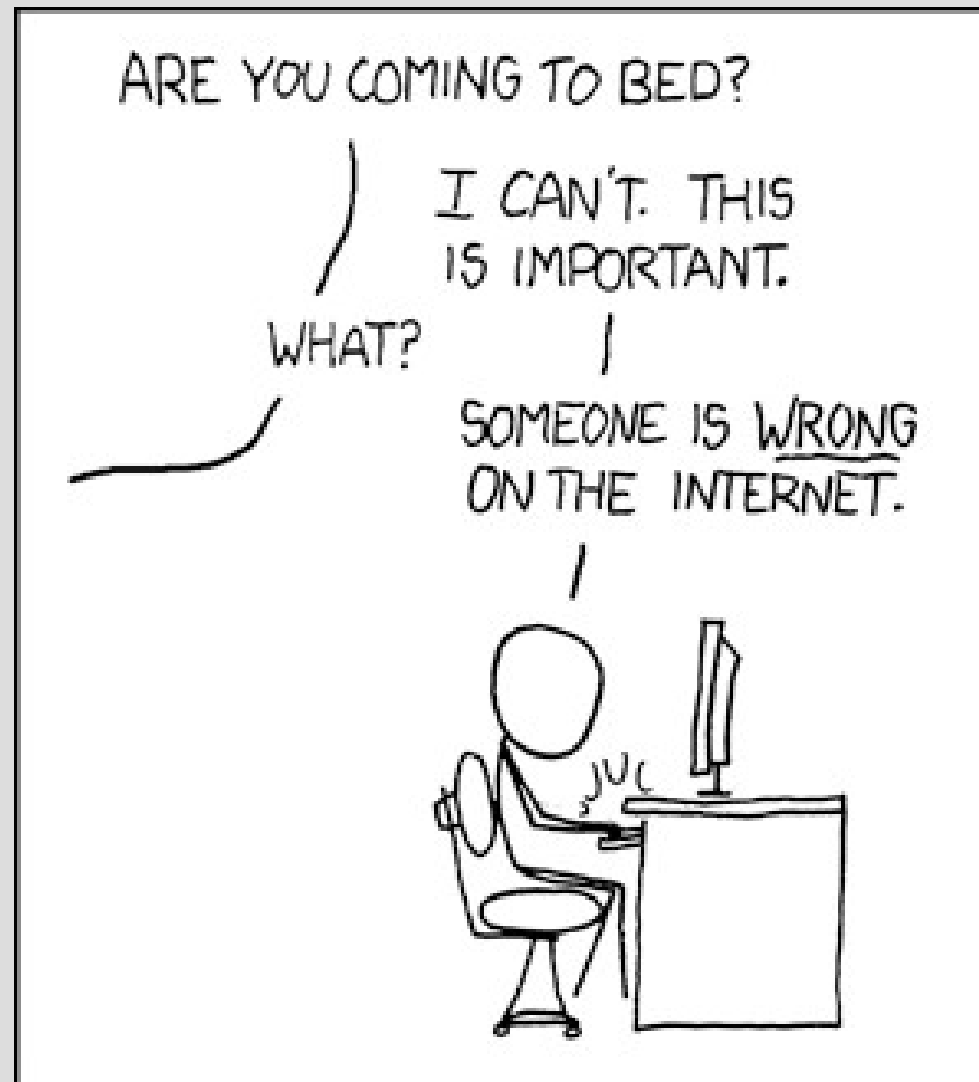


# Over the Internet



# Highlights

- Sockets and packets and ports, oh my!



# Packet

When data travels through the Internet, it passes through many “stations” along the way

To understand where each station will pass it on to, you need an agreed upon layout/format

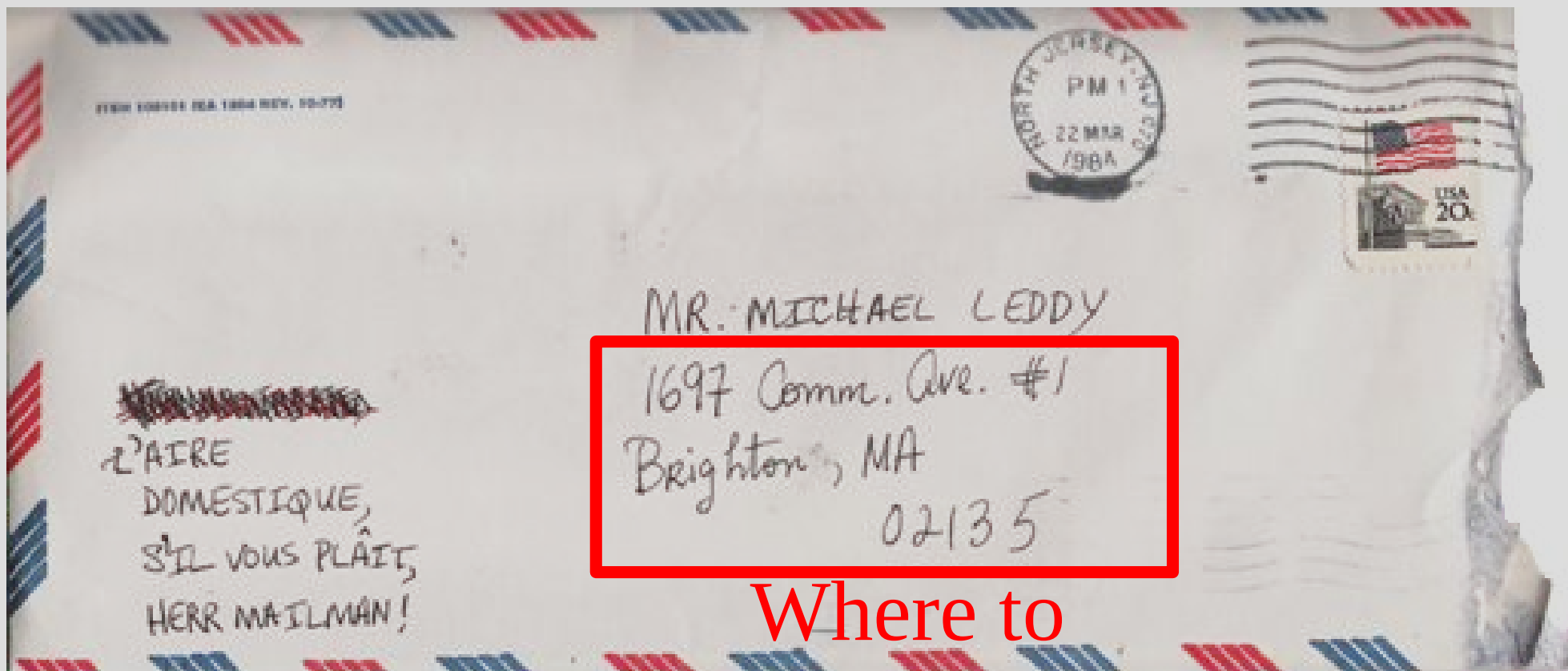
This is very similar to the normal/snail mail



# Packet



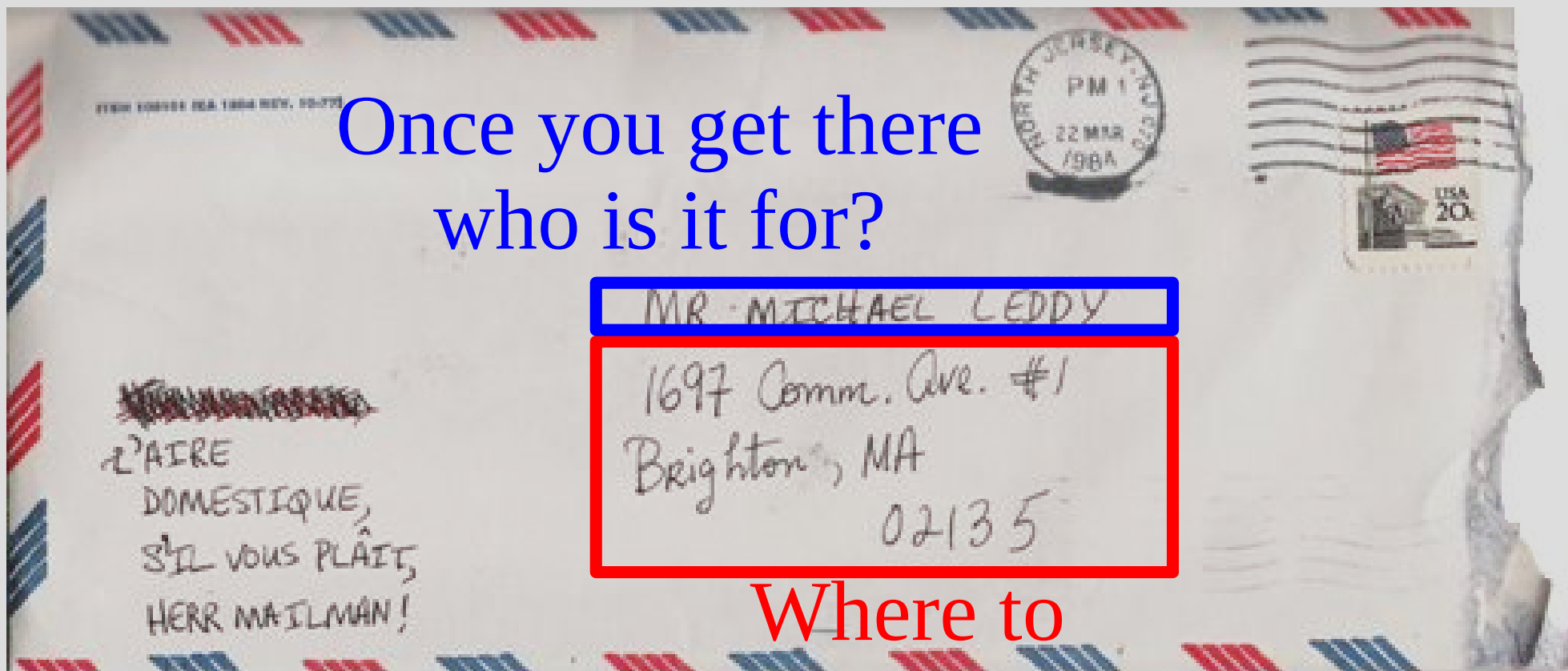
# Packet



Where to  
(standard format with zip code)

# Packet

Once you get there  
who is it for?



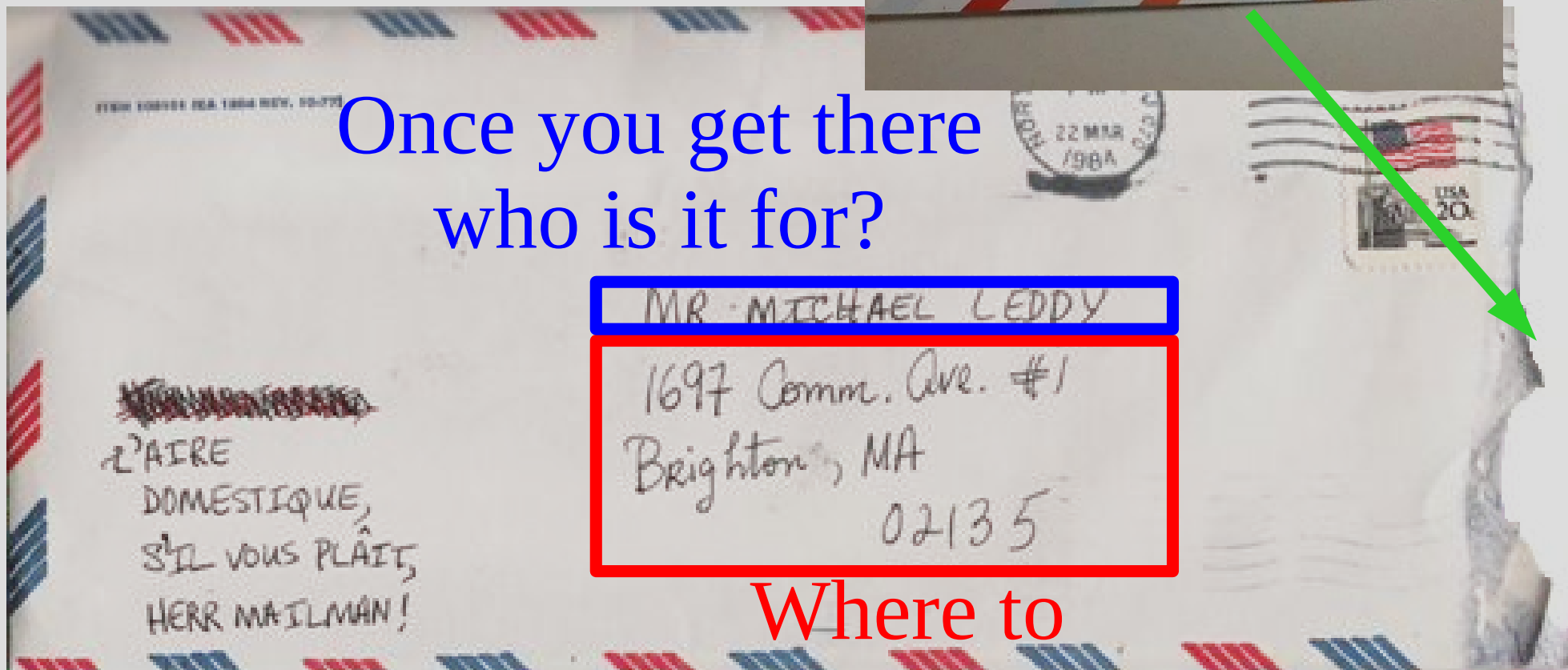
Where to  
(standard format with zip code)

# Package

Message goes inside



Once you get there  
who is it for?



Where to

(standard format with zip code)

# Package

(Payload)

Message goes inside



Once you get there  
who is it for?

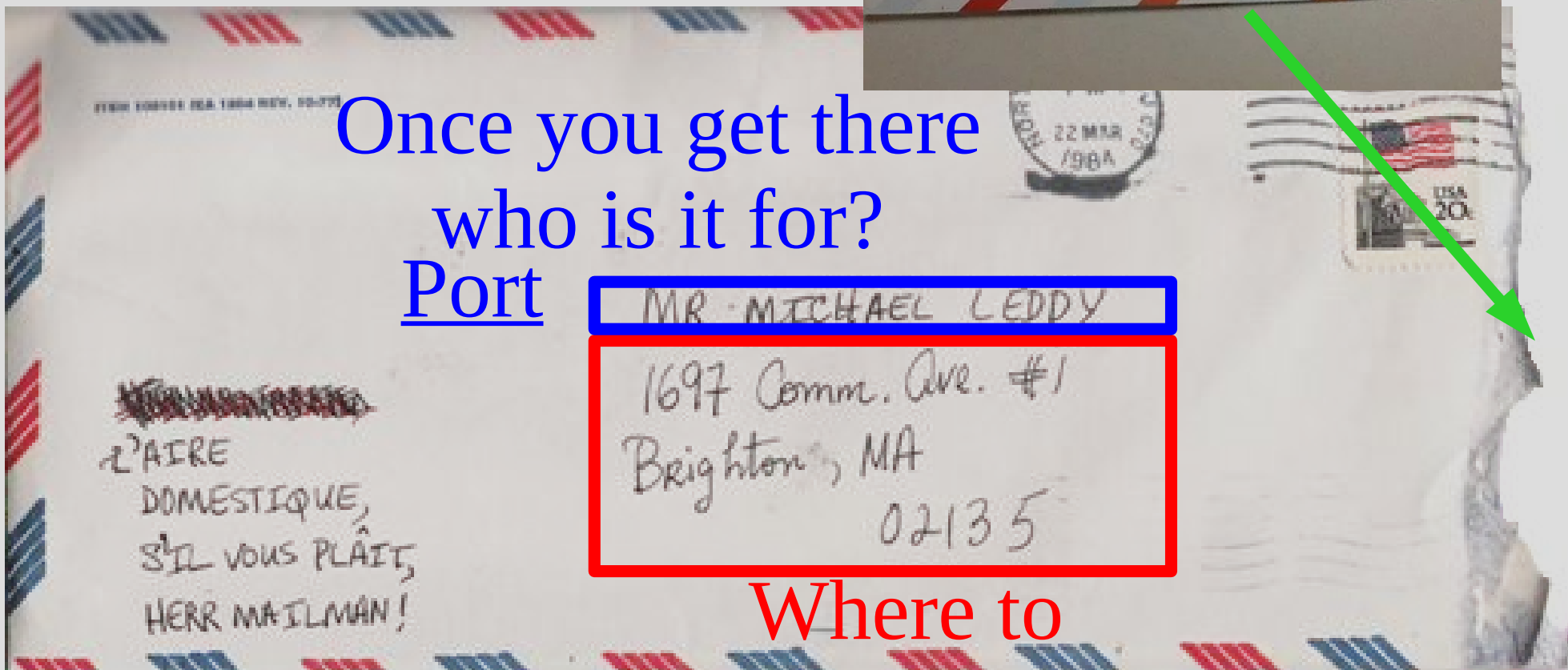
Port

MR. MICHAEL LEDDY

1697 Comm. Ave. #1  
Brighton, MA  
02135

Where to

(standard format with zip code)  
IP address (34.22.123.45)



# Packet

The “IP address” is where on the Internet you want to send the information (what computer)

The “Port” is which app/program on the computer the data is for

Typically both the port and IP address are represented by numbers (or a set of numbers) (No real representation other than the rules we impose, like where “zip codes” are)

# Packet

All these things together:

- Message (Payload)
- IP address (where)
- Port (who)

... is what we call a packet

Packets have a fixed size, so larger messages are broken up over multiple packets

# Sockets

In order to send in C++ over the Internet, you also need a variable for the “connection”

This is very similar to ifstream and ofstream where the variable represents the “file”

These variables that represent the connection are called sockets and you have to set them up much like how you .open() files in C++

# Sockets

Since we are sending things over the Internet, we actually need to make two programs:

- Server = one who receives (sorta)
- Client = one who sends (sorta)

Technically, they both transfer information, but the servers need to be setup first and listen for the clients to send them a message

# Sockets

IP address:

216.239.34.21

Server



Port: 443

IP address:

123.45.67.89

Client



# Sockets

IP address:

216.239.34.21

Server



Port:

443



IP address:

123.45.67.89

Client



# Sockets

IP address:

216.239.34.21

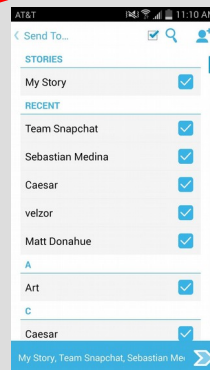
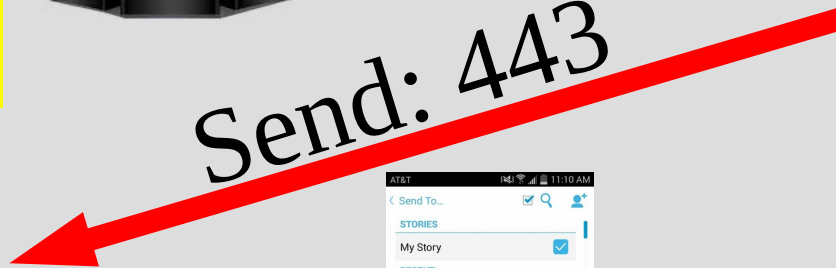
Server



Port:  
443



Send: 443



IP address:

123.45.67.89

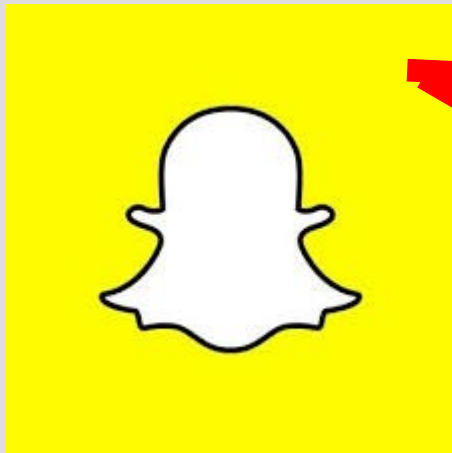
Client



# Sockets

IP address:  
216.239.34.21

Server



IP address:  
123.45.67.89

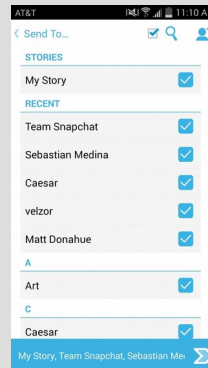
Client



Port:  
443



Send: 443



# Sockets

There are quite a bit of technical details to setting up the variables in C++...

To make the server run, you need to:

- make a “socket” number
- “bind” the socket number to an actual spot
- start “listening” for people to connect  
(i.e. program is ready to take requests)
- “accept” an incoming request
- send data back and forth (“read”&”write”)

# Sockets

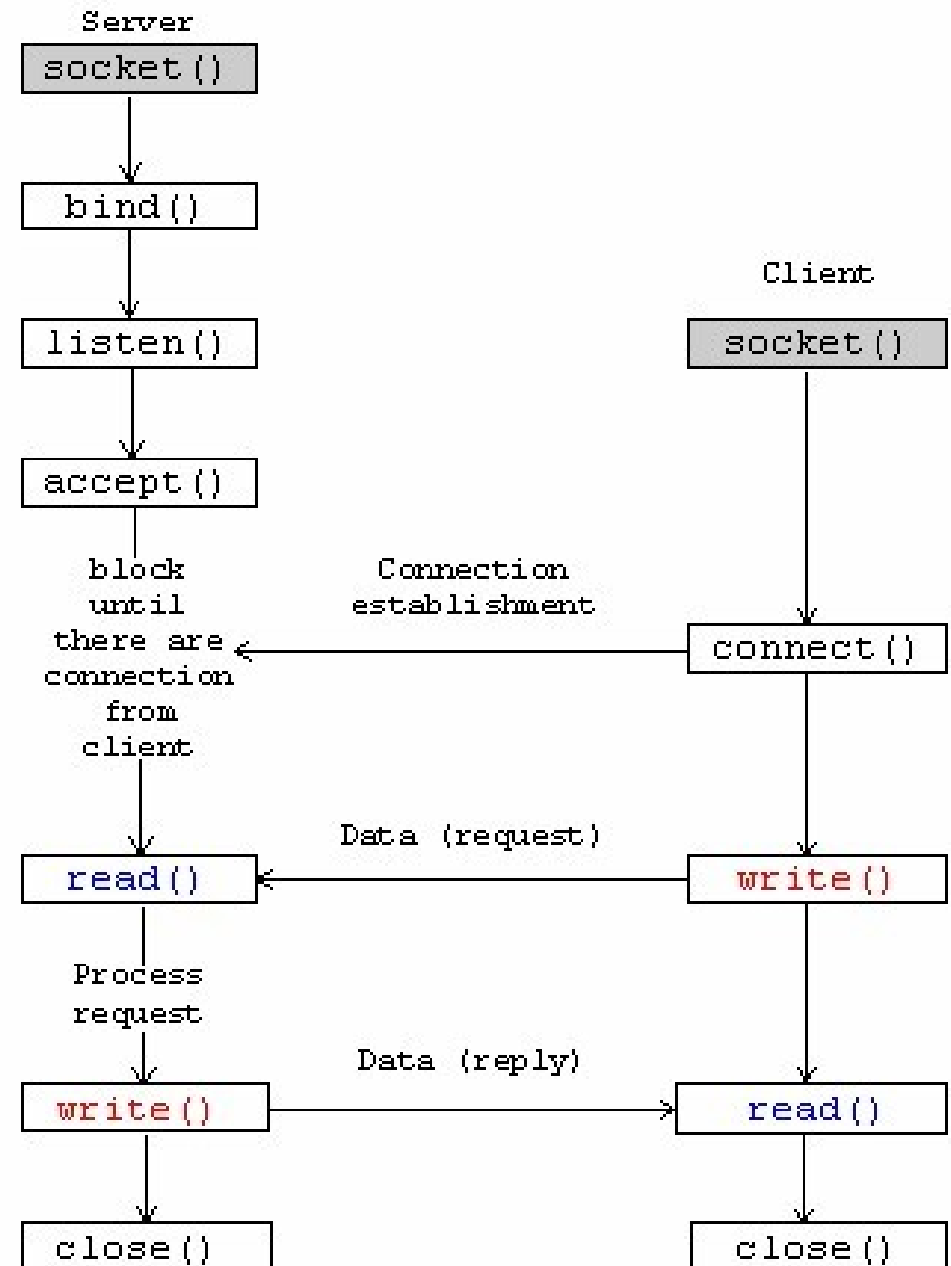
Clients are slightly easier as they don't need to be setup to listen

To make the client run, you need to:

- make a “socket” number
- “bind” the socket number to an actual spot
- try to “connect” to a server
- send data back and forth (“read”&”write”)
- (close connection at end)

# Sockets

(see: server.cpp)  
(see: client.cpp)





WHAT IS YOUR ADDRESS?



173.168.16.11



NO, YOUR LOCAL ADDRESS



127.0.0.1



I MEAN YOUR PHYSICAL ADDRESS !



28:05:FF:58:31:05