

Socket Programming

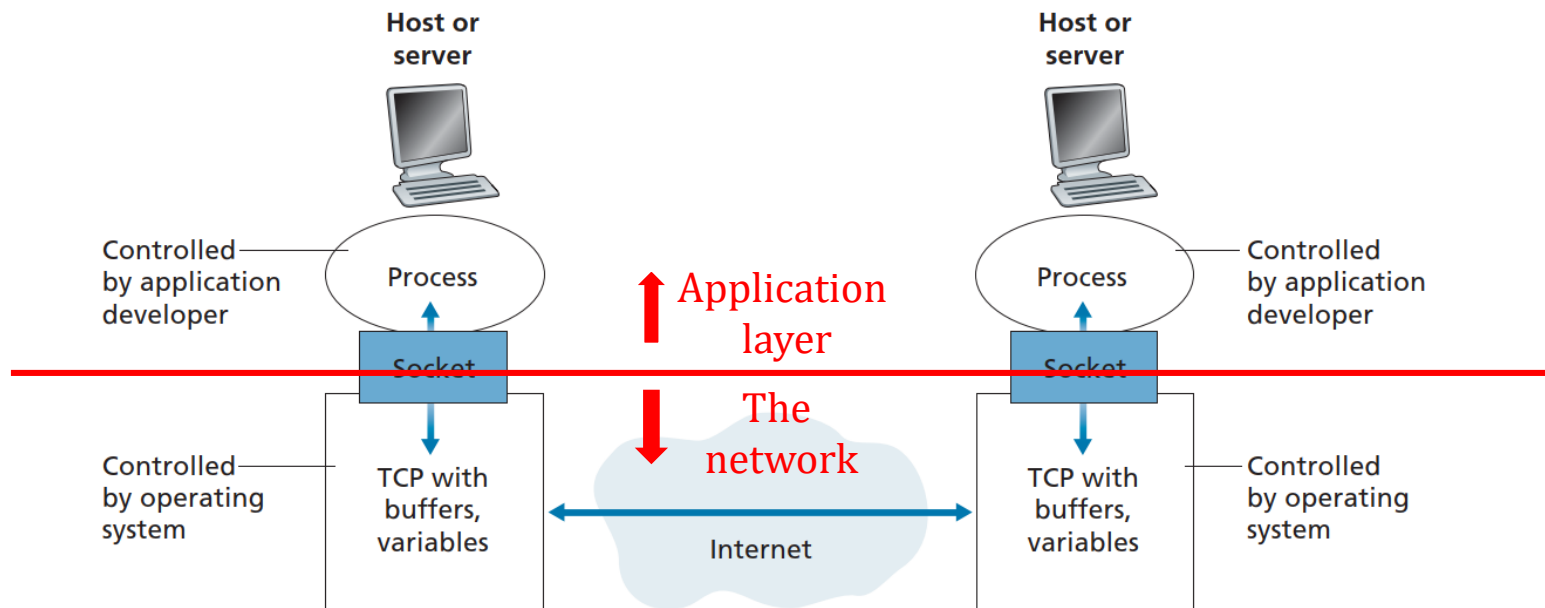
Chih-Yu Lin

2025/03/13



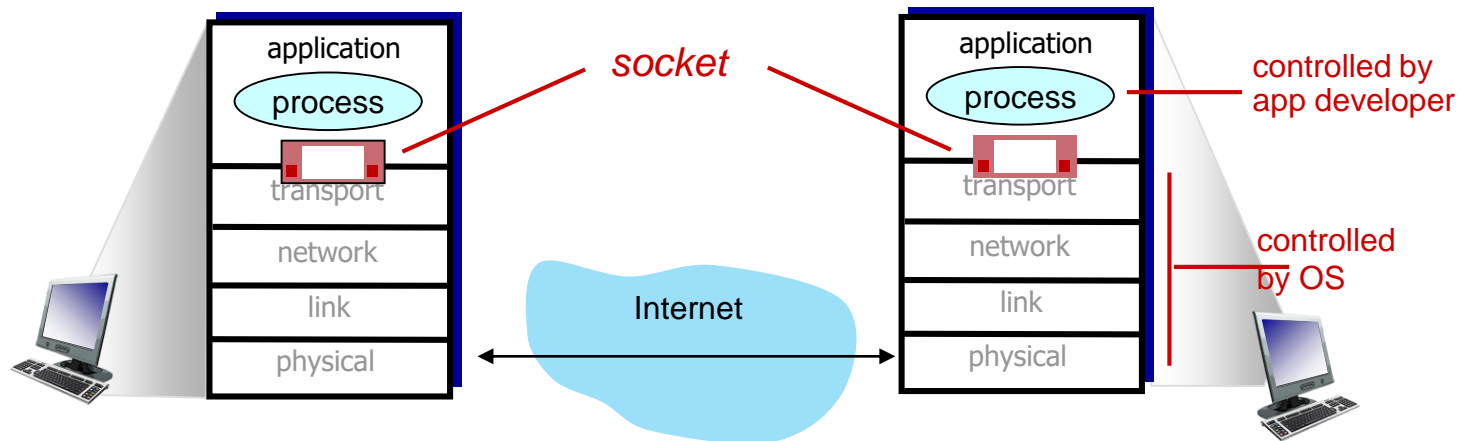
Sockets

- A network application consists of a pair of processes in communication
 - A client process and a server process for each pair
- **Socket**: A software interface (API) between the process and the network



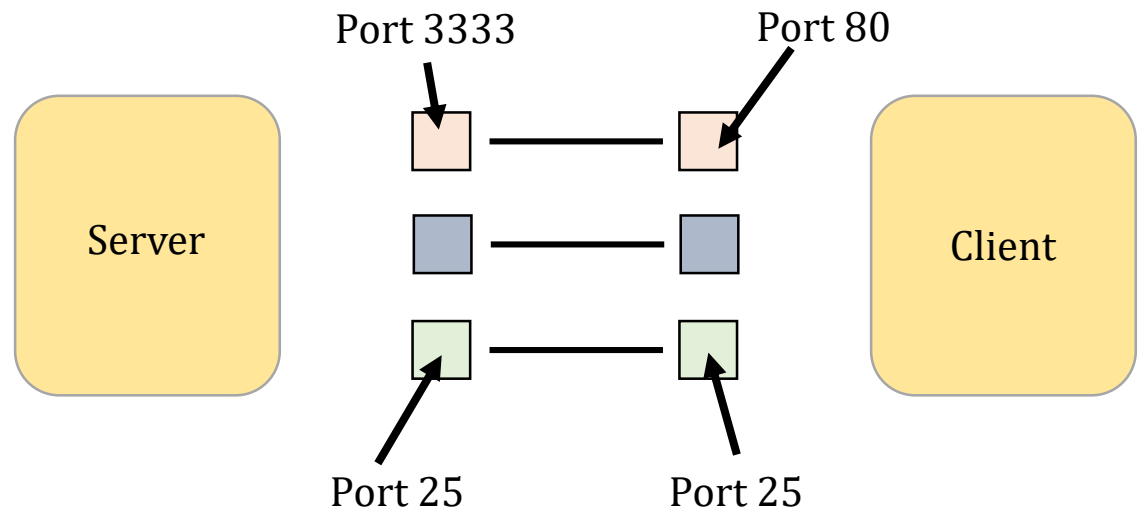
Sockets

- Socket is the API between the application layer and the transport layer
 - Application layer
 - Transport layer: Choose transport protocols or adjust some transport layer parameters



How to address processes?

- Identify the receiving process by specifying
 - The address of the host
 - An identifier that specifies the receiving process
- The host address: IP address
- The identifier: Destination port number
 - HTTP: port 80
 - HTTPS: port 443
 - FTP: port 21
 - DNS: port 53



IP Address + Port number = Socket

Socket programming

Two socket types for two transport services:

- *Transmission Control Protocol (TCP)*: reliable, byte stream-oriented
- *User Datagram Protocol (UDP)*: unreliable datagram

Examples:

- *TCP*: Web, buffered-streaming
- *UDP*: Live streaming (could lose some frames), multi-player games

Socket programming with TCP

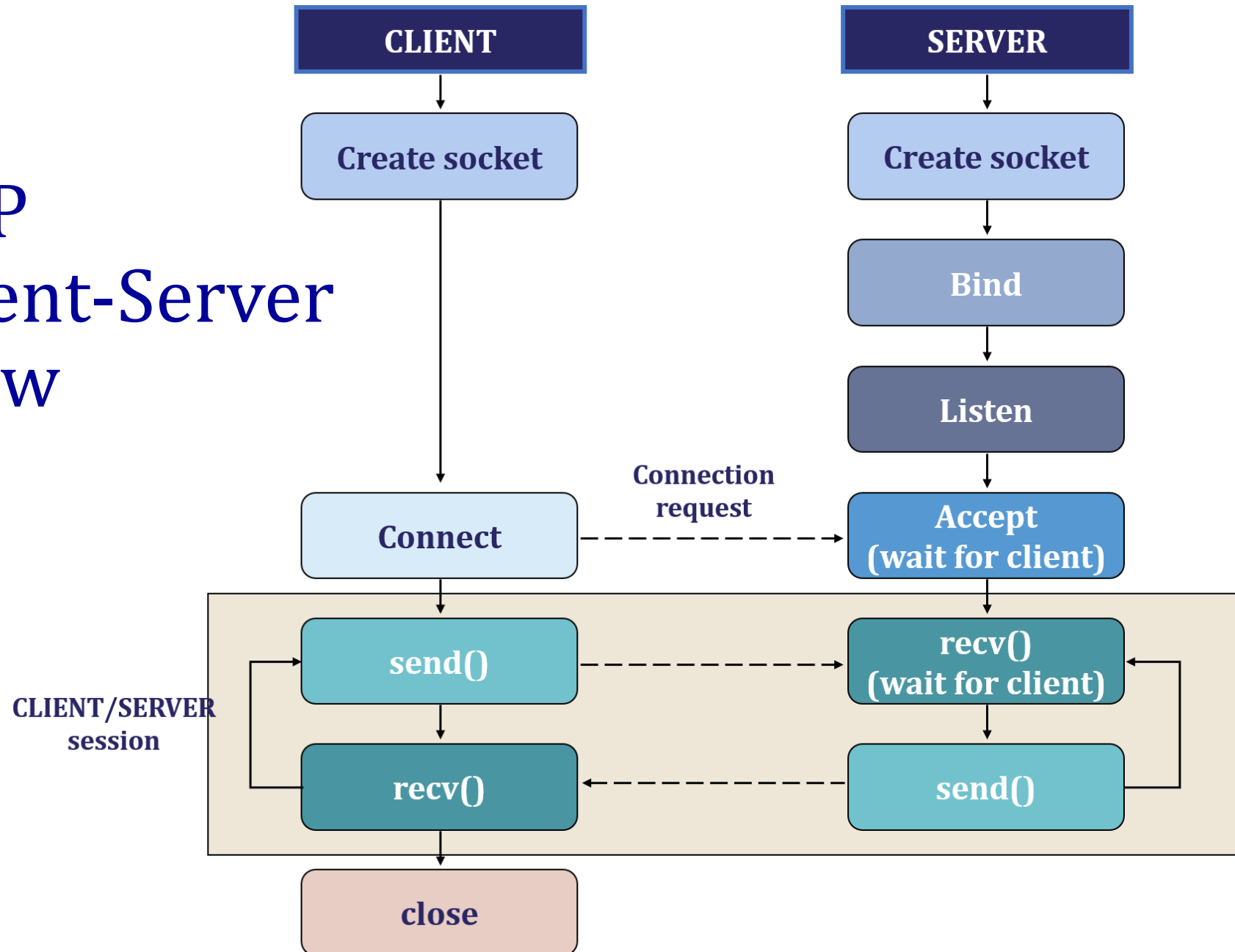
Server

- Server process must run first
- Server must create a socket that welcomes client's contact

Client

- Create TCP socket
- Specify IP address and port number of server process
- When client creates the socket, client TCP establishes connection to server TCP by three-way handshaking

TCP Client-Server Flow



TCP Socket Example (Server)

1) Create a new socket

- `socket.AF_INET`: The *Address Family* format is host and port number
- `socket.SOCK_STREAM`: The socket type for TCP

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

2) Bind the socket to the a specified (ID, Port Number)

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

3) Listen to a new request with the socket

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


TCP Socket Example (Server)

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

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

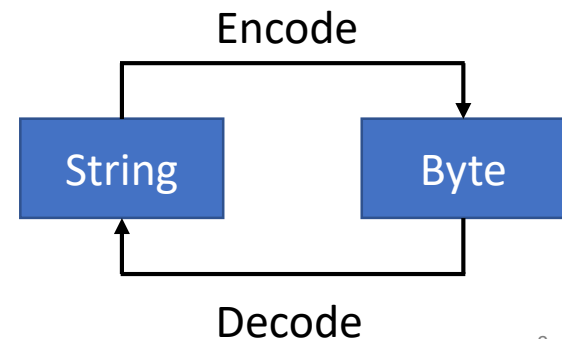
- 5) Send and receive data by socket

- The data transmitted by socket is a byte object
- Convert the message to string first before printing it on the screen

```
Message = client.recv(1000).decode('utf-8')
```

- 6) Close the socket

```
client.close()
```



TCP Socket Example (Client)

1) Create a new socket

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

2) Connect to a remote socket

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

3) Receive and send data

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

Example: TCP Server

Create TCP socket

Server begins to listen for incoming TCP requests

Server waits on `accept()` for incoming requests, then new socket is created on return

Read bytes from socket

```
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 True:
    connectionSocket, addr = serverSocket.accept()

    sentence = connectionSocket.recv(1024).decode()
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence.encode())
    connectionSocket.close()
```

Example: TCP Client

Create a TCP socket
Set remote port to 12000



```
from socket import *

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

HTML

(Hyper Text Markup Language)

- The standard markup language
 - Create and describe the structure of Web pages
- Basics of a HTML File
 - An HTML element is defined by a start tag, some content, and an end tag
- Build up a HTML file
 - 1) Create a *.txt* file
 - 2) Change the *.txt* to *.html*
 - 2) Open *.html* by your browser

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Reference

- Python Document for Socket

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

- TCP Socket Guide

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

- HTML Tutorial

<https://www.w3schools.com/html/>