

Socket Programming in Python

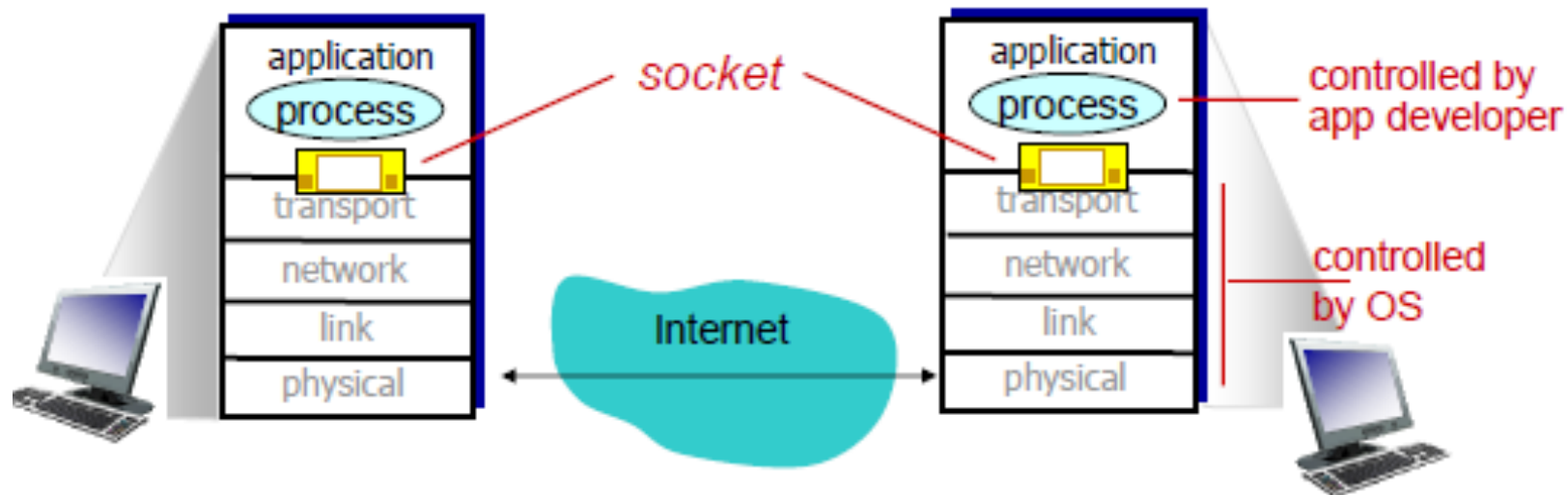
TA 連潔琳 R08942159@ntu.edu.tw

Outline

- Socket Concept
- UDP Socket example
- TCP Socket example
- Homework

Socket Concept

- “Socket” means by the way to enable 2 or more process to be connected
- Socket :
 1. door between application process and end-to-end-transport protocol
 2. interface from application process and kernel process



Socket Concept

Two socket types for two transport services:

- UDP: unreliable datagram
- TCP: reliable byte-stream oriented

Application Example:

- UDP: Media streaming(lost frames), Multi-players game
- TCP: Web, SSH(PTT), SMTP(Mail)

Server-client model

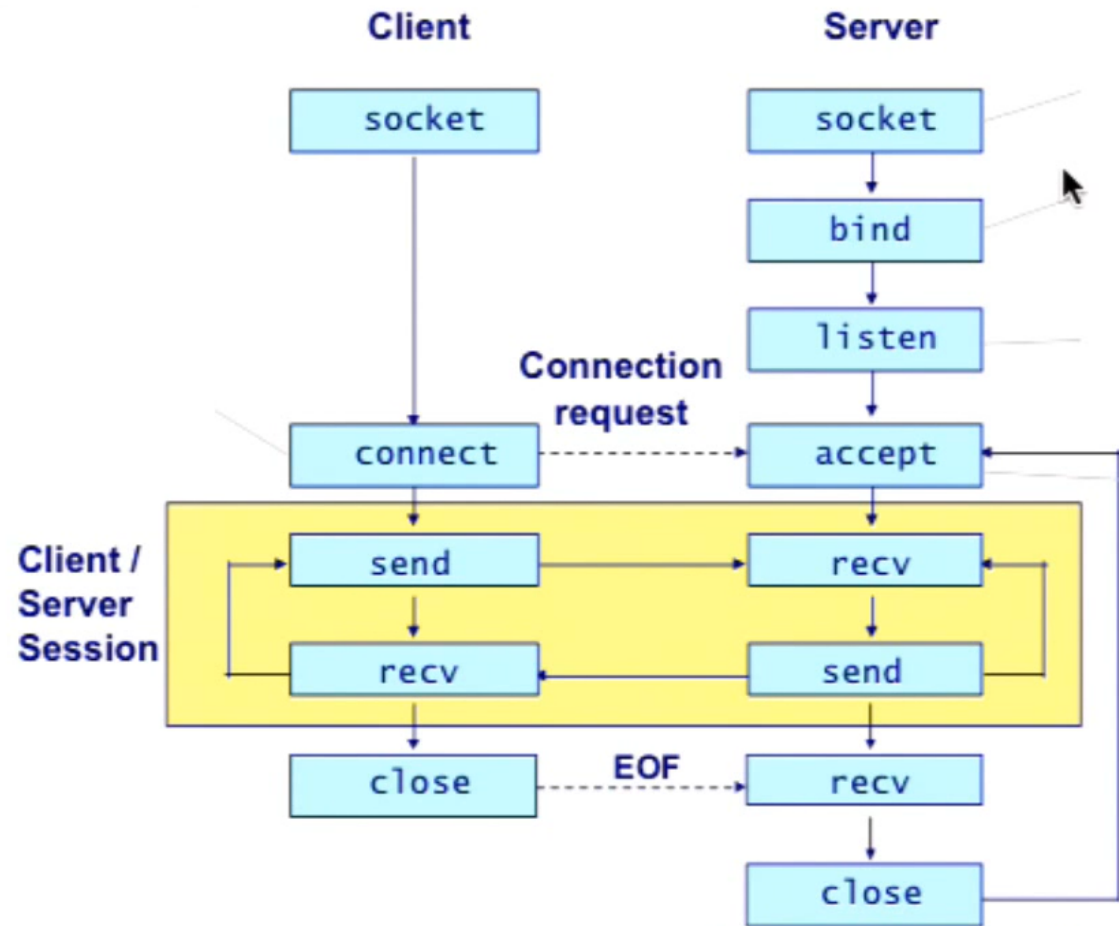
Server

- Server process must be running first
- Server must have created socket(door) for client's contact

Client

- Creating TCP or UDP socket, specifying socket address of server process
- When using TCP, client and server establish connection by “handshaking”

TCP Socket example



TCP Socket example(Server)

- Create a new socket

Address family for
(host, port)

Socket type for TCP

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

- Bind the socket to the specified (ID, Port number)

```
s.bind((HOST, PORT))
```

HOST = "" means INADDR_ANY, which is used to bind all the interfaces

- Listen for new request with the socket

```
s.listen(0)
```

TCP Socket example(Server)

- Accept a new connection (request) and create a new socket for the connection

```
client, address = s.accept()
```

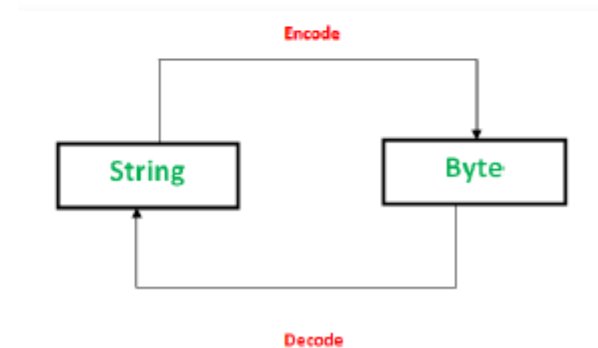
- Send and receive data by the socket

- The data transmitted by the socket is “Byte object” → change it to “string” and print on the screen

```
client.send(b"What's your student ID?\n")  
student_id = client.recv(1000).upper().decode('utf-8')
```

- Close the socket

```
client.close()
```



TCP Socket example(Client)

- Create a new socket

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

- Connect to a remote socket at address

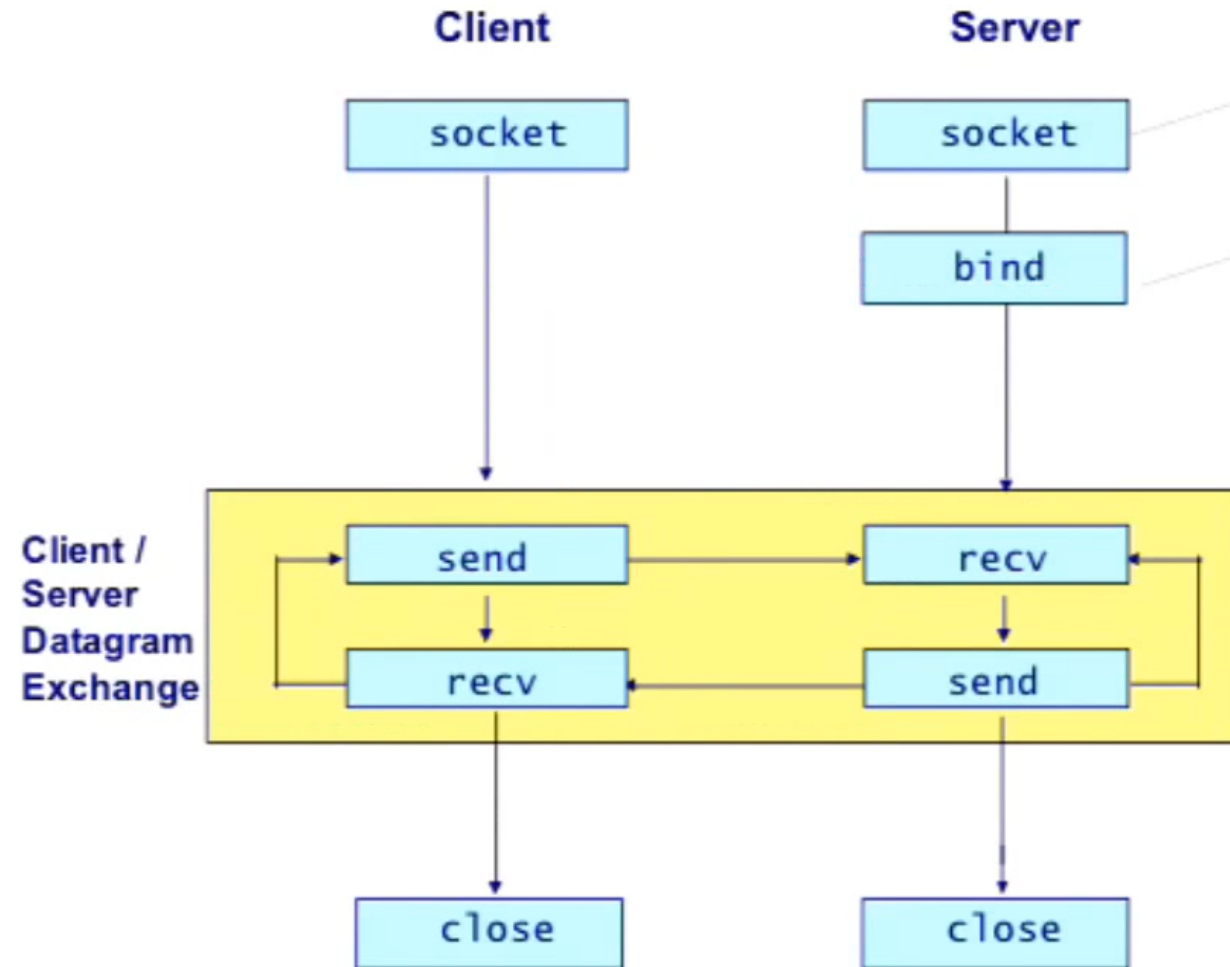
```
s.connect((HOST, PORT))
```

A pair (host, port) for address

- Send and receive data

```
response = input(s.recv(1000).decode('utf-8')).encode('utf-8')  
s.send(response)
```

UDP Socket example



UDP Socket example(Server)

- Create a new socket

Socket type for UDP

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

- Bind the socket to the specified (ID, Port number)

```
sock.bind((UDP_IP, UDP_PORT))
```

- Send and receive data from the socket

```
data, addr = sock.recvfrom(1024)
```

Address of the client

```
sock.sendto(b'data received', addr)
```

UDP Socket example(Client)

- Create a new socket

Socket type for UDP

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

- Send and receive data from the socket

```
sock.sendto(MESSAGE.encode(), (UDP_IP, UDP_PORT))
```

```
data, addr = sock.recvfrom(1024)
```

For More Information

- Python document for socket

<https://docs.python.org/3/library/socket.html>

- TCP Socket

<https://realpython.com/python-sockets/>

- UDP Socket

<https://wiki.python.org/moin/UdpCommunication>