

Programação com UDP



UDP: não há “conexão” entre cliente e servidor

- Sem negociação (handshaking) antes de enviar dados
- O emissor coloca explicitamente o endereço IP de destino e o nº da porta em cada datagrama
- O recetor extrai o endereço de retorno e a porta do datagrama recebido

UDP: os dados podem ser perdidos ou recebidos fora de ordem

Do ponto de vista da aplicação:

- UDP fornece transferência *não fiável* de grupos de bytes (datagramas ou mensagens) entre cliente e servidor

Interação cliente/servidor usando sockets UDP



servidor (a correr em serverIP)

Criar socket associado à porta X
`serverSocket =
socket(AF_INET,SOCK_DGRAM)`

Ler datagrama do
`serverSocket`

Escrever resposta em
`serverSocket`
especificando
endereço IP do cliente
Número da porta

client



Criar Socket
`clientSocket =
socket(AF_INET,SOCK_DGRAM)`

Criar datagrama e associar serverIP e porta X ;
Enviar datagrama através de `clientSocket`

Ler datagrama de
`clientSocket`

fechar
`clientSocket`



Cliente UDP



Cliente Python UDPClient

Incluir a biblioteca de sockets do Python

→ `from socket import *`

`serverName = 'hostname'`

`serverPort = 12000`

Criar socket UDP

→ `clientSocket = socket(AF_INET,
SOCK_DGRAM)`

Obter entrada do teclado

→ `message = input('Input lowercase sentence:')`

Definir endereço IP e porta de destino; enviar pelo socket

→ `clientSocket.sendto(message.encode(),
(serverName, serverPort))`

Ler resposta para a string

→ `modifiedMessage, serverAddress =
clientSocket.recvfrom(2048)`

Imprime a mensagem recebida e fecha o socket

→ `print modifiedMessage.decode()
clientSocket.close()`

`message.encode()` converte a variável do tipo String do Python para um array de bytes a colocar no datagrama

`modifiedMessage.decode()` faz o inverso

Servidor UDP



Servidor Python UDPServer

	from socket import *
	serverPort = 12000
Criar socket UDP	→ serverSocket = socket(AF_INET, SOCK_DGRAM)
Associar à porta local 12000	→ serverSocket.bind(("", serverPort))
	print ("The server is ready to receive")
Em ciclo	→ while True:
Ler do socket UDP para message, obtém endereço do cliente(client IP and port)	→ message, clientAddress = serverSocket.recvfrom(2048)
	modifiedMessage = message.decode().upper()
Envia mensagem de volta para o cliente	→ serverSocket.sendto(modifiedMessage.encode(), clientAddress)

Servidor arranca e está disponível



```
pm — Python — 80x24
Last login: Tue Sep 20 09:53:20 on console
pm@10-22-205-61 ~ % python
Python 3.9.13 (main, May 21 2022, 02:36:14)
[Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

```
pm — Python — 80x24
pm@10-22-205-61 ~ % python
Python 3.9.13 (main, May 21 2022, 02:36:14)
[Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> from socket import *
[>>> serverPort=12000
[>>> serverSocket = socket(AF_INET, SOCK_DGRAM)
[>>> serverSocket.bind(("",serverPort))
[>>> message, clientAdress = serverSocket.recvfrom(2048)
]
```

Cliente antes de enviar



```
pm — Python — 80x24
pm@10-22-205-61 ~ % python
Python 3.9.13 (main, May 21 2022, 02:36:14)
[Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from socket import *
>>> clientSocket = socket(AF_INET, SOCK_DGRAM)
>>> serverPort=12000
>>> serverName='localhost'
>>> message = input("Input lowercase sentence:")
Input lowercase sentence:asdfghjklzxcvbn
>>> clientSocket.sendto(message.encode(),(serverName, serverPort))
```

```
pm — Python — 80x24
pm@10-22-205-61 ~ % python
Python 3.9.13 (main, May 21 2022, 02:36:14)
[Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from socket import *
>>> serverPort=12000
>>> serverSocket = socket(AF_INET, SOCK_DGRAM)
>>> serverSocket.bind(("",serverPort))
>>> message, clientAdress = serverSocket.recvfrom(2048)
>>>
```

Após envio pelo cliente e receção pelo servidor



```
pm@10-22-205-61 ~ % python
Python 3.9.13 (main, May 21 2022, 02:36:14)
[Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from socket import *
>>> clientSocket = socket(AF_INET, SOCK_DGRAM)
>>> serverPort=12000
>>> serverName='localhost'
>>> message = input("Input lowercase sentence:")
Input lowercase sentence:asdfghjklzxcvbn

>>> clientSocket.sendto(message.encode(),(serverName, serverPort))
15
>>>
```

```
pm@10-22-205-61 ~ % python
Python 3.9.13 (main, May 21 2022, 02:36:14)
[Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from socket import *
>>> serverPort=12000
>>> serverSocket = socket(AF_INET, SOCK_DGRAM)
>>> serverSocket.bind(("", serverPort))
>>> message, clientAddress = serverSocket.recvfrom(2048)
>>> print(message.decode())
asdfghjklzxcvbn

>>> print(clientAddress)
('127.0.0.1', 57701)
>>>
```

Envio da resposta pelo servidor



```
[>>>
[>>>
[>>> modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
[ ]
```

```
] [>>>
] [>>>
] [>>> modifiedMessage=message.decode().upper()
] [>>> serverSocket(modifiedMessage.encode(),clientAddress)
]
```

```
[>>>
[>>>
[>>>
[>>> modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
[>>> ]
```

```
] [>>>
] [>>>
] [>>> serverSocket.sendto(modifiedMessage.encode(),clientAddress)
] 15
[>>> ]
```

```
[>>> modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
[>>> print(modifiedMessage.decode())
ASDFGHJKLZXCVBN
[>>> ]
```

```
] [>>>
] [>>> serverSocket.sendto(modifiedMessage.encode(),clientAddress)
] 15
[>>> ]
```


Cliente e servidor



```
1 from socket import *
2
3 clientSocket = socket(AF_INET, SOCK_DGRAM)
4
5 serverName="localhost"
6 serverPort= 12000
7
8 message=input("Input lowercase sequence: ")
9
10 clientSocket.sendto( message.encode(), (serverName, serverPort))
11
12 modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
13
14 print(modifiedMessage.decode())
15
16 clientSocket.close()
```

```
from socket import *
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverPort= 12000
serverSocket.bind(('', serverPort))
while True:
    message, clientAddress = serverSocket.recvfrom( 2048)
    modifiedMessage = message.decode().upper()
    serverSocket.sendto( modifiedMessage.encode(), clientAddress)
serverSocket.close()
```

Desktop — Python UDPserver.py — 80x24

```
[pm@10-22-205-61 Desktop %
[pm@10-22-205-61 Desktop %
[pm@10-22-205-61 Desktop %
[pm@10-22-205-61 Desktop % python UDPserver.py
█
```

Desktop — -zsh — 80x24

```
[pm@10-22-205-61 Desktop % python UDPCClient.py
Input lowercase sequence: qwertyuiop
QWERTYUIOP
[pm@10-22-205-61 Desktop % python UDPCClient.py
Input lowercase sequence: asdfghjkl
ASDFGHJKL
[pm@10-22-205-61 Desktop % python UDPCClient.py
Input lowercase sequence: zxcvbnm
ZXCVCBNM
[pm@10-22-205-61 Desktop % █
```