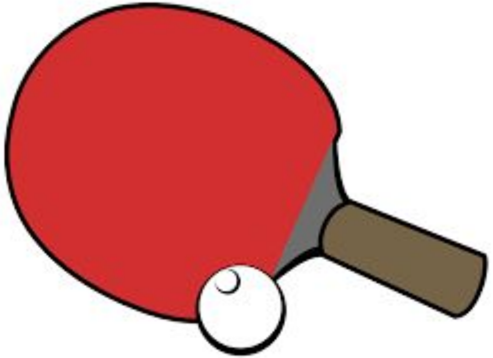
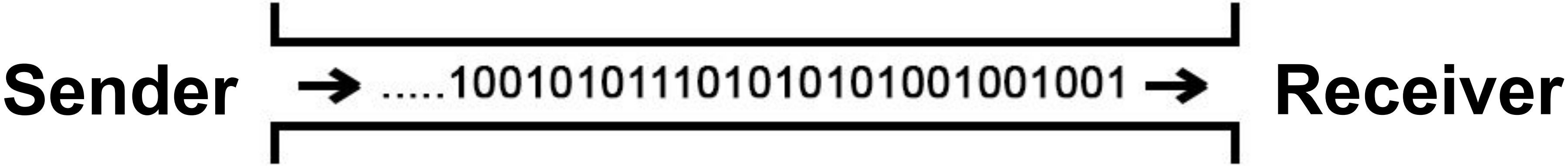
Ping Pong



Pipes & Flow Control

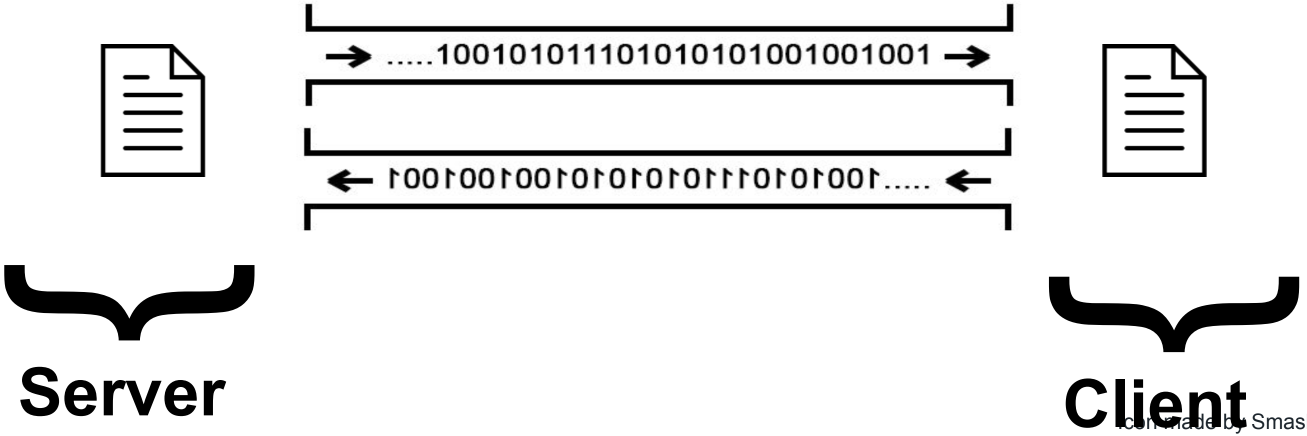
- Pipes are one-directional streams of data that operates within a single machine.
- When the pipe is full, the sender will be suspended until the receiver has cleared space for new data
- Can be accessed either through a ***Shared memory (Fork)*** or a ***File***



Sender → .....10010101110101010101001001001 → **Receiver**

Sockets - When you need an answer

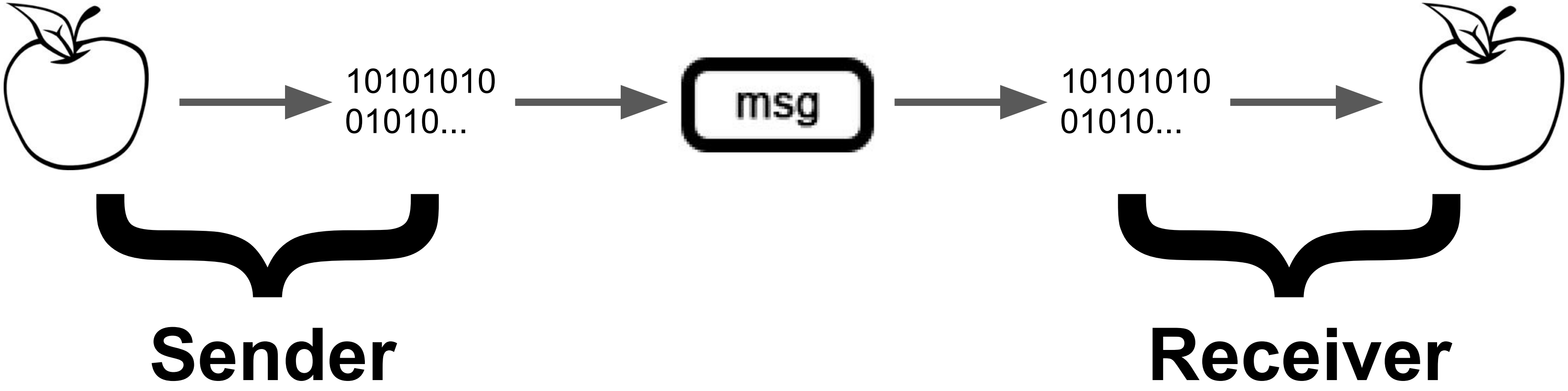
- One limitation of pipes is that they are **one way**
- Sockets provide **two way** connections and **location transparency** (you can send things to you friends!)
- Supports communication both locally and over a network



Icon made by Smashicon from www.flaticon.com

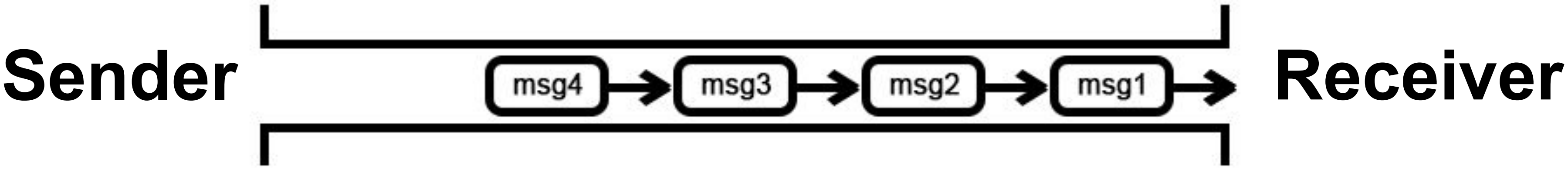
Marshalling

- Everything is computers is encoded as **ones** & **zeros**
- Hence we can send any datatype as a binary message
- The way we encode and decode datatypes is known as **marshalling**



Datagrams

- Every messages includes the length of that actual message.
- This gives us the opportunity to easily work with more complex messages



2017-12-18

We can easily do a ***fork()*** and then set up stdin and stdout for the two processes to communicate through a so called **pipe**.
How can we achieve the same for two processes although they're not created using a fork()? Meaning we still want one process create a pipe that another process can read from.

2017-12-18

We can easily do a ***fork()*** and then set up stdin and stdout for the two processes to communicate through a so called **pipe**.
How can we achieve the same for two processes although they’re not created using a fork()? Meaning we still want one process create a pipe that another process can read from.

Answer:

One of the processes can register a pipe with a agreed upon file name, using the ***mkfifo*** command. Now the other pipe can open that pipe as a file, using the specified name.

2017-06-07

A so called ***pipe*** is a simple way to send data from one process to another. It does have its limitations and a better way is to use so called ***sockets***. If we use a stream socket between two processes we will have several advantages. Describe *two* advantages that a steam socket gives us that we will not have if we use a pipe.

2017-06-07

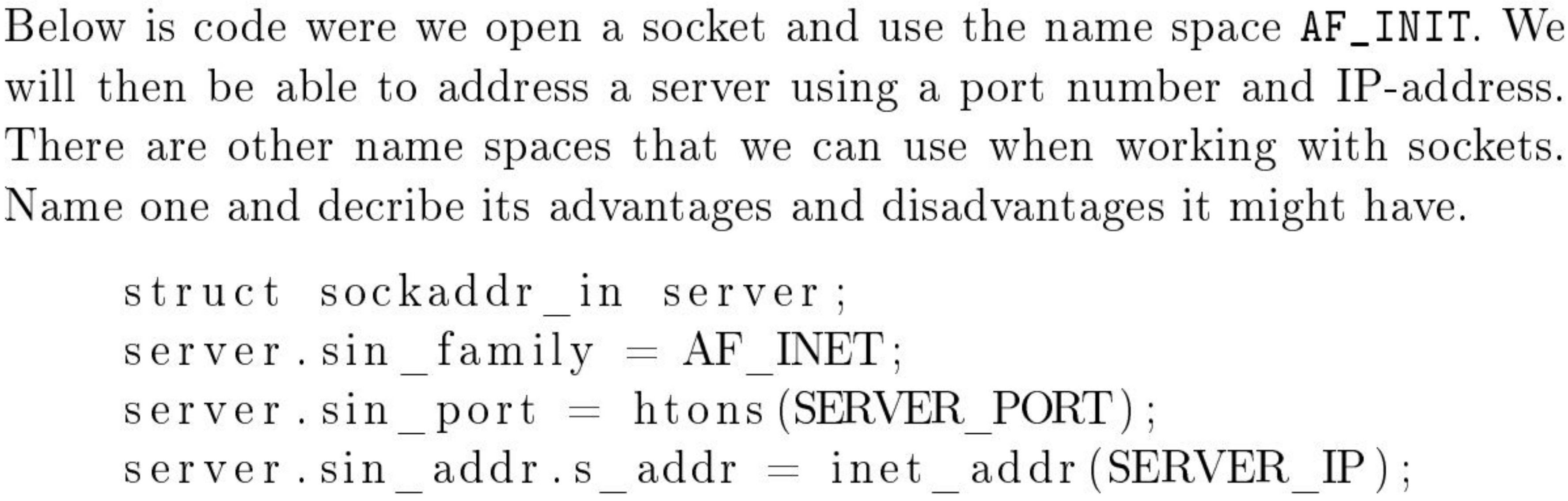
A so called ***pipe*** is a simple way to send data from one process to another. It does have its limitations and a better way is to use so called ***sockets***. If we use a stream socket between two processes we will have several advantages. Describe *two* advantages that a steam socket gives us that we will not have if we use a pipe.

Answer:

When using a stream socket we have a two-way communication, unlike the one-directional pipes.

Pipes are also limited to applications running on the same machine, unlike sockets that can be utilized to pass data over a network.

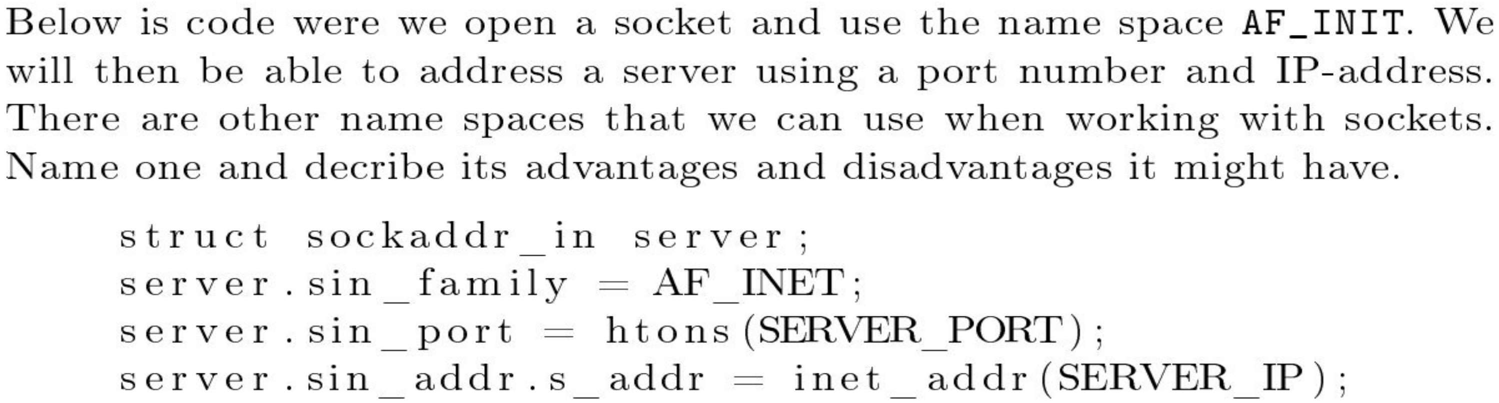
2017-06-07



Below is code were we open a socket and use the name space `AF_INIT`. We will then be able to address a server using a port number and IP-address. There are other name spaces that we can use when working with sockets. Name one and decribe its advantages and disadvantages it might have.

```
struct sockaddr_in server;
server.sin_family = AF_INET;
server.sin_port = htons(SERVER_PORT);
server.sin_addr.s_addr = inet_addr(SERVER_IP);
```

2017-06-07



Below is code were we open a socket and use the name space `AF_INIT`. We will then be able to address a server using a port number and IP-address. There are other name spaces that we can use when working with sockets. Name one and decribe its advantages and disadvantages it might have.

```
struct sockaddr_in server;
server.sin_family = AF_INET;
server.sin_port = htons(SERVER_PORT);
server.sin_addr.s_addr = inet_addr(SERVER_IP);
```

Answer:

For example we have the ***AF_UNIX*** domain socket, that’s constrained to working with passing data between applications on the same machine. It works by binding a specific file on file system that can opened from other applications

This communication is faster, since we can skip the overhead of for example a TCP protocol. But as stated is constrained to only working on communication within the machine.