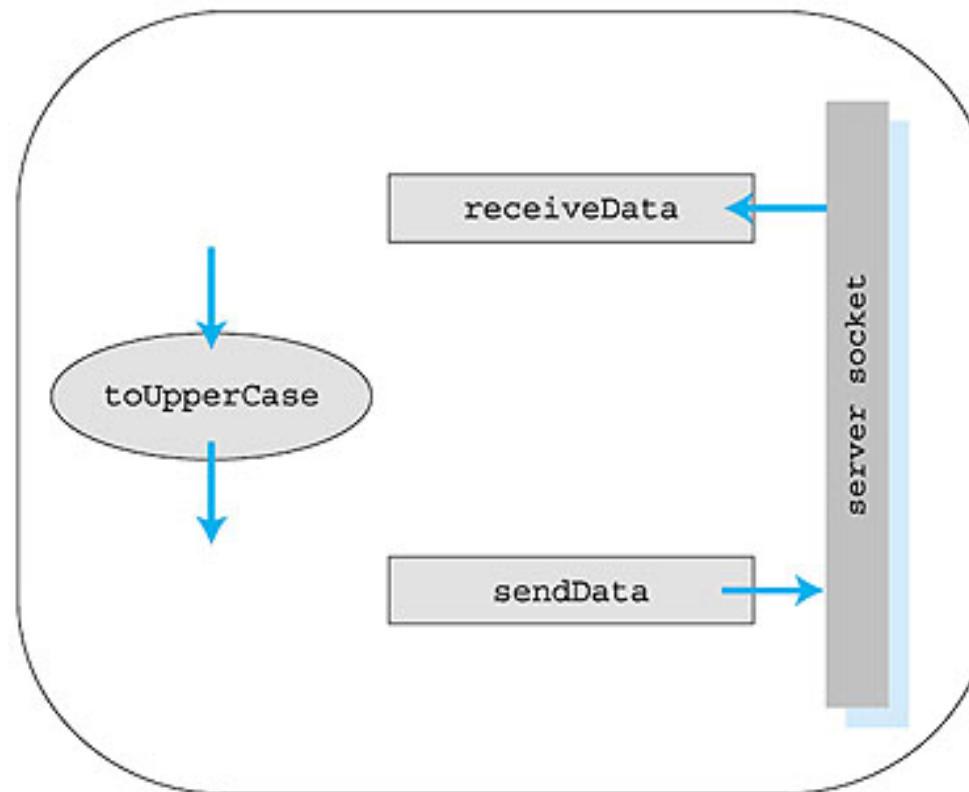


Do ponto de vista da aplicação:
*TCP provê transferência
confiável, ordenada de bytes (“tubo”) entre
cliente e servidor*



Exemplo de uma aplicação com sockets TCP

- Conversor de texto minúsculo para maiúsculo
 - O servidor recebe um texto enviado pelo cliente, transforma em letras maiúsculas e retorna o texto transformado para o cliente



Exemplo de uma aplicação com sockets TCP

Servidor (em linguagem Python)

Cria socket TCP de recepção →

Atribui o endereço e a porta ao socket →

Servidor inicia a escuta por solicitações TCP →

Servidor espera no `accept()` por solicitações, um novo socket é criado no retorno →

Lê bytes do socket →

Envia para o cliente →

Fecha conexão para este cliente (mas *não* o socket de recepção) →

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('', serverPort))
serverSocket.listen(1)
print 'The server is ready to receive'
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024)
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence)
    connectionSocket.close()
```

Socket Internet do tipo TCP (stream)

Exemplo de uma aplicação com sockets TCP

Cliente (em linguagem Python)

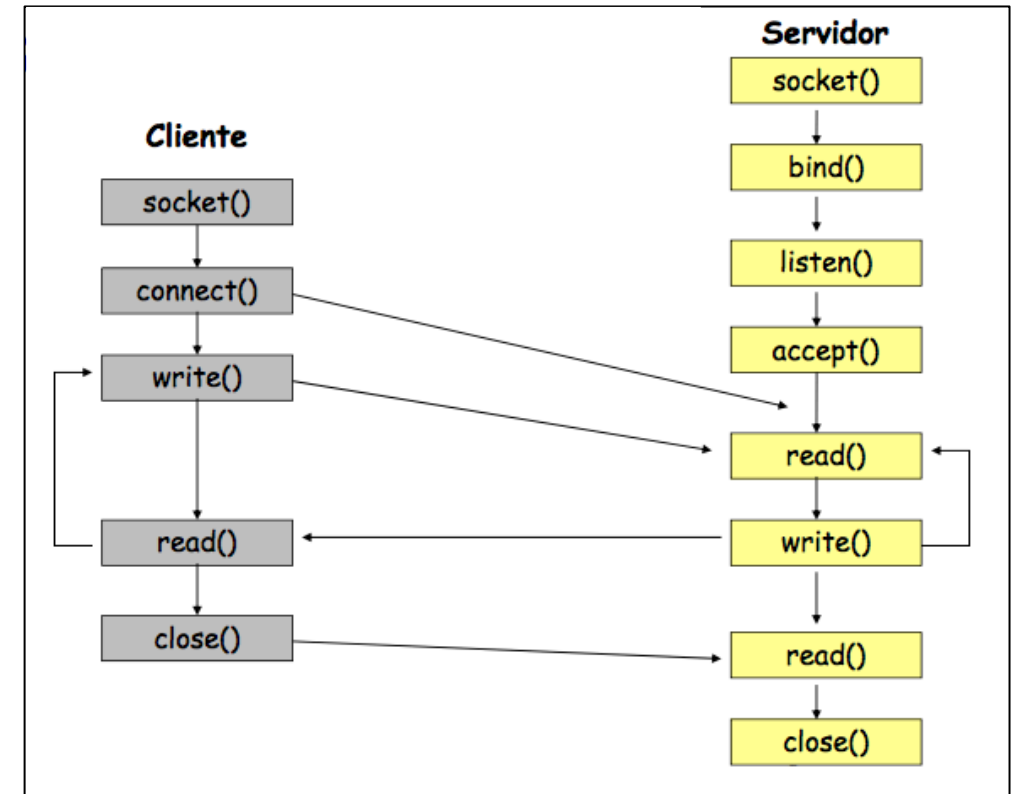
Conecta o socket
criado ao servidor, por
meio da porta remota
12000

```
from socket import *
serverName = 'servername'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = raw_input('Input lowercase sentence:')
clientSocket.send(sentence)
modifiedSentence = clientSocket.recv(1024)
print 'From Server:', modifiedSentence
clientSocket.close()
```

Programação com sockets TCP

- Métodos TCP (em Python):

Primitiva	Significado
<code>socket()</code>	Cria um novo ponto de comunicação
<code>bind()</code>	Atribui o endereço local a um socket
<code>listen()</code>	Anunciar a disposição para aceitar conexões
<code>accept()</code>	Bloquear o responsável pela chamada até uma tentativa de conexão ser recebida
<code>connect()</code>	Tentar estabelecer uma conexão ativamente
<code>send()</code>	Enviar dados por meio da conexão
<code>recv()</code>	Receber dados por meio da conexão
<code>close()</code>	Encerrar conexão

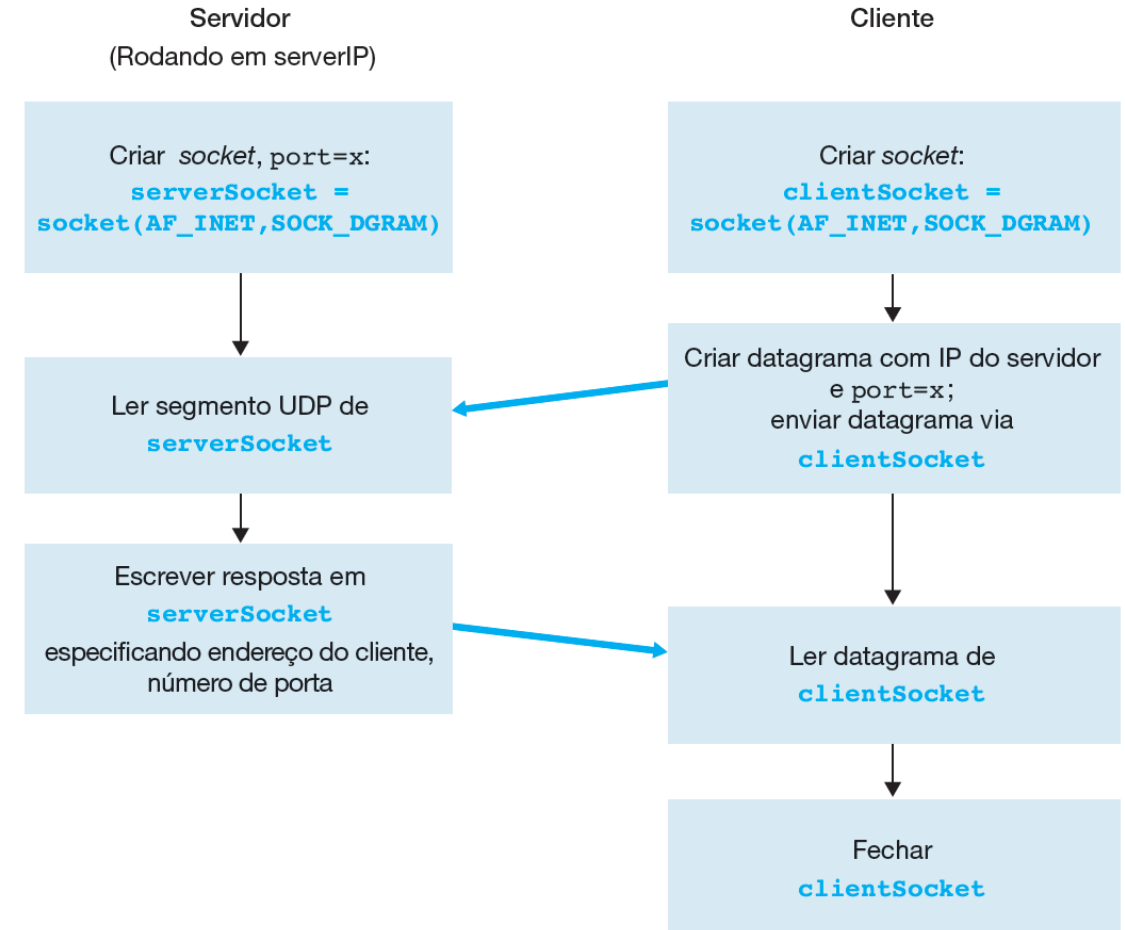


Programação com sockets usando UDP

- Serviço UDP:

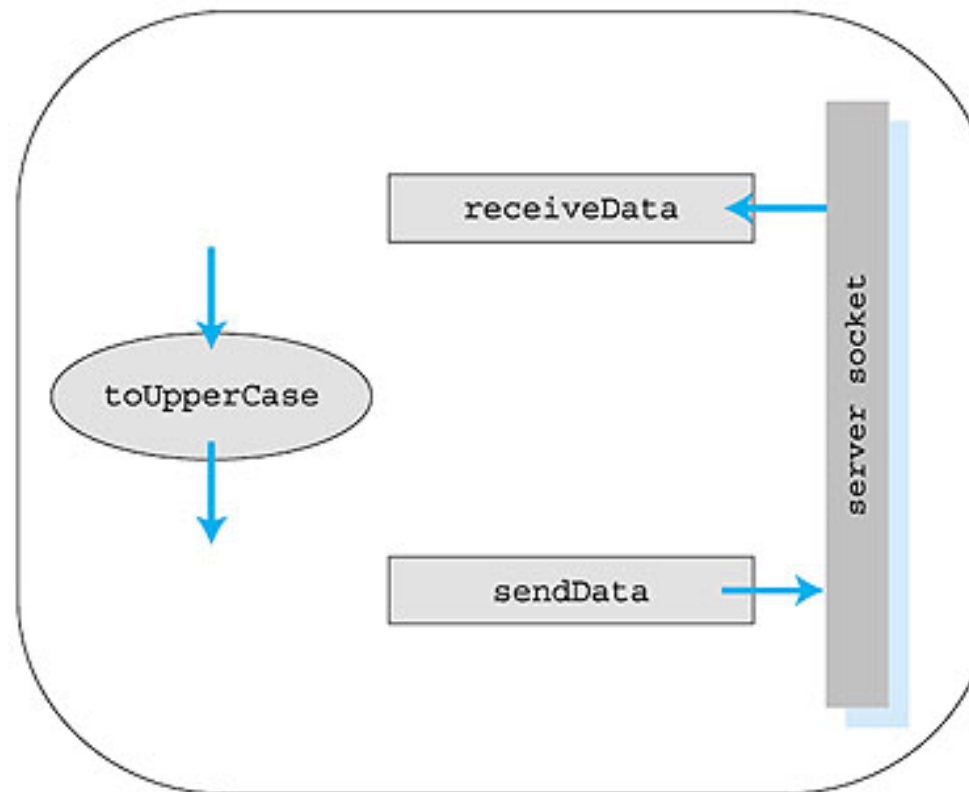
- Transferência não confiável
- Não tem “conexão” entre cliente e servidor
- Não tem “*handshake*”
- Os dados são “simplesmente” enviados

Do ponto de vista da aplicação:
*UDP provê transferência não confiável
de grupos de bytes (“datagramas”)
entre cliente e servidor*



Exemplo de uma aplicação com sockets UDP

- Conversor de texto minúsculo para maiúsculo
 - O servidor recebe um texto enviado pelo cliente, transforma em letras maiúsculas e retorna o texto transformado para o cliente



Exemplo de uma aplicação com sockets UDP

Servidor (em linguagem Python)

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print 'The server is ready to receive'
while 1:
    message, clientAddress = serverSocket.recvfrom(2048)
    modifiedMessage = message.upper()
    serverSocket.sendto(modifiedMessage, clientAddress)
```

Cria socket UDP →

Atribui o endereço e a porta ao socket →

Lê mensagem do socket UDP, obtendo endereço do cliente →

Retorna *string* em maiúsculas para o cliente →

Socket Internet do tipo UDP (dgram)

Exemplo de uma aplicação com sockets UDP

Cliente (em linguagem Python)

```
from socket import *
serverName = 'hostname'
serverPort = 12000
clientSocket = socket(socket.AF_INET, socket.SOCK_DGRAM)
message = raw_input('Input lowercase sentence:')
clientSocket.sendto(message, (serverName, serverPort))
modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
print modifiedMessage
clientSocket.close()
```

Obtém entrada do teclado do usuário →

Acrescenta o endereço do servidor e porta à mensagem; envia pelo socket →

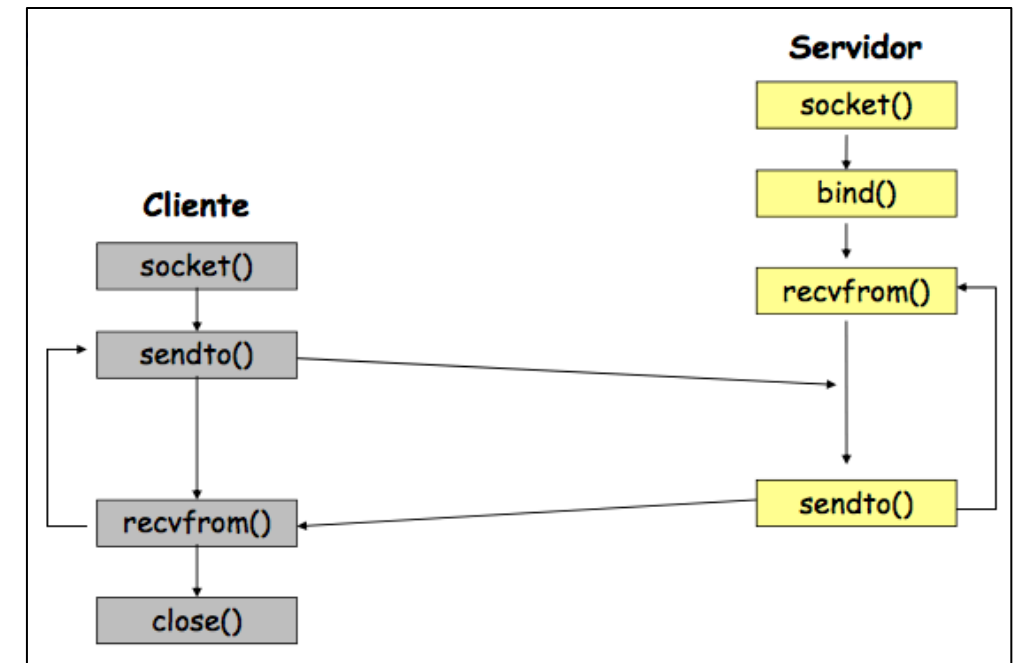
Imprime string recebido e fecha socket →

Lê caracteres de resposta do socket →

Programação com sockets UDP

- Métodos UDP (em Python):

Primitiva	Significado
<code>socket()</code>	Cria um novo ponto de comunicação
<code>bind()</code>	Atribui o endereço local a um socket
<code>sendto()</code>	Enviar dados
<code>recvfrom()</code>	Receber dados
<code>close()</code>	Fechar socket



Programação com sockets

- Todos os códigos apresentados (bem como outros) estão disponíveis para prática na página de apoio à disciplina:

<https://www.dca.ufrn.br/~viegas>

Ensino > DCA0130 - Redes de Computadores > Sockets

- Para executar os códigos, se faz necessária a instalação de um interpretador Python disponível no site: <https://www.python.org/downloads/>