COMPUTER NETWORKS

Sivaraman Eswaran Ph.D.
Department of Computer Science and Engineering

COMPUTER NETWORKS

Application Layer

Sivaraman Eswaran Ph.D.
Department of Computer Science and Engineering

Unit – 2 Application Layer

Socket programming with TCP

Client must contact server

- server process must first be running
- server must have created socket (door) that welcomes client's contact

Client contacts server by:

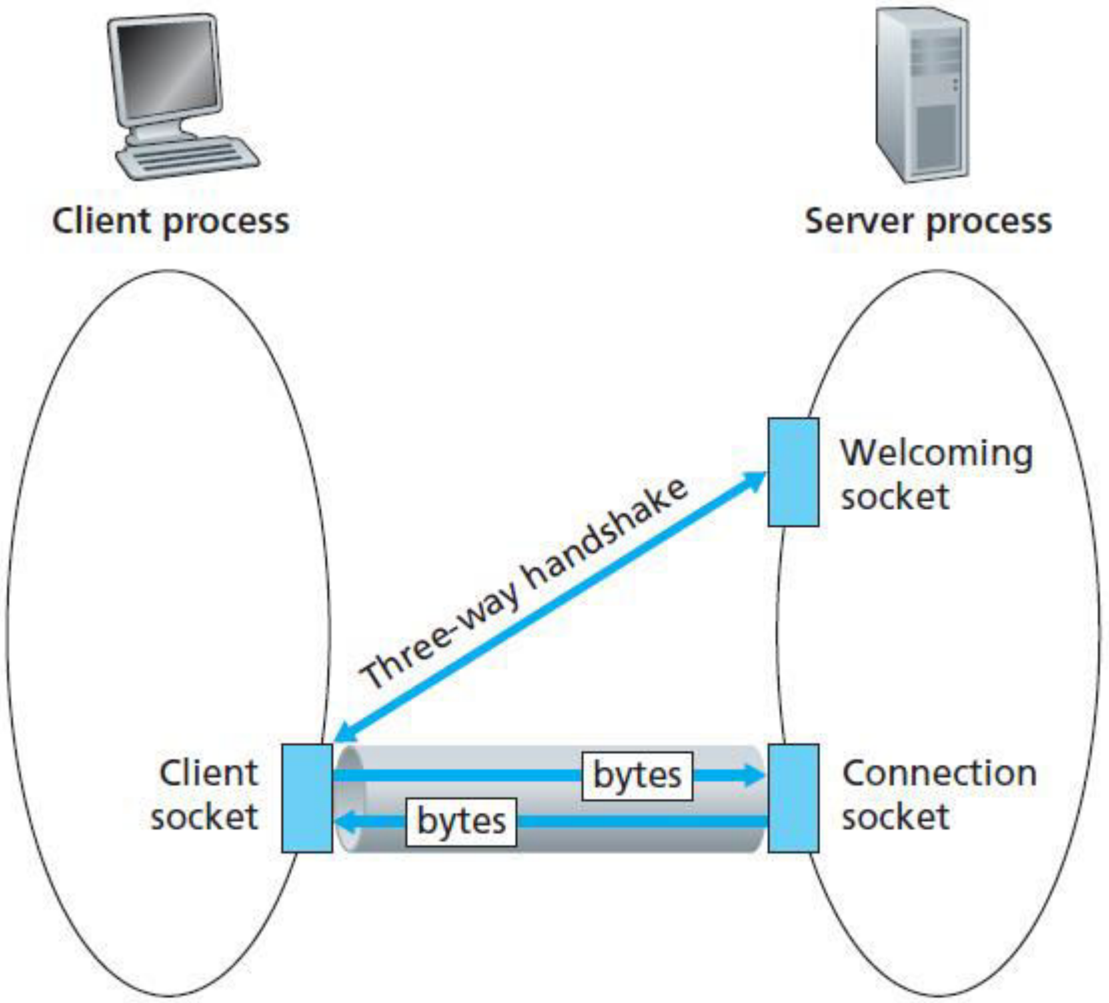
- Creating TCP socket, specifying IP address, port number of server process
- *when client creates socket:* client TCP establishes connection to server TCP

- when contacted by client, *server TCP creates new socket* for server process to communicate with that particular client
  - allows server to talk with multiple clients
  - source port numbers used to distinguish clients (more in Chap 3)

Application viewpoint

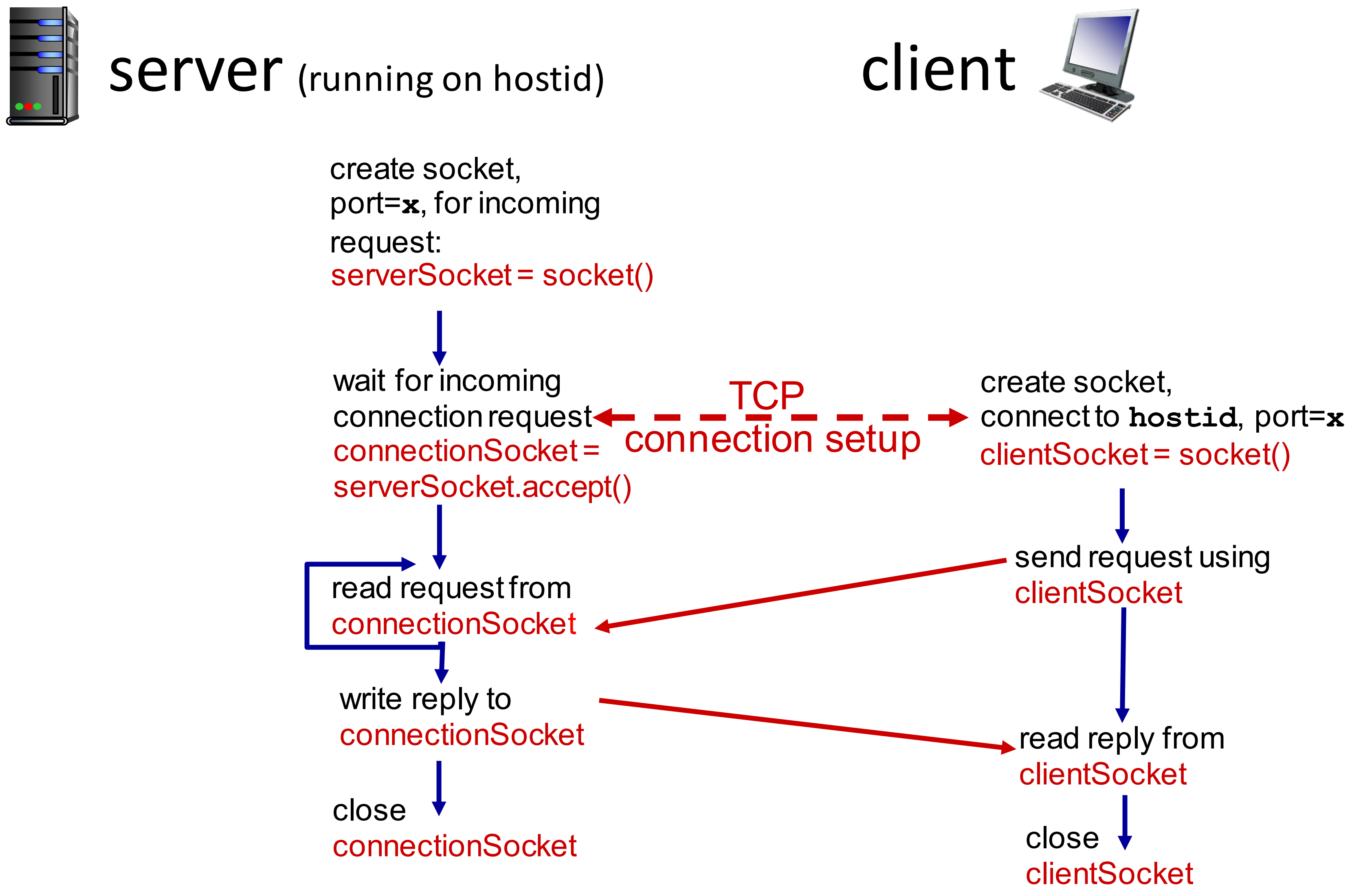
TCP provides reliable, in-order byte-stream transfer ("pipe") between client and server

The TCPServer Process has Two Sockets



COMPUTER NETWORKS

Client/server socket interaction: TCP


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

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

create TCP socket for server, remote port 12000 → `clientSocket = socket(AF_INET, SOCK_STREAM)`

No need to attach server name, port → `modifiedSentence = clientSocket.recv(1024)`

Example app: TCP server

Python TCPServer

```
 from socket import *
serverPort = 12000
```

create TCP welcoming socket

```
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((‘’,serverPort))
```

server begins listening for incoming TCP requests

```
serverSocket.listen(1)
print ‘The server is ready to receive’
```

loop forever

```
while True:
```

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

close connection to this client (but *not* welcoming socket)

```
    connectionSocket.close()
```

THANK YOU

Sivaraman Eswaran Ph.D.
Department of Computer Science and Engineering

sivaramane@pes.edu

+91 80 6666 3333 Extn 834