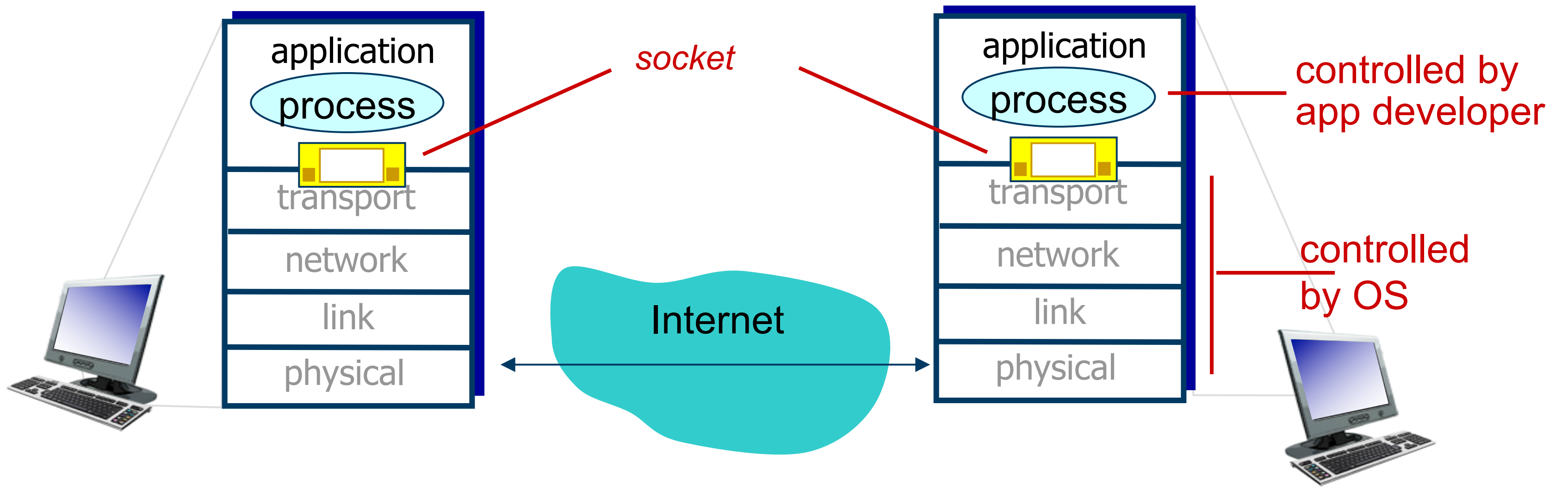
CSC5930/9010: Security and Privacy in Cyber-physical Systems

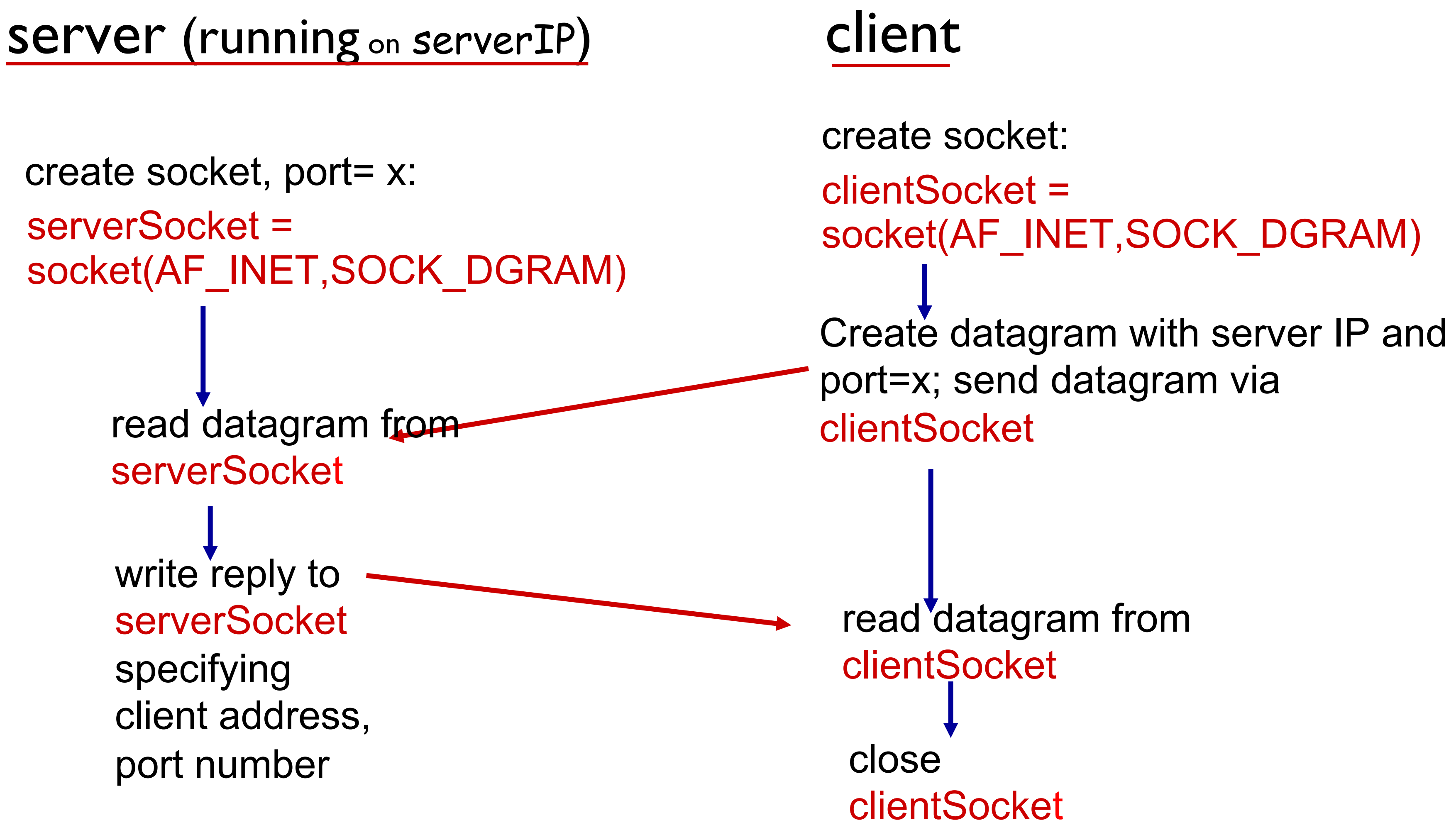
Socket Programming using Python

Socket Programming

- Sockets
  - A means of sending data over a network from one application to another.
  - A door between application process and end-end-transport protocol
- One socket node listens on a particular port and IP, while other socket reaches out to the other to form a connection.



Client/server socket interaction: UDP



server (running on serverIP)

create socket, port= x:
serverSocket =
socket(AF_INET,SOCK_DGRAM)

read datagram from
serverSocket

write reply to
serverSocket
specifying
client address,
port number

client

create socket:
clientSocket =
socket(AF_INET,SOCK_DGRAM)

Create datagram with server IP and
port=x; send datagram via
clientSocket

read datagram from
clientSocket

close
clientSocket

Example app: UDP Client

Python UDPClient

include Python's socket library → `from socket import *`

```
from socket import *

serverName = ‘hostname’

serverPort = 12000

clientSocket = socket(AF_INET,

                        SOCK_DGRAM)

message = input(’Input lowercase sentence:’)

clientSocket.sendto(str.encode(message),

                              (serverName, serverPort))

modifiedMessage, serverAddress =

                        clientSocket.recvfrom(2048)

print (bytes.decode(modifiedMessage))

clientSocket.close()
```

include Python's socket library

create UDP socket for server

get user keyboard input

Attach server name, port to message; send into socket

read reply characters from socket into string

print out received string and close socket

Example app: UDP Server

Python UDPServer

create UDP socket

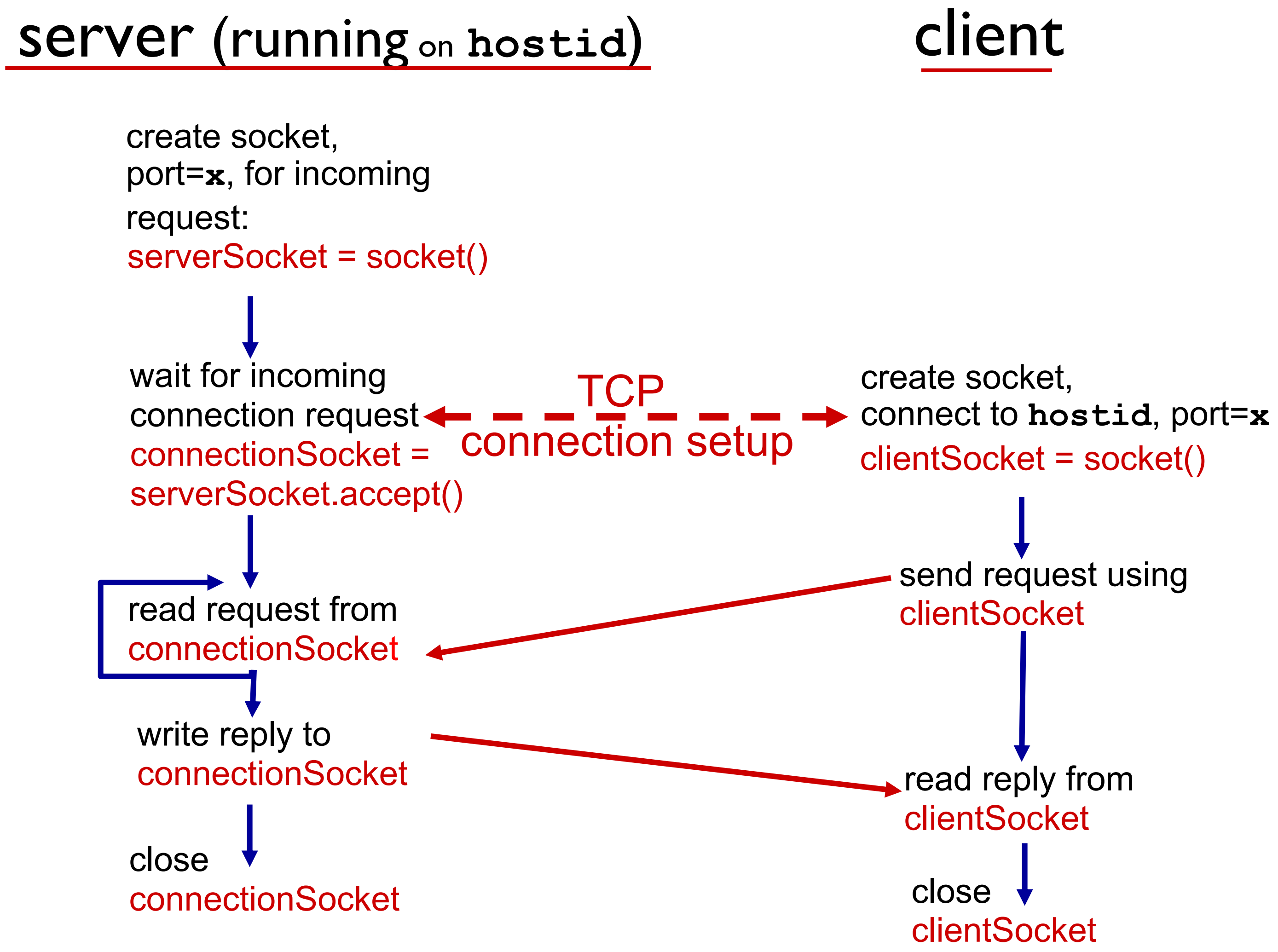
bind socket to local port number 12000

loop forever

Read from UDP socket into message, getting client’s address (client IP and port)

send upper case string back to this client

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind((‘’, serverPort))
print (“The server is ready to receive” )
while 1:
    message, clientAddress = serverSocket.recvfrom(2048)
    modifiedMessage = message.upper()
    serverSocket.sendto(modifiedMessage, clientAddress)
```



server (running on hostid)
client
create socket, port=x, for incoming request:
serverSocket = socket()
wait for incoming connection request
connectionSocket = serverSocket.accept()
TCP connection setup
create socket, connect to hostid, port=x
clientSocket = socket()
send request using clientSocket
read request from connectionSocket
write reply to connectionSocket
read reply from clientSocket
close connectionSocket
close clientSocket

Example app: TCP Client

Python TCPClient

create TCP socket for server, remote port 12000

No need to attach server name, port

```
from socket import *
serverName = ’servername’
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input(‘Input lowercase sentence:’)
clientSocket.send(str.encode(sentence))
modifiedSentence = clientSocket.recv(1024)
print (‘From Server: ’, bytes.decode(modifiedSentence))
clientSocket.close()
```

Example app: TCP Server

Python TCPServer

```
 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 1:
    connectionSocket, addr = serverSocket.accept()

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

create TCP welcoming socket → serverSocket = socket(AF_INET,SOCK_STREAM)

server begins listening for incoming TCP requests → serverSocket.listen(1)

loop forever → while 1:

server waits on accept() for incoming requests, new socket created on return → connectionSocket, addr = serverSocket.accept()

read bytes from socket (but not address as in UDP) → sentence = connectionSocket.recv(1024)

close connection to this client (but *not* welcoming socket) → connectionSocket.send(capitalizedSentence)

Echo Program - Server.py

```
#!/usr/bin/env python3

import socket
HOST = '127.0.0.1'  # Standard loopback interface address (localhost)
PORT = 65432        # Port to listen on (non-privileged ports are > 1023)

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    conn, addr = s.accept()
    with conn:
        print('Connected by', addr)
        while True:
            data = conn.recv(1024)
            if not data:
                break
            conn.sendall(data)
```

Echo Program - Client.py

```
#!/usr/bin/env python3

import socket

HOST = '127.0.0.1'  # The server's hostname or IP address
PORT = 65432        # The port used by the server

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    s.sendall(b'Hello, world')
    data = s.recv(1024)

print('Received', repr(data))
```