

Sockets

Apr. 8rd

Client to Server

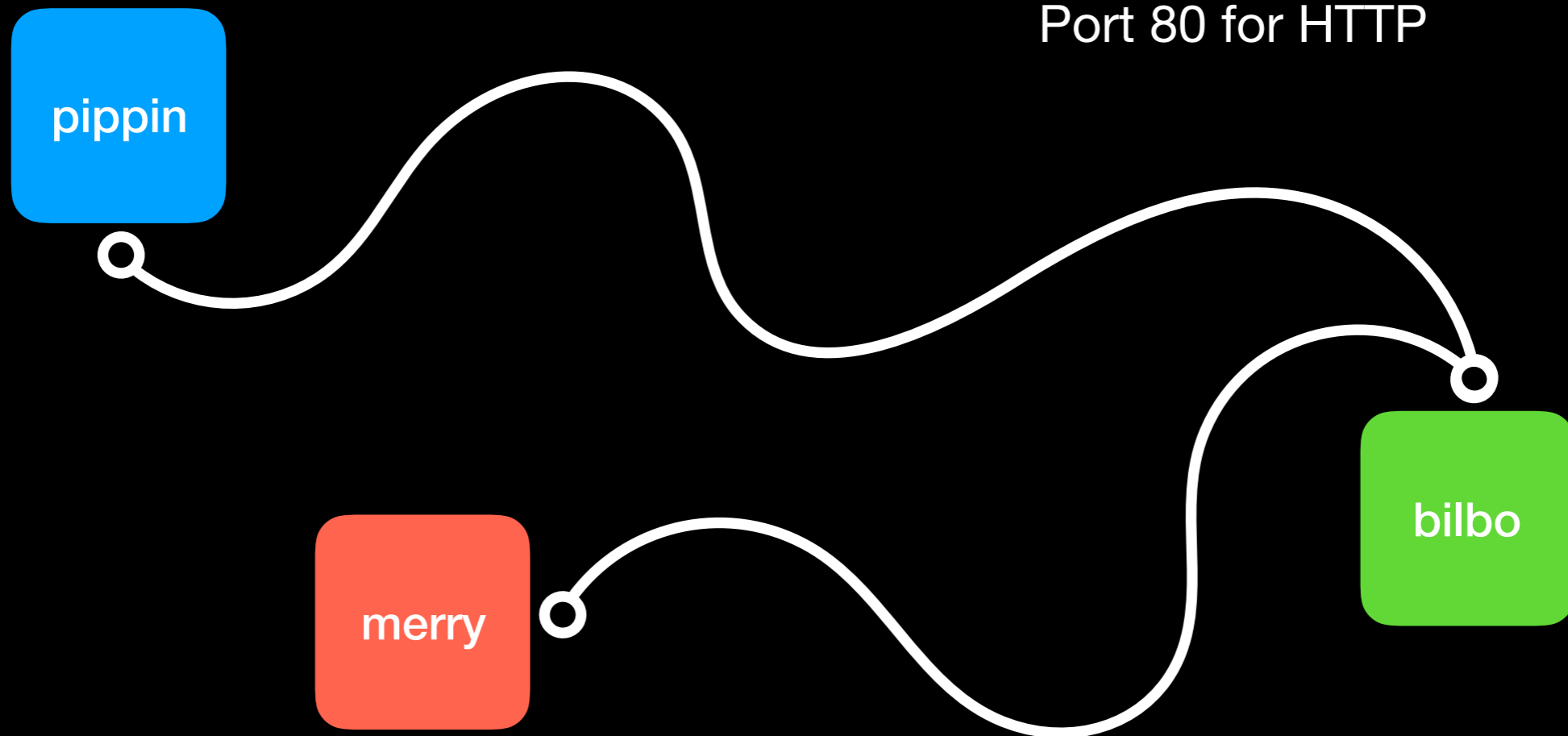
Port — reference a service on a server

Port 22 is for SSH

Port 23 is for telnet

Port 25 for SMTP

Port 80 for HTTP



Client to Server

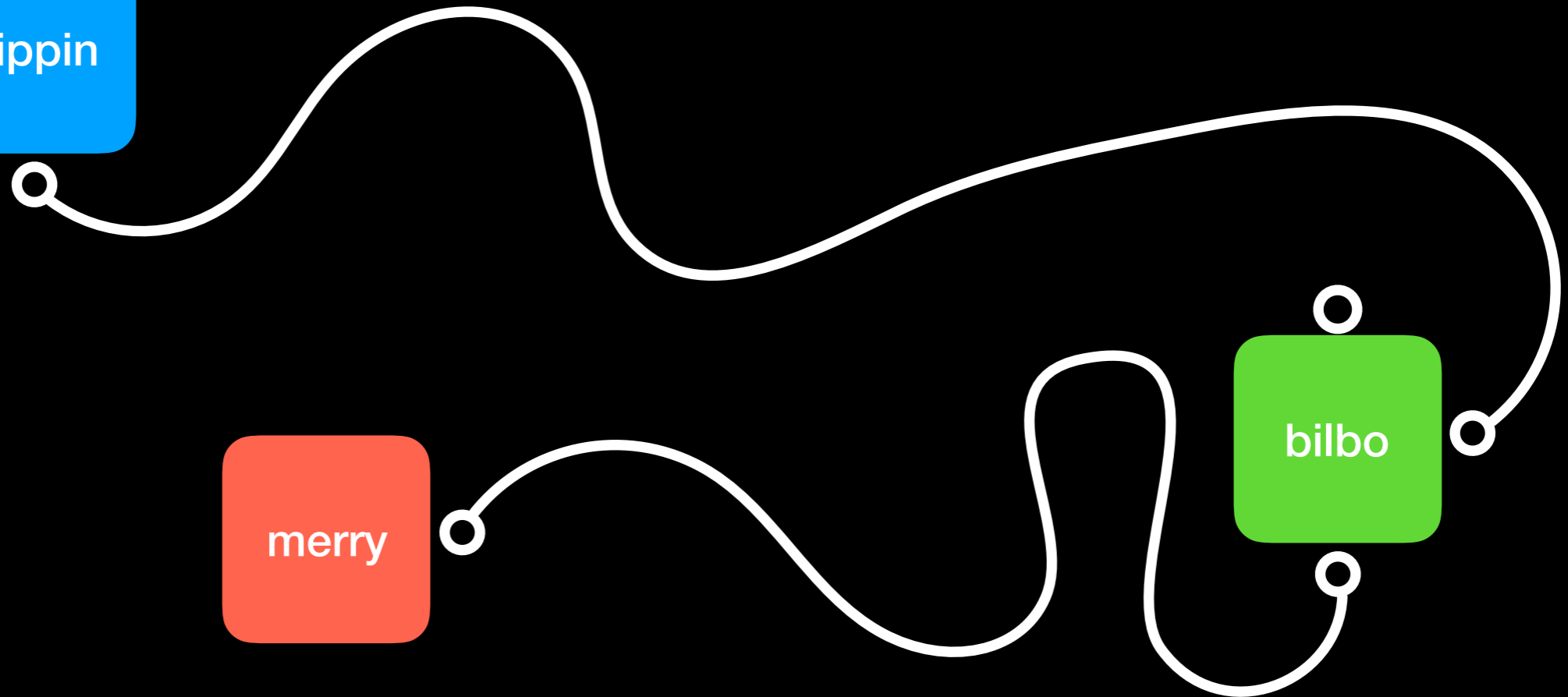
CLIENT

Will create a socket to connect to the server



SERVER

A socket will listen on a port and then create connections for each client



Sockets

Client:

open a *socket*
connect to a server
read/write data
close

Server:

open a *socket*
bind a socket to a name
listen for connections
accept new connections
read/write data
close

Client

Create a new socket

```
int socket(int domain, int type, int protocol)
```

```
int connect(int socket,  
            const struct sockaddr *address,  
            socklen_t address_len) Connect to a server
```

Server

int socket(**int** domain, **int** type, **int** protocol)

Create a new socket

int bind(**int** socket,

const struct sockaddr *address,

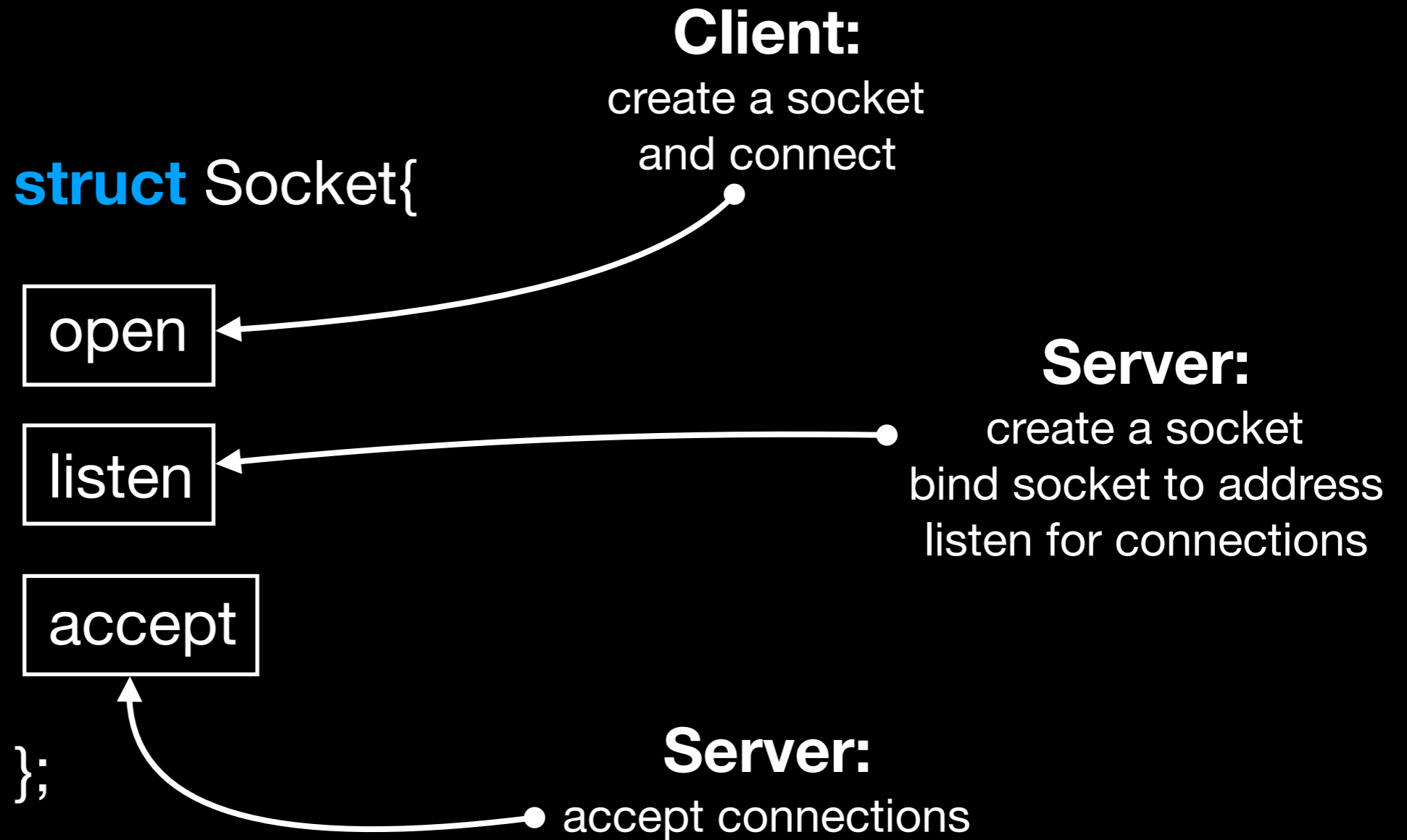
socklen_t address_len) Associate socket with an address

int listen(**int** socket, **int** backlog) Listen for connections

int accept(**int** socket,
struct sockaddr *address,
socklen_t address_len)

Accept the
connection, creating a
new socket for the
connection

Abstraction



open

Output:

nothing (or maybe an error)

Input:

an address and port

```
void Socket::open(string hostname, string port){
```

```
    fd = socket( /*stuff goes in here*/ );
```

```
}
```

could be an int
if you want

returns a file
descriptor

stored in the struct

open

Output:

nothing (or maybe an error)

Input:


an address and port

```
void Socket::open(string hostname, string port){
```

```
    fd = socket( /*stuff*/ );
```

```
     connect( fd, /*stuff*/ );  
}
```

returns if there was
an error



Stuff

How to convert a hostname
and port into a sockaddr?

getaddrinfo system call

```
void Socket::open(string hostname, string port){
```

```
    getaddrinfo(hostname.c_str(), port.c_str(), &hints, &info);  
  
    fd = socket(info->ai_family, info->ai_socktype, 0);  
    connect( fd, info->ai_addr, info->ai_addrlen);  
}
```

returns if there was
an error

what are these?

Stuff

```
void Socket::open(string hostname, string port){  
    struct addrinfo hints;  
    struct addrinfo *info;  
    getaddrinfo(hostname.c_str(), port.c_str(), &hints, &info);  
    fd = socket(info->ai_family, info->ai_socktype, 0);  
    connect( fd, info->ai_addr, info->ai_addrlen);  
    freeaddrinfo(info);  
}
```

allocates memory!

free the memory

Stuff

```
void Socket::open(string hostname, string port){  
    struct addrinfo hints;  
    struct addrinfo *info;  
    memset(&hints,0, sizeof(hints));  
    hints.ai_family = AF_INET6; //or AF_INET for IPv4  
    hints.ai_socktype = SOCK_STREAM;  
    getaddrinfo(hostname.c_str(), port.c_str(), &hints, &info);  
    fd = socket(info->ai_family, info->ai_socktype, 0);  
    connect( fd, info->ai_addr, info->ai_addrlen);  
    freeaddrinfo(info);  
}
```