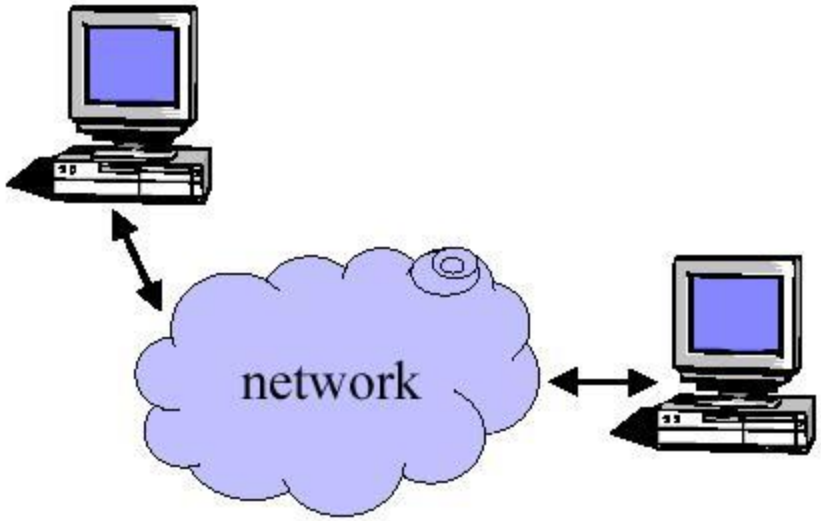
Introduction to Sockets

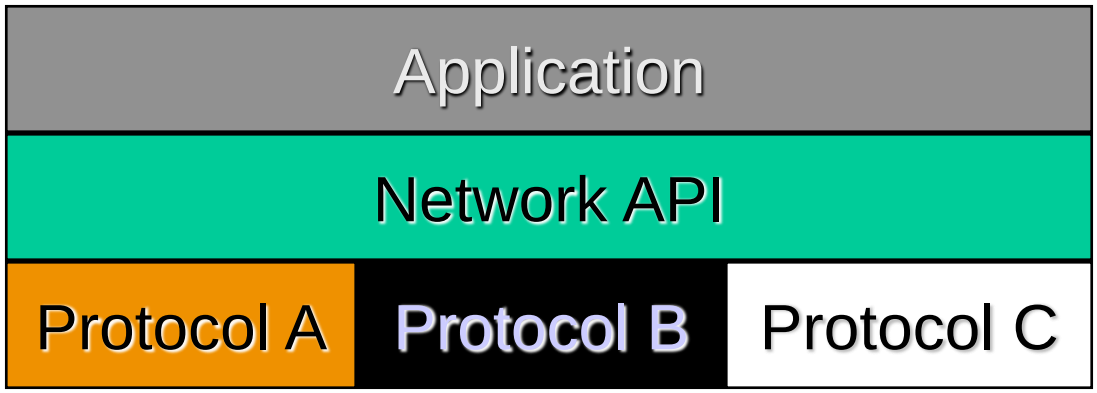
Why do we need sockets?

Provides an abstraction for interprocess communication



Definition

- The services provided (often by the operating system) that provide the interface between application and protocol software.



Functions

- Define an “end- point” for communication
- Initiate and accept a connection
- Send and receive data
- Terminate a connection gracefully

Examples

- File transfer apps (FTP), Web browsers
- (HTTP), Email (SMTP/ POP3), etc…

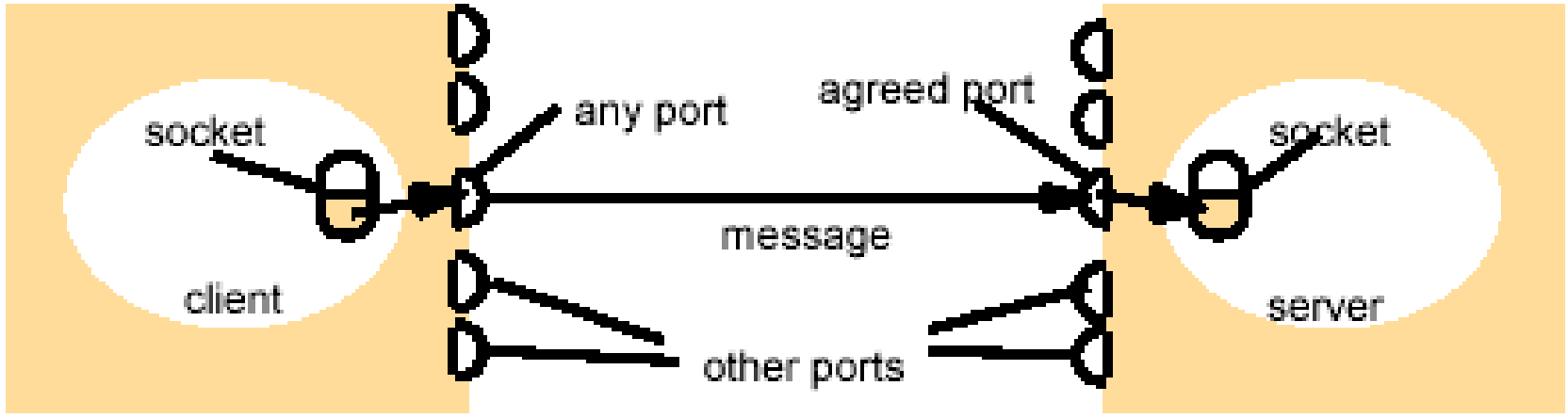
Types of Sockets

- Two different types of sockets :
  - stream vs. datagram

- Stream socket :(*a. k.* a. connection- oriented socket)
  - It provides reliable, connected networking service
  - Error free; no out- of- order packets (uses TCP)
  - applications: telnet/ ssh, http, …

- Datagram socket :(*a. k.* a. connectionless socket)
  - It provides unreliable, best- effort networking service
  - Packets may be lost; may arrive out of order (uses UDP)
  - applications: streaming audio/ video (realplayer), …

Addressing

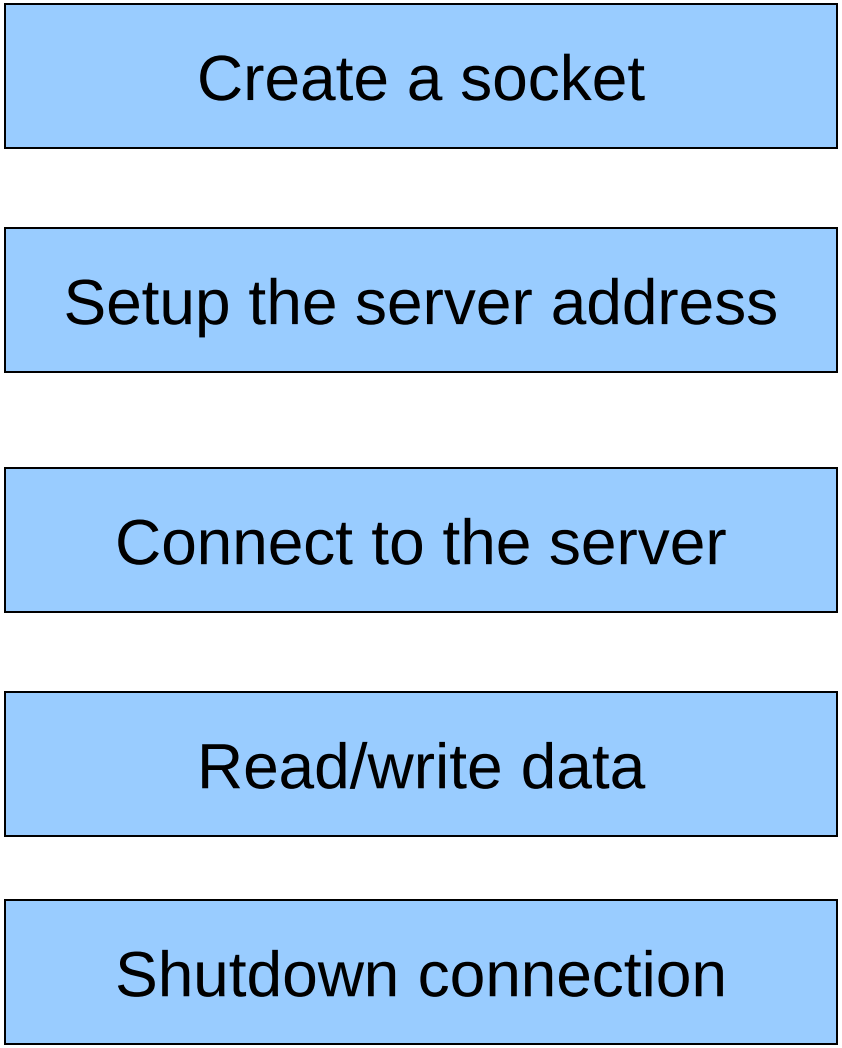


Client ⟷ Server

Addresses, Ports and Sockets

- Like apartments and mailboxes
  - You are the application
  - Your apartment building address is the address
  - Your mailbox is the port
  - The post-office is the network
  - The socket is the key that gives you access to the right mailbox

Client – high level view



Server – high level view

Create a socket

Bind the socket

Listen for connections

Accept new client connections

Read/write to client connections

Shutdown connection