Socket Programming

Basics

- Socket is an interface between application and network
  - Application creates a socket
  - Socket type dictates the style of communication
- Once socket is configured, applications
  - Pass data to the socket for network transmission
  - Receive data transmitted across the network from the socket

Server Program

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((' ',50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
           print data
server.close()
```

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

Creating Sockets

sockfd = socket(socket_family, socket_type)

socket_family: Network Layer Protocol

- AF_INET – IPv4
- AF_INET6 – IPv6

socket_type: Transport Layer Protocol

- SOCK_STREAM – TCP
- SOCK_DGRAM – UDP

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((' ',50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

Binding Sockets

sockfd.bind((host_address, port))

Binds the socket to particular address and port

- ‘ ’ indicates “any interface” address

Why no bind called for client??

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((' ',50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

Listen for Connections

sockfd.listen(backlog)

- Prepares socket to accept connections
  - backlog: number of pending connections allowed
- Allows sockets to respond to new connections using the three-way handshake

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

Accept Connections

client, address = sockfd.accept()

- *WAITS* for a client to establish the connection
  - client: socket fd for handling the client connection
  - address: IP address of the client

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

Receive Data

data = sockfd.recv(sz, [flags])

- WAITS for data on sockfd
- Retrieves up to '*sz*' bytes of data when available

- *flags* indicate property of recv function-
  - MSG_DONTWAIT: make recv non-blocking
  - MSG_PEEK: only peek data; don't remove from buffer
  - And many more… (do *man recv*)

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

Close Socket

sockfd.close()

Close the connection by sending FIN

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

Wait for Input

- *accept* and *recv* are blocking
- Server needs to handle multiple connections
- Cannot proceed by blocking on every connection
- Need a single function to wait for input on “any” socket fd

```
import select
import socket
import sys
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((‘ ‘,50000))
server.listen(30)
input = [server]
while 1:
    inputready,outputready,exceptready = select.select(input,[],[])
    for s in inputready:
        if s == server:
         client, address = server.accept()
         input.append(client)
        else:
           data = s.recv(4096)
            print data
server.close()
```

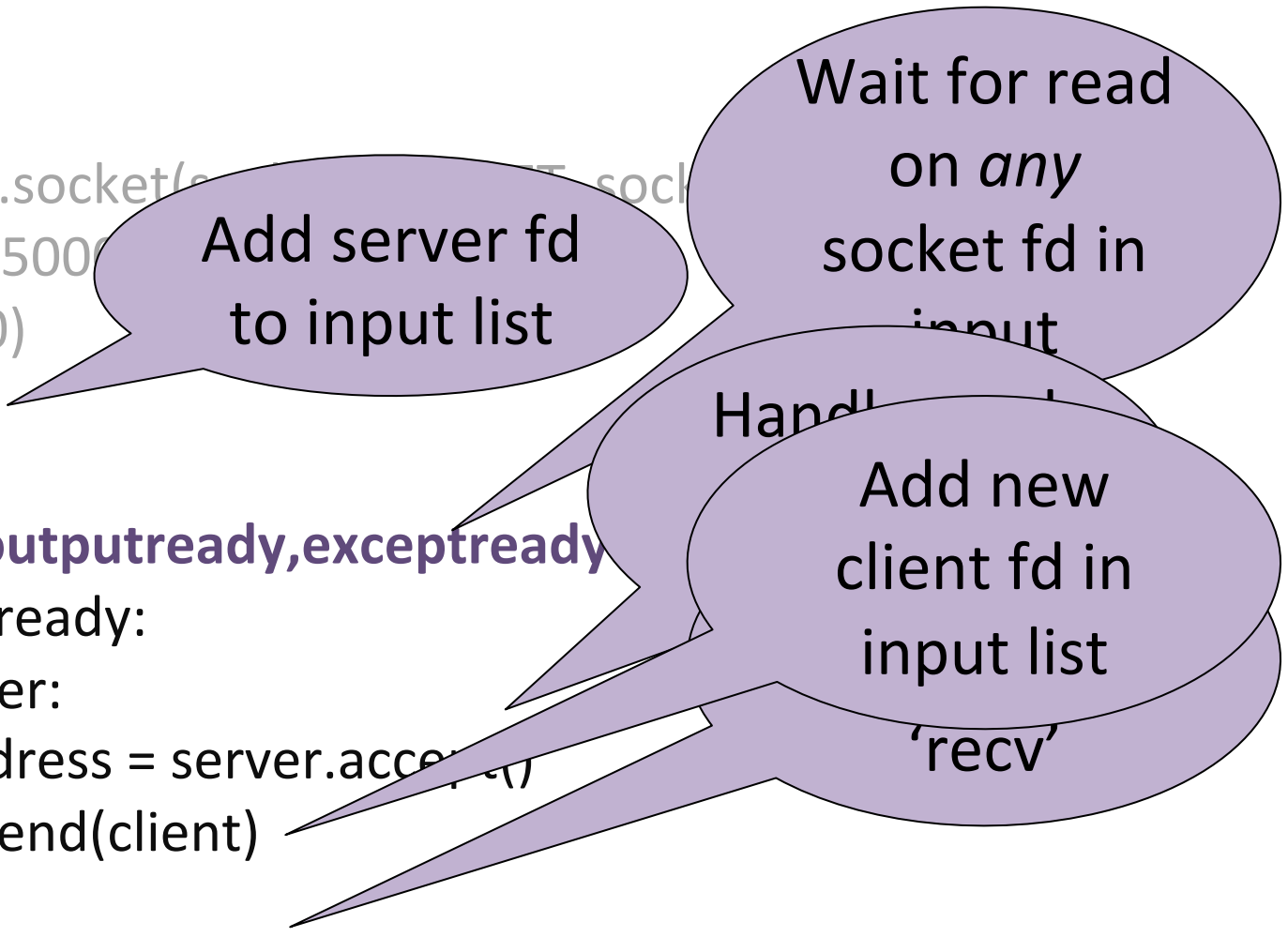
Wait for Input

r,w,x = select(rlist, wlist, xlist, [timeout])

- rlist: list of file descriptor to wait on for reading
- r: file descriptor ready for reading
- wlist: list of file descriptor to wait on for writing
- w: file descriptor ready for writing
- xlist: list of file descriptor to wait on for exceptions
- x: file descriptor ready for exception handling

- Waits on **any** fd in **any** of these lists until “timeout”

```
import select
import socket
import sys
server = socket.socket(
server.bind((' ',500
server.listen(30)
input = [server]
while 1:
   inputready,outputready,exceptready
   for s in inputready:
      if s == server:
       client, address = server.acce
       input.append(client)
      else:
         data = s.recv(4096)
          print data
server.close()
```

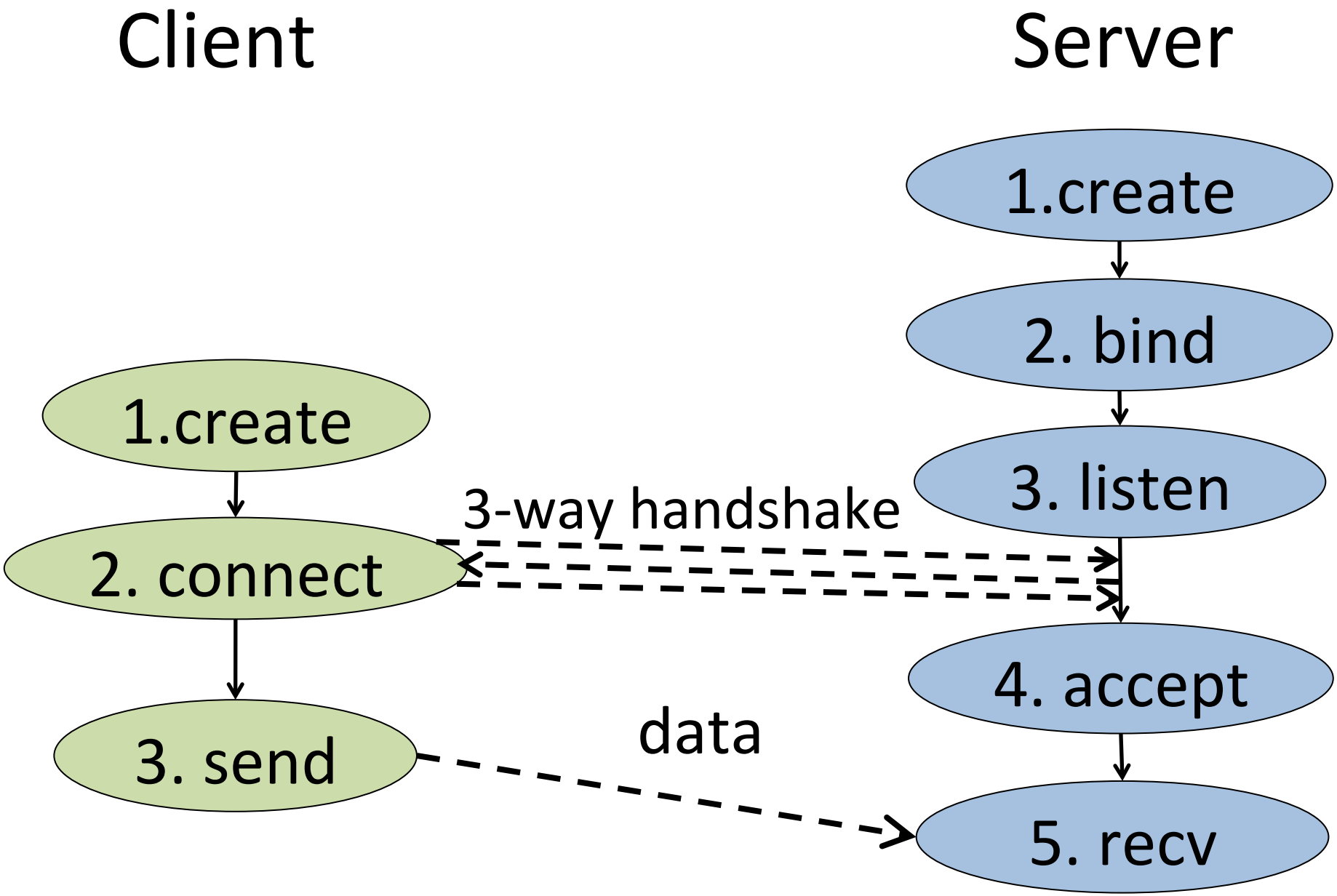


Client Program

```
import socket
import sys

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('10.0.0.1', 50000))
s.send("Hello Server!")
s.close()
```

Summary


Client
Server
1.create
2. bind
3. listen
4. accept
5. recv
1.create
2. connect
3. send
3-way handshake
data