Code Exploration: Sockets

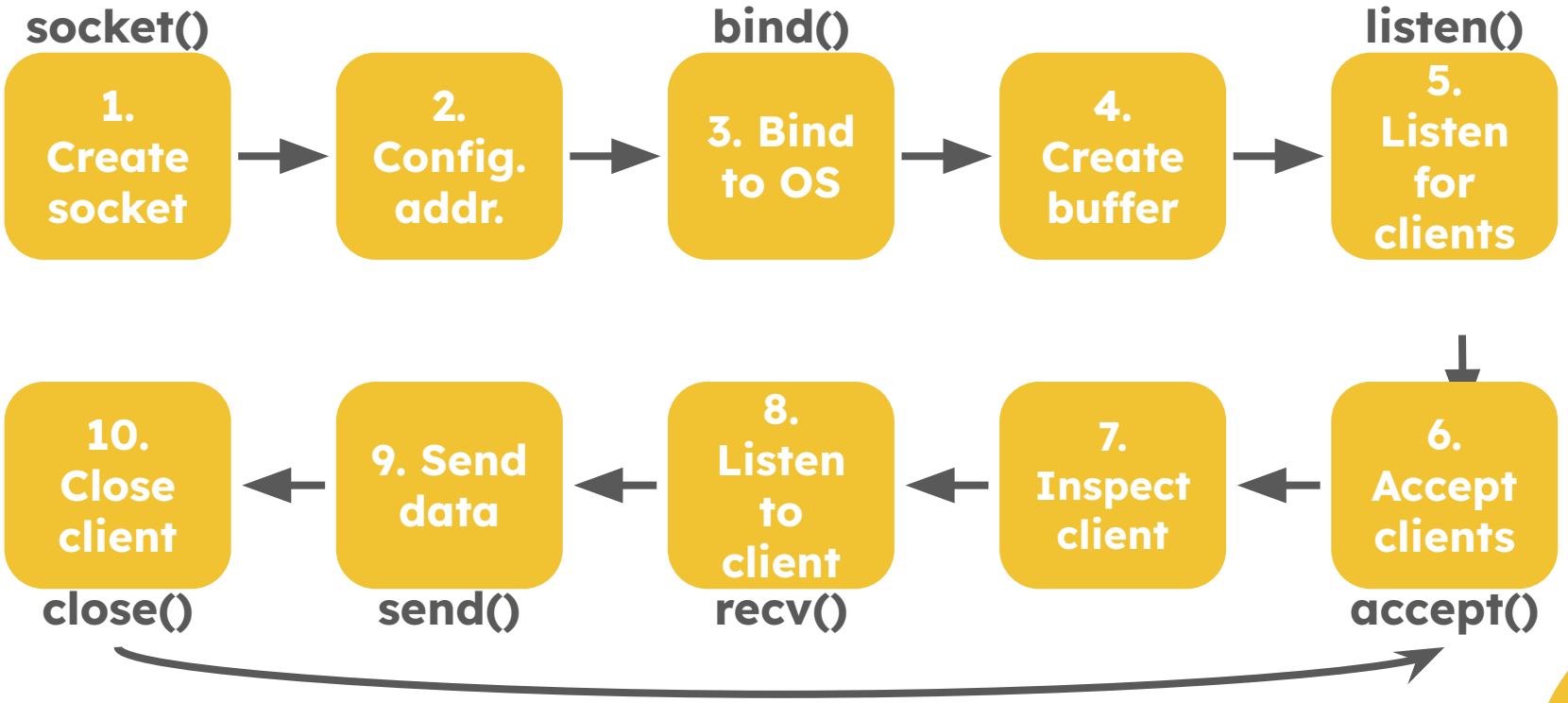
Computer Network Fundamentals

George Varghese - Fall 2024
Discussion - Week 2

Socket Programming Basics

...continued

Stream (TCP) Server Steps



Stream (TCP) Server Steps

listen()

5. Listen for clients

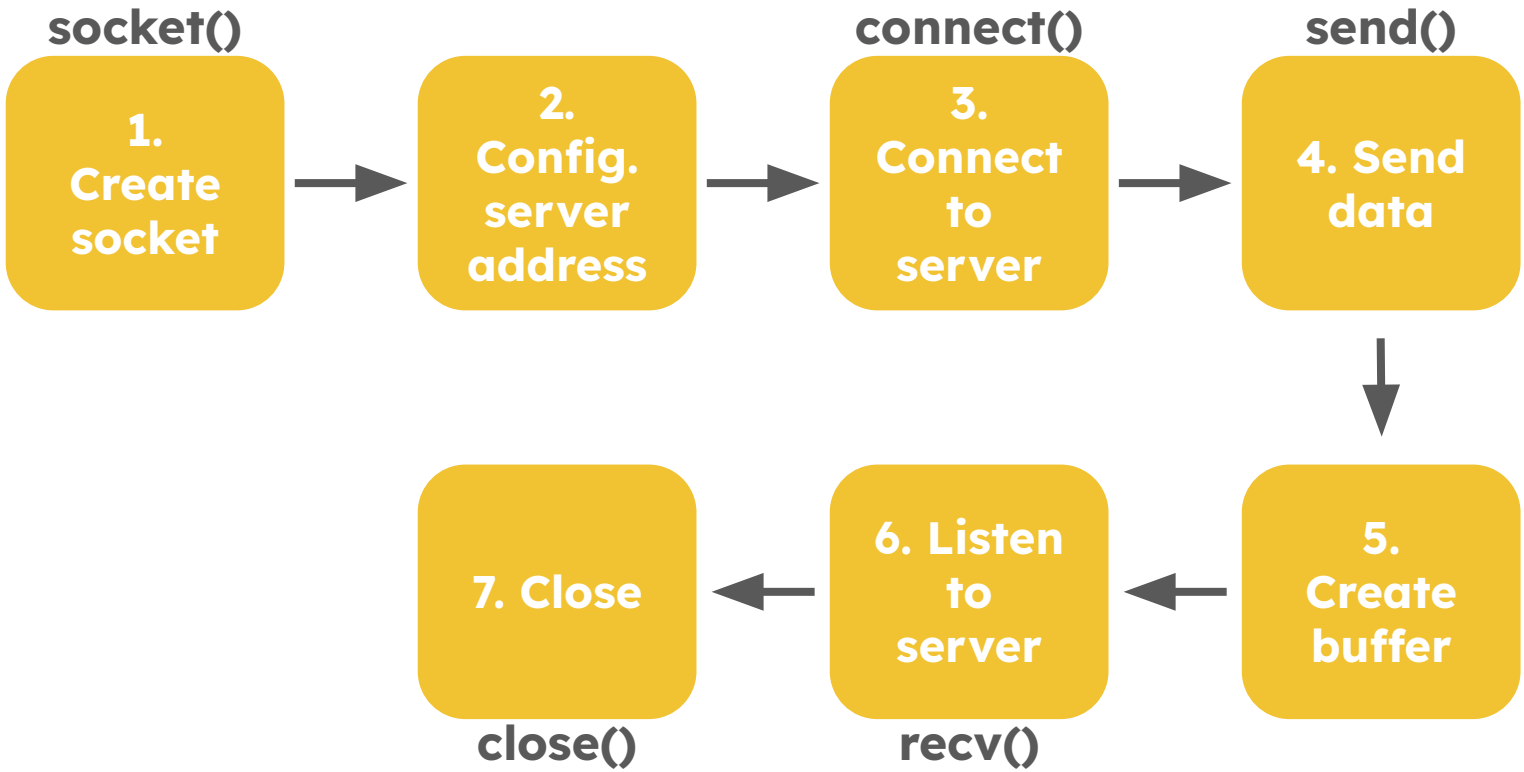
will listen on the **socket** that you **bind**ed a port to

accept()

6. Accept clients

will return a *completely different* **socket** with a new port

Stream (TCP) Client Steps



Code Exploration

NetSift

It’s Monday, June 27th, 2005. Professor George Varghese, then a professor at UCSD, just sold his startup *NetSift* to Cisco for a large amount.

However, in CSE 123 (UCSD’s CS 118), it’s a normal day. Professor Varghese comes up to the stage and starts his lecture on bit stuffing.

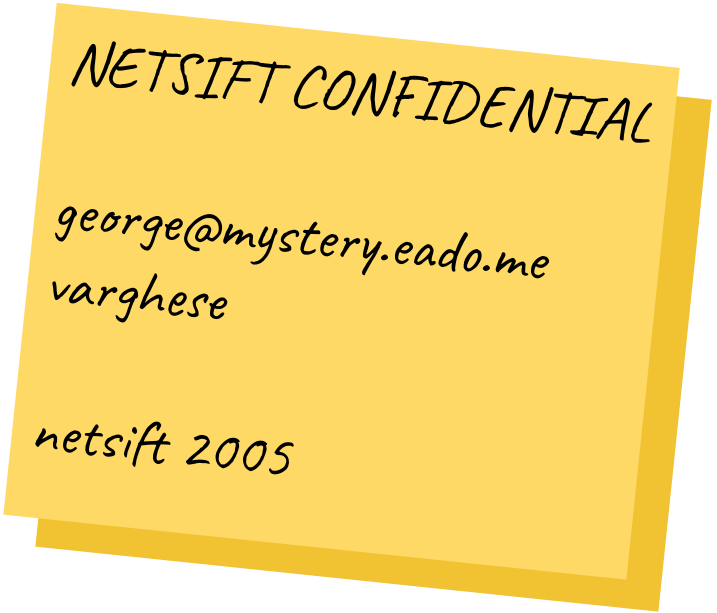
Unfortunately, he forgot his microphone…

An unfortunate slip-up

He gave one of his students his office keys and asked them to go fetch the microphone from his office.

While in the office, a notepad caught this student's attention.

Suspecting that the professor was on the cusp of something big, this student decides to give this note to premier networking experts...



// instructions.txt

For Project 0, there’s a lot of setup involved to get your systems up and running.

Luckily for this activity, all you need is **ssh**!

// instructions.txt

Split up into groups of **3**.
Each group will be assigned a number.

Connect using `**ssh**` to your infiltration server:
$ ssh george@mystery.eado.me -p 22<group number>

For example, group number **1** will connect using:
$ ssh george@mystery.eado.me -p 22**01**

(When you first connect, `ssh` will ask about fingerprints.
Make sure to enter **yes** to proceed.)

// instructions.txt

When connecting to sockets, you must provide IP addresses, but hints give you hostnames. Use the `dig` command to find the IP address.

For example, if you want to find the IP address of **netsift**:

```
$ dig netsift
[...]
netsift.                600     IN      A       10.0.0.47
[...]
```

// instructions.txt

Use the command `**./socket**` to connect to servers (and make them too). You'll need to use your sockets API knowledge to use this tool.

These hosts use HTTP.
- You must send **GET /<request>**

// tips.txt

Connecting: $ ssh george@mystery.eado.me -p 2201
    (type yes when asked about fingerprint)
Finding IPs: $ dig netsift
Using sockets: $ ./socket
HTTP:
 - Start with **GET /**
 - You need to create a new socket each request
Server:
 - **socket**, bind, listen, accept, recv, send
Client:
 - **socket**, connect, send, recv

On recv, **try 3000 bytes**.

If you don't know how to use a command,
type it.