Reti di Calcolatori I

Prof. Roberto Canonico
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

Corso di Laurea in Ingegneria Informatica

A.A. 2019-2020

Esempi di programmi client/server in Python 3: comunicazione con UDP

Università degli Studi di Napoli Federico II

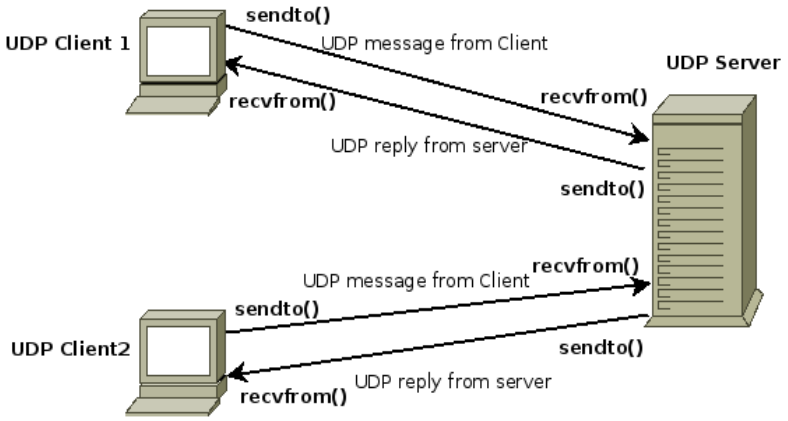
Nota di copyright per le slide COMICS

Nota di Copyright

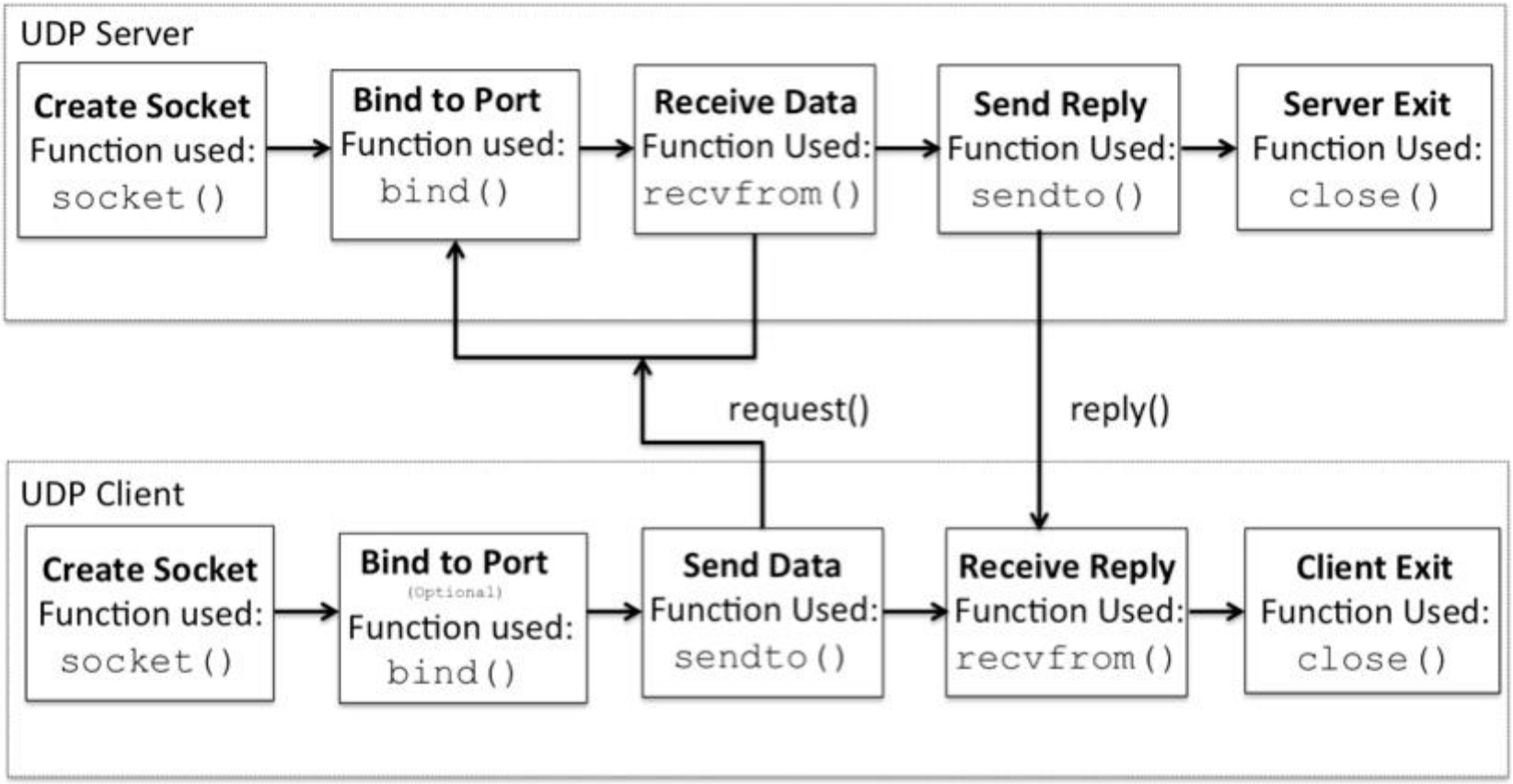
Autori:
Simon Pietro Romano, Antonio Pescapè, Stefano Avallone,
Marcello Esposito, Roberto Canonico, Giorgio Ventre

Comunicazione con UDP

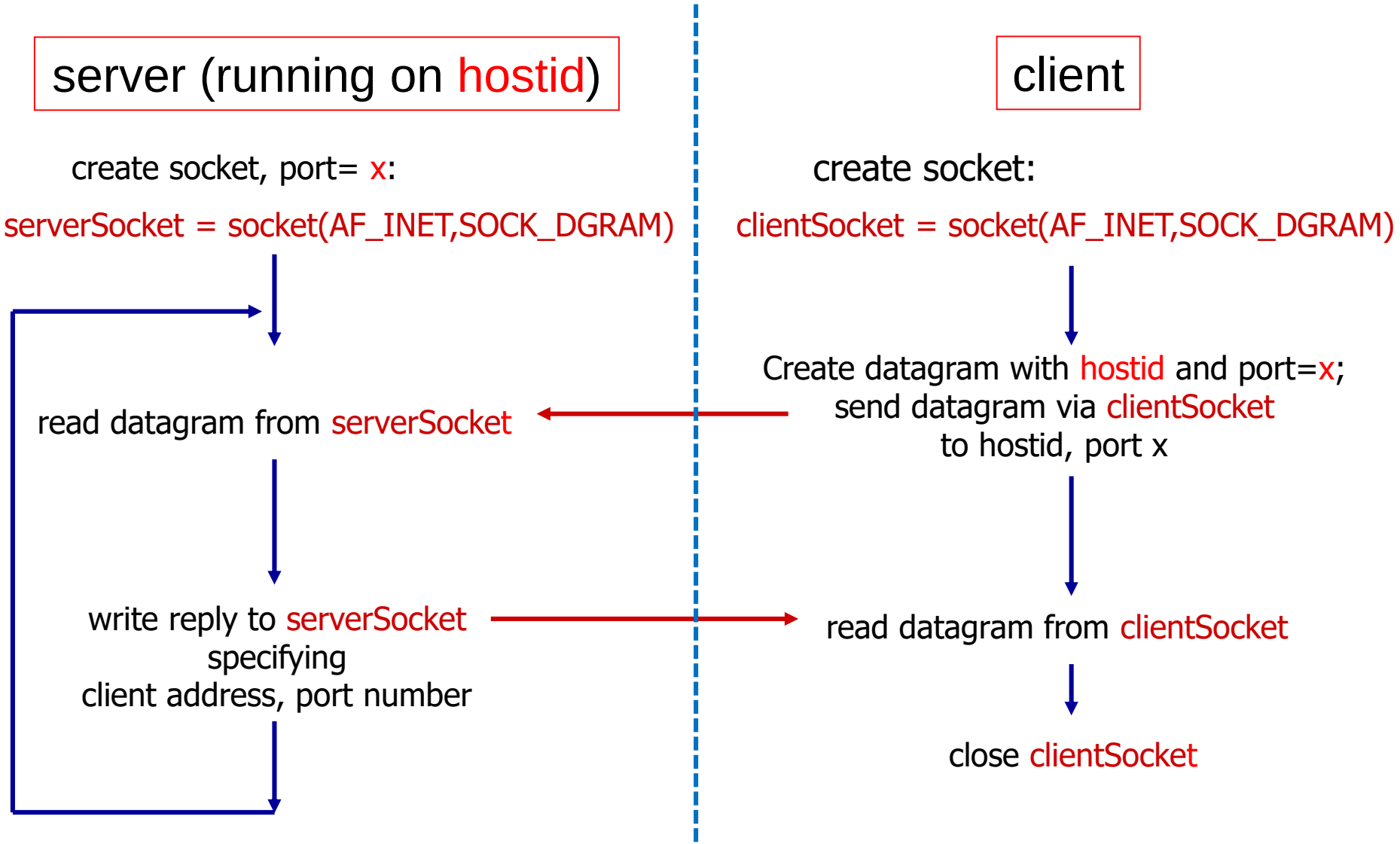
- La comunicazione con UDP è *connectionless*
- Il client invia al server un messaggio in un singolo datagramma UDP sul *port number* usato dal server per ricevere i messaggi
  - Il *port number sorgente* usato dal client è scelto dal client tra quelli non usati
  - Il *port number destinazione* usato dal client è quello sul quale è noto che il server stia in ricezione (*well known*)
- Il server risponde al mittente con un messaggio di risposta in un datagramma UDP indirizzato al client
  - Nel messaggio di risposta i port number sorgente e destinazione si scambiano
- Le primitive usate sono:
  - sendto()
  - recvfrom()



UDP in Python: client e server



Client/server con UDP: schema



UDP client in Python 3

```
import sys, getopt, socket
# ... def usage():
# ... def read_args():
SERVER_ADDRESS = "127.0.0.1"
SERVER_PORT = 12000
# read command line arguments
read_args()
# get user keyboard input
request = input("Input lowercase sentence: ")
# create UDP socket
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
# send request to server
clientSocket.sendto(request.encode(), (SERVER_ADDRESS, SERVER_PORT))

# wait for server response
print("Waiting for server response...")
response, serverAddress = clientSocket.recvfrom(2048)

# Print out received response
print(response.decode())

# Close socket
clientSocket.close()
```

UDP server in Python 3

```
import sys, getopt, socket
# ... def usage():
# ... def read_args():
SERVER_ADDRESS = "0.0.0.0"
SERVER_PORT = 12000
# read command line arguments
read_args()
# create UDP socket
serverSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serverSocket.bind((SERVER_ADDRESS, SERVER_PORT)) # bind socket to local port

while True:
    print("Server waiting on (%s, %d)" % (SERVER_ADDRESS, SERVER_PORT))
    # receive request
    request, clientAddress = serverSocket.recvfrom(2048)
    print("Received request from (%s, %d)" % (clientAddress[0], clientAddress[1]))
    # convert message to upper case
    response = request.upper()
    # send back modified string to client
    serverSocket.sendto(response, clientAddress)
    print("Sent reply to (%s, %d)" % (clientAddress[0], clientAddress[1]))
```