

Multithreaded Echo Client/Server

TJ Maynes, Chris Migut

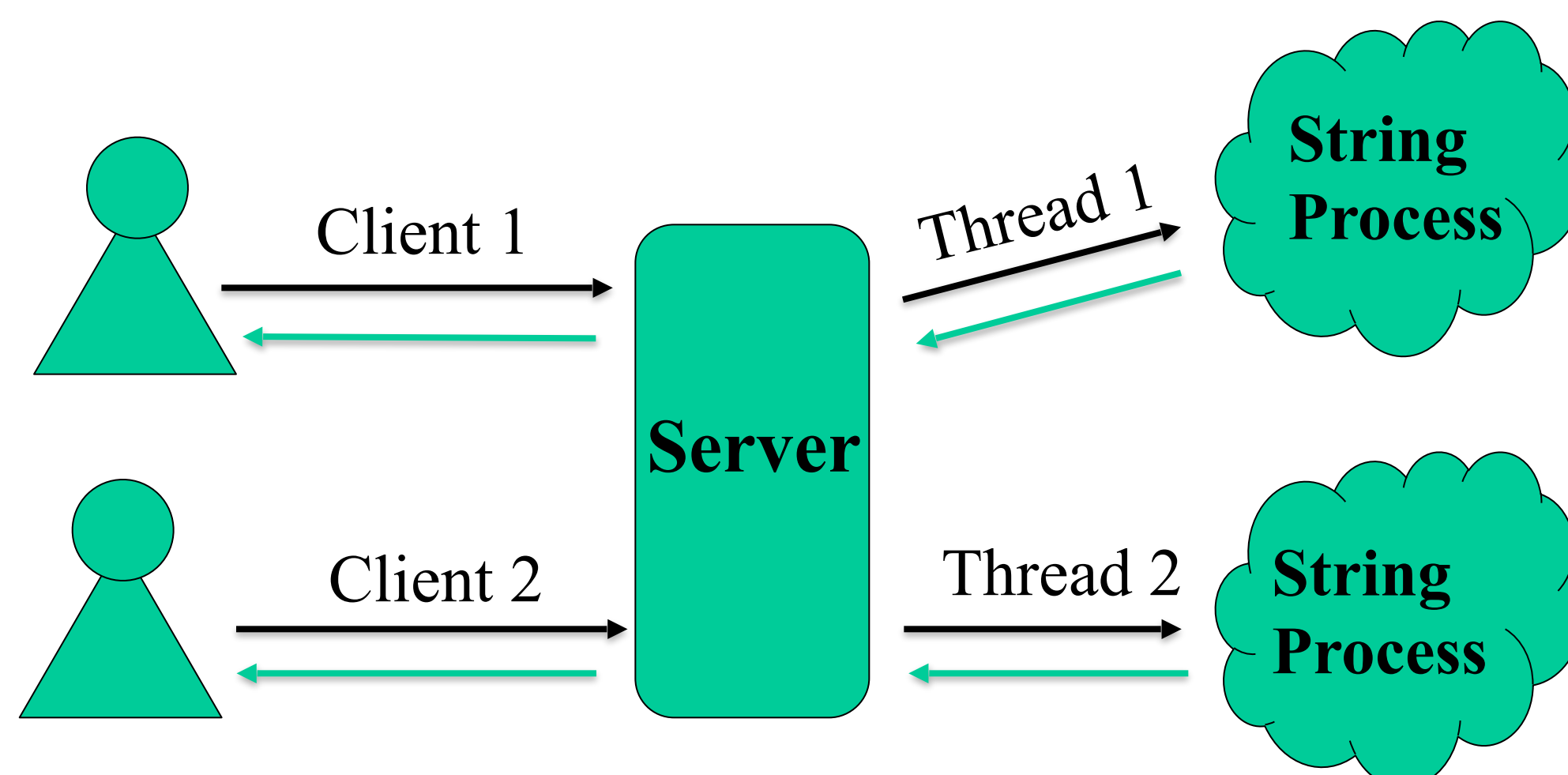
Project Description:

This project will demonstrate the importance of multithreading, semaphores, sockets, and shared **memory** in building a client/server program. The basic idea of the project is to implement a client to send a 12 character message to a server and the server respond with the 12 character message in reverse.

Project Goals:

- Implement multithreading to create a thread for each incoming accepted connection with a client (on the server).
- Use of semaphores to make changes the number of currently connected clients (in shared memory) without affecting other threads using the shared memory buffer (on the server).
- Creating of sockets to build TCP connections between client and server to share information from shared memory.
- The server will receive a message and return the reversed version of the message back to the client.
- The TCP/IP client-server relationship will be implemented in the C programming language.

Project Diagram:



Background Information:

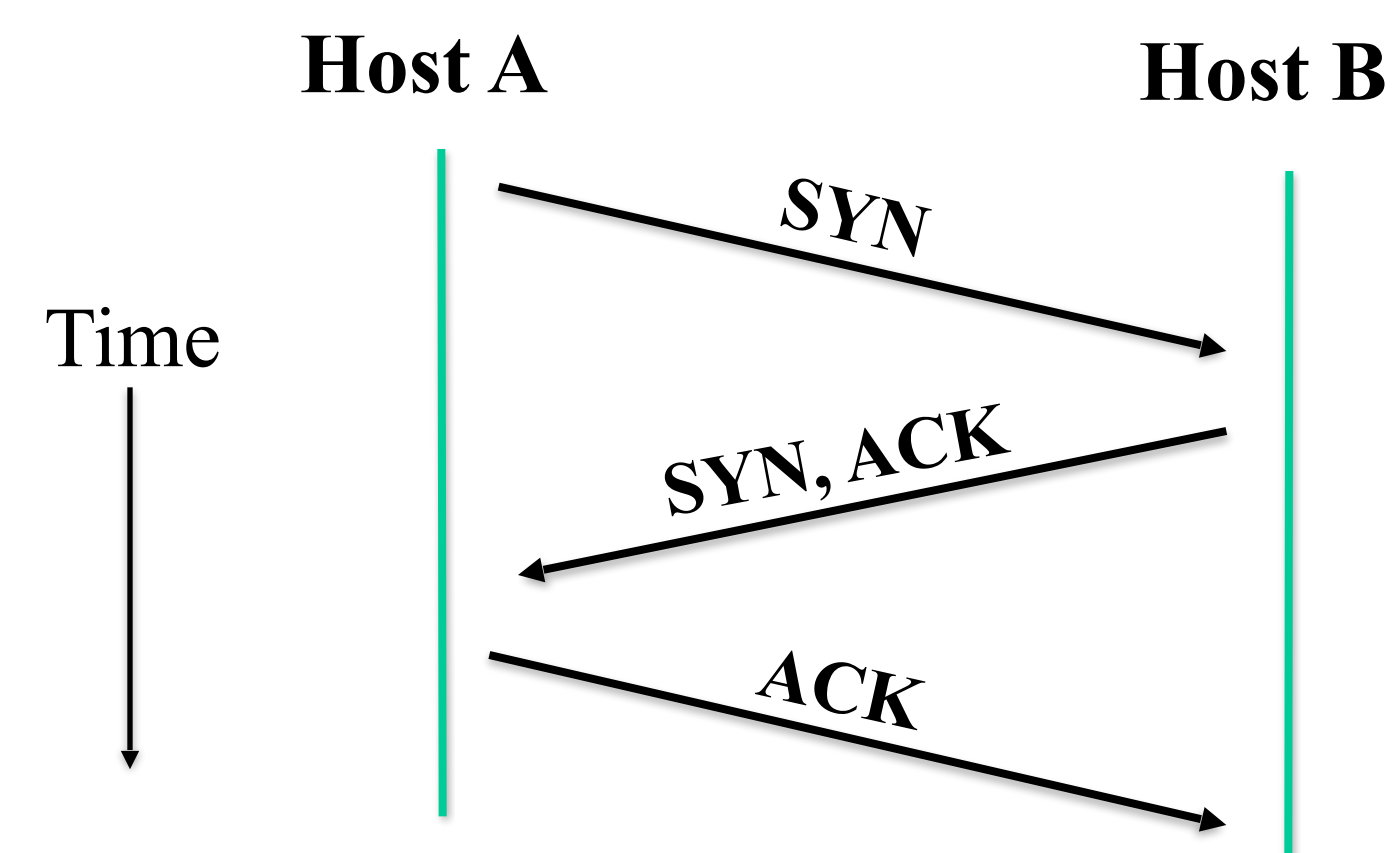
TCP/IP Client-Server relationship

TCP protocol is a connection-oriented protocol used by devices to communicate over a network. By having TCP be a connection-oriented protocol provides a quality of service to the different parties involved in the communication.

Services provided:

Three-way handshake:

Method used by TCP to negotiate a connection between two devices to setup a TCP/IP connection on a network. Once the session between the parties is finished, a similar method is used to close the connection between the devices.



SYN : Synchronize
ACK : Acknowledgement

Quality of Service methods

Flow Control:

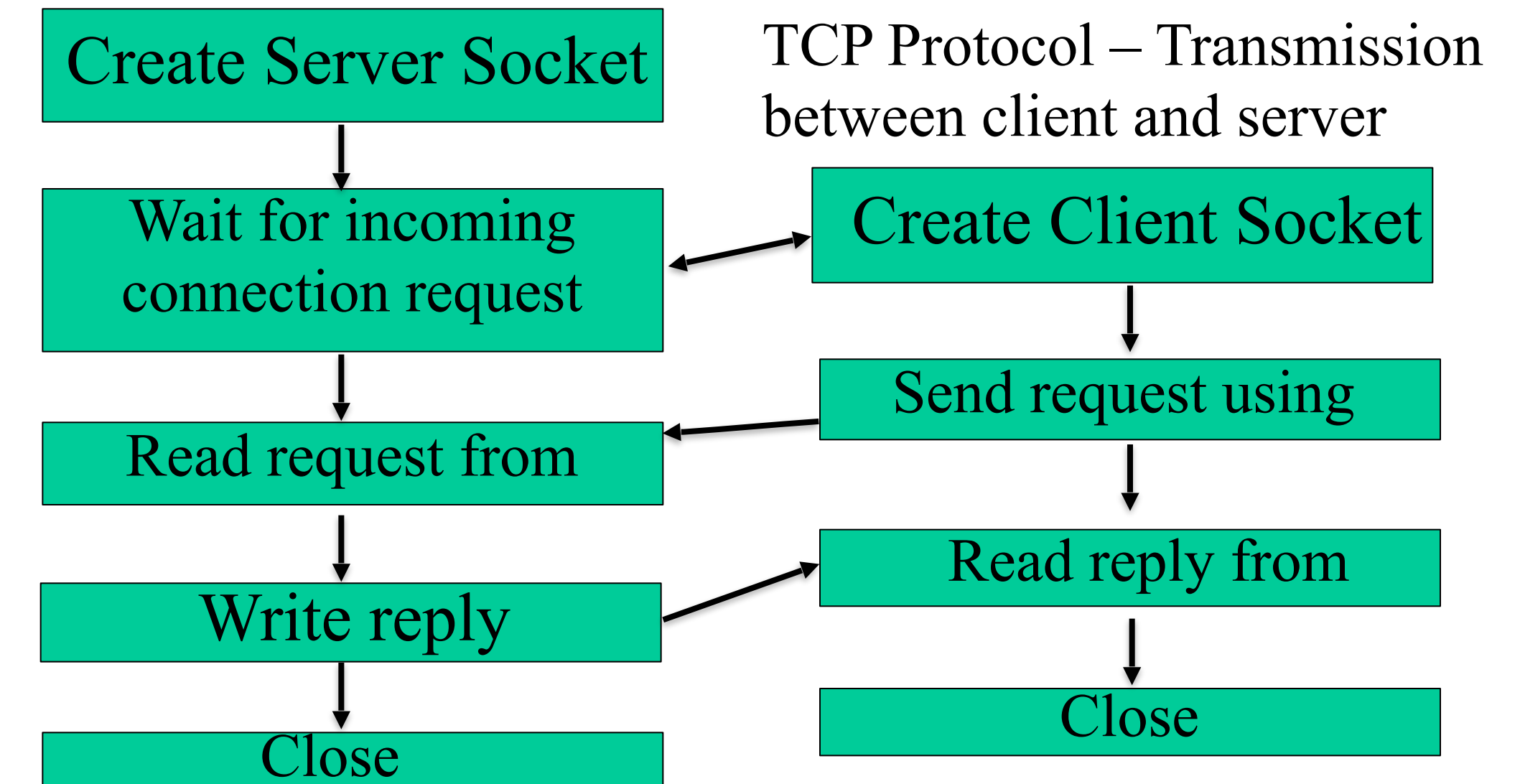
Stop-and-Wait
Sliding-Window

Error Control:

Go-Back-N
Selective Repeat

More information on TCP:

- RFC 793, RFC 1122, RFC 1323, RFC 2018, RFC 2581



Results:

```
Welcome to the Server!
Waiting for incoming connections...
....Accepted Connection....
hellotj
jtolleh
....Client disconnected....

....Client disconnected....
....Accepted Connection....
heyChris
sirhcyeh
....Client disconnected....
```

```
Connected to server: 127.0.0.1 on Port: 8080

Enter a message to send (up to 10 characters): hellotj

Message received from the server: jtolleh

Total number of client connections (at start of client program): 1

Thank you for using this program!
```

```
Welcome to the Client Connection Program!
Connected to server: 127.0.0.1 on Port: 8080

Enter a message to send (up to 10 characters): heyChris

Message received from the server: sirhcyeh

Total number of client connections (at start of client program): 2

Thank you for using this program!
```