TCP socket programming

Sigurd Eskeland

Socket programming with TCP

Server

- server process must first be running
- server has created a TCP socket that waits for a contacting client

Client contacts server by:

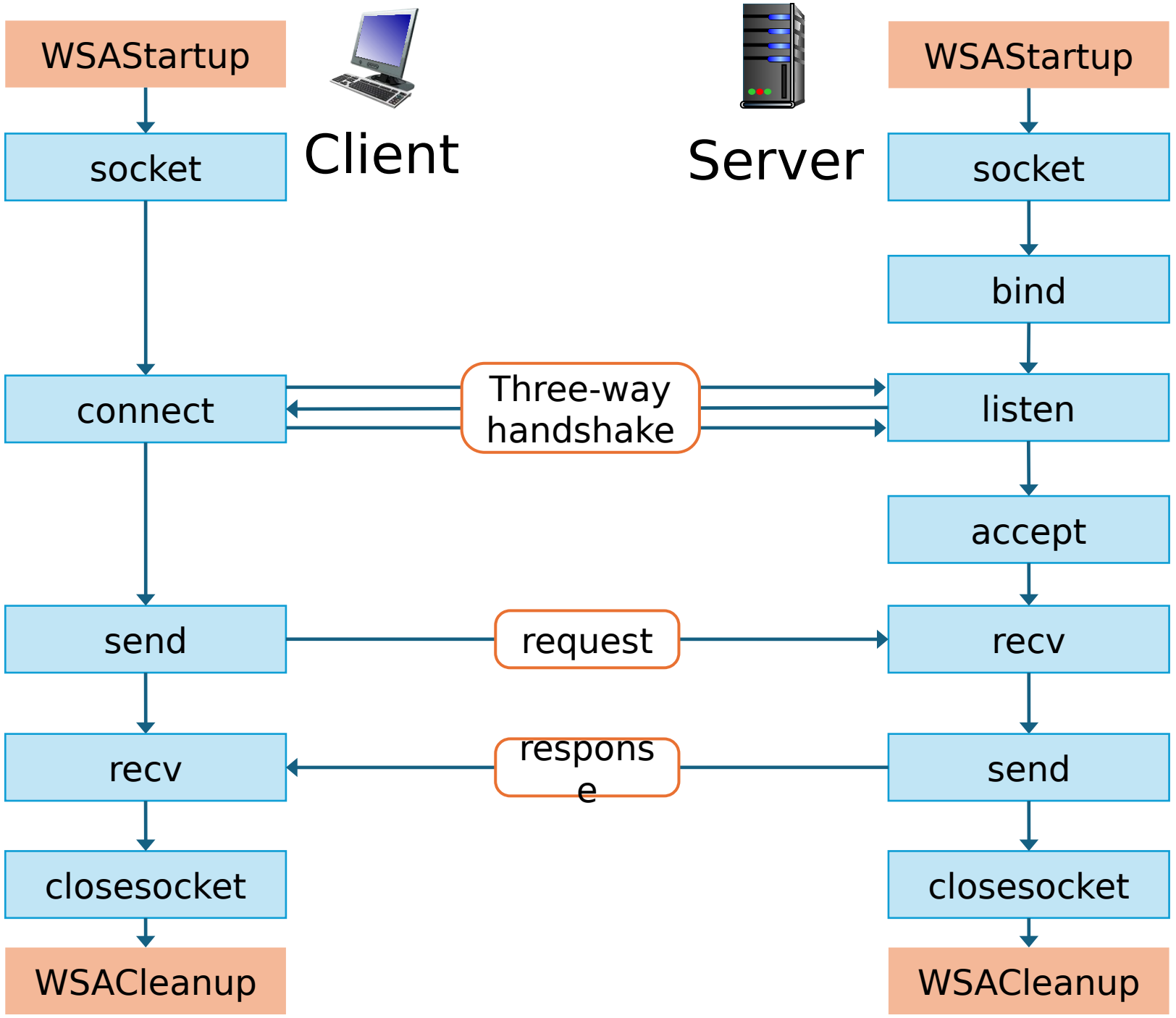
- Creating TCP socket, specifying IP address, port number of server process
- *When the server accepts a request,* a TCP connection is established

- 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 processes

TCP client/server socket interaction



TCP server

```
#include <winsock2.h>

WSADATA wsa;
SOCKET socketListen;
SOCKET socketTCPconnection;
struct sockaddr_in addressServer;
struct sockaddr_in addressClient;

WSAStartup(MAKEWORD(2,2), &wsa);

socketListen = socket(AF_INET,  SOCK_STREAM, 0 );

// Initialize address structure
addressServer.sin_addr.s_addr = INADDR_ANY;
addressServer.sin_family = AF_INET;
addressServer.sin_port = htons(nPortNumber);   // LSB -> MSB

bind(socketListen, (struct sockaddr*)& addressServer,
                                        sizeof(addressServer));
listen(socketListen, 5);

socketTCPconnection = accept(socketListen,
```

Holds address of a received message

Start use of Winsock DLL (*Ws2_32.dll*)

Create TCP-socket (IPv4)

Any address

Address is IPv4

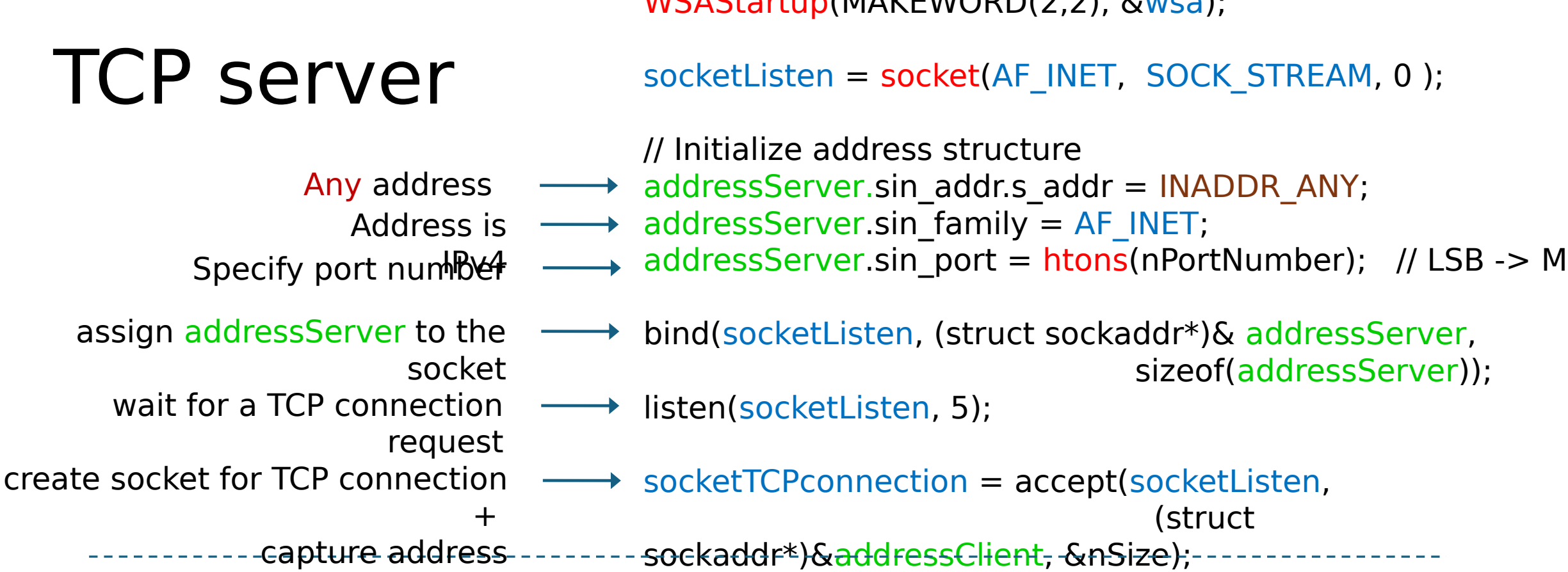
Specify port number

assign addressServer to the socket

wait for a TCP connection request

create socket for TCP connection:

TCP server


WSAStartup(MAKEWORD(2,2), &wsa);
socketListen = socket(AF_INET, SOCK_STREAM, 0);
// Initialize address structure
Any address
addressServer.sin_addr.s_addr = INADDR_ANY;
Address is IPv4
addressServer.sin_family = AF_INET;
Specify port number
addressServer.sin_port = htons(nPortNumber); // LSB -> M
assign addressServer to the socket
bind(socketListen, (struct sockaddr*)& addressServer, sizeof(addressServer));
wait for a TCP connection request
listen(socketListen, 5);
create socket for TCP connection + capture address
socketTCPconnection = accept(socketListen, (struct sockaddr*)&addressClient, &nSize);
recv(socketTCPconnection, sReceivedString, BUF_SIZE, 0);
send(...);
closesocket(socketTCPconnection); closesocket(socketListe
WSACleanup();

TCP client

Create TCP-socket (IPv4) →
Server's IP address – convert text-address to binary form →
Address is IPv4 →
Server's port number →
connect →
Send message to server address →
Receive message from server →
Close the socket →

```
SOCKET socketClient;

struct sockaddr_in addressServer;
char sTargetAddress[] = "127.0.0.1";    // loopback address

WSAStartup(MAKEWORD(2,2), &wsa);

socketClient = socket(AF_INET, SOCK_STREAM, 0 );

// Initialize targe taddress structure
InetPton(AF_INET, __TEXT(sTargetAddress), &
addressServer.sin_addr);
addressServer.sin_family = AF_INET;
addressServer.sin_port = htons(nPortNumber);   // LSB ->
MSB

connect(socketClient, (struct sockaddr *)&addressServer ,
sizeof(addressServer) );

send(socketClient, sMessage, strlen(sMessage), 0);

recv( ... );

closesocket(socketClient);
```

Firewall blocking TCP port numbers

- Scanning for open TCP port numbers

```
C:\Users\Sigurd.PCL342>nmap 10.0.0.9 -p 54555
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-12 14:04 W. Europe Standard Time
Nmap scan report for 10.0.0.9
Host is up (0.0020s latency).

PORT      STATE    SERVICE
54555/tcp filtered unknown
MAC Address: E8:C8:29:85:A3:3B (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds
```

Active TCP connections

```
C:\Users\sigurde>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    128.39.201.58:12293    20.50.201.204:https    ESTABLISHED
  TCP    128.39.201.58:12294    20.189.173.23:https    ESTABLISHED
  TCP    [::]:135               UIA5CG4081L51:0        LISTENING
  TCP    [::]:445               UIA5CG4081L51:0        LISTENING
```

- netstat –b lists the process names
  - Requires elevated command prompt