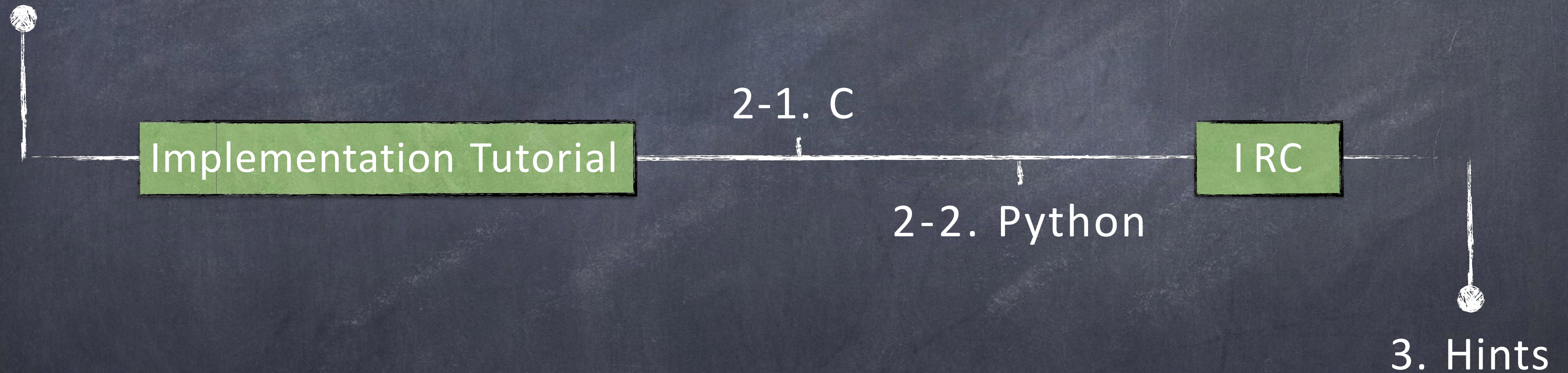


Homework Assignment 1 - Socket Programming & IRC Robot



Outline

1. Introduction



Introduction

- * Socket is the API for the TCP/IP protocol stack.
 - Provides communication between Application layer and Transport layer.
- * Process sends/receives messages to/from socket.

What is Socket Address ?

IP address + Port number

- * IP address : Address the machine
- * Port number : Address the process

Port Number

- * FTP (21) - File Transfer Protocol
- * SSH (22) - Secure Shell
- * telnet (23) - Secure Shell
- * SMTP (25) - Simple Mail Transfer Protocol
- * DNS (53) - Domain Name Server
- * HTTP (80) - Hyper Text Transfer Protocol
- * POP3 (110) - Post Office Protocol

Socket Programming in **C**

[\[Click\] Reference Site 1](#)

Socket Programming in **Python**

[\[Click\] Python Tutorial](#)

[\[Click\] Python Socket](#)


```
# AF_INET : IPv4
# SOCK_STREAM : TCP, SOCK_DGRAM : UDP
Socket = socket.socket( socket.AF_INET, socket.SOCK_STREAM )
```

Server Socket

Method	Description
Socket. bind ()	Binds address (Hostname + Port number) to socket.
Socket. listen ()	Sets up and start TCP listener.
Socket. accept ()	Passively accepts TCP client connection, waiting until arrives.

Client Socket

Method	Description
Socket. connect ()	Actively initiates TCP server connection.

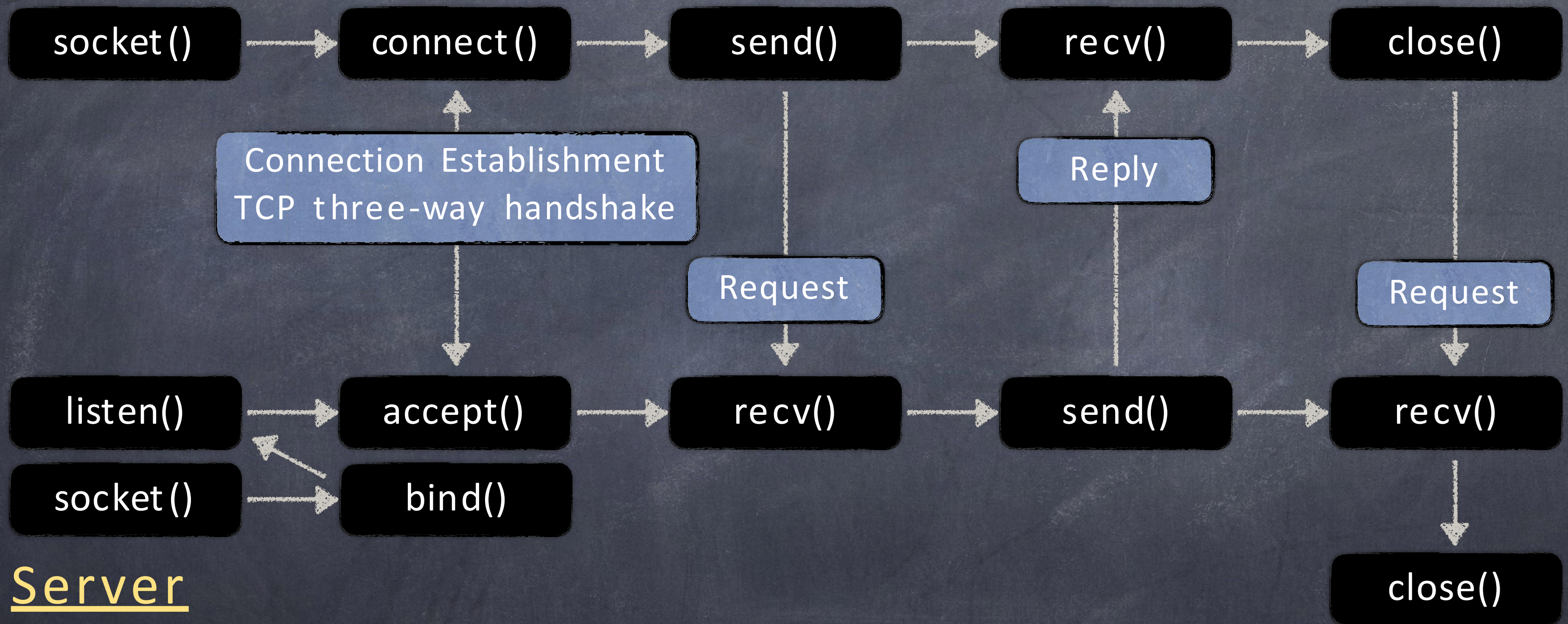

```
# AF_INET : IPv4
# SOCK_STREAM : TCP, SOCK_DGRAM : UDP
Socket = socket.socket( socket.AF_INET, socket.SOCK_STREAM )
```

General Socket

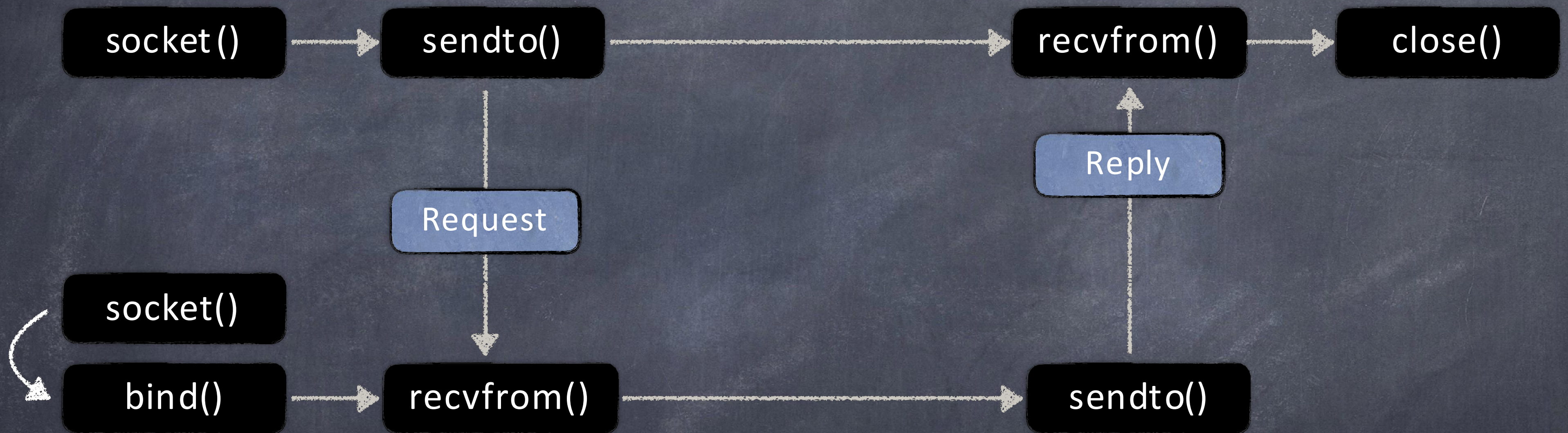
Method	Description
Socket. recv () Socket. send ()	Receives TCP message. Transmits TCP message.
Socket. recvfrom () Socket. sendto ()	Receives UDP message. Transmits UDP message.
Socket. close () Socket. gethostname ()	Close socket. Returns the hostname.

Client

TCP



Server

UDPClientServer

TCP Server

```
import socket # Include library

ServerSocket = socket.socket( socket.AF_INET, socket.SOCK_STREAM )

HostIP = socket.gethostname()
PortNumber = 15566
ServerSocket.bind( ( HostIP, PortNumber ) )
ServerSocket.listen( 5 )

while True :
    Client, Address = ServerSocket.accept()
    Message = Client.recv( 1024 )
    print "Got connection from ", Address
    print "Msg from client : ", Message
    Client.send( "Thank you for connecting" )
    Client.close()
```

Get the name of local machine
Reserve a port for your service
Bind to the port
Wait for client connection
Establish connection with client
Get message from client
Send message back to client
Close the connection

TCP Client

```
import socket # Include library

ClientSocket = socket.socket( socket.AF_INET, socket.SOCK_STREAM )

HostIP = socket.gethostname() # Get the name of local machine
PortNumber = 15566 # Reserve a port for your service
ClientSocket.connect( ( HostIP, PortNumber ) ) # Connect to server

ClientSocket.send( 'Message from client' ) # Send message to server
print ClientSocket.recv( 1024 ) # Output received message

ClientSocket.close() # Close the connection
```


Internet Relay Chat (IRC)

★ IRC is an application layer protocol that facilitates communication in the form of text. The chat process works on a **client/server** networking model.

→ Freenode : An IRC network.

How to use IRC ?

★ Operating System : **Linux, ubuntu 14.04**

→ Step 1. Install **irssi** package

→ Step 2. Simply type irssi command in the shell

→ Step 3. Connect to the Freenode IRC server

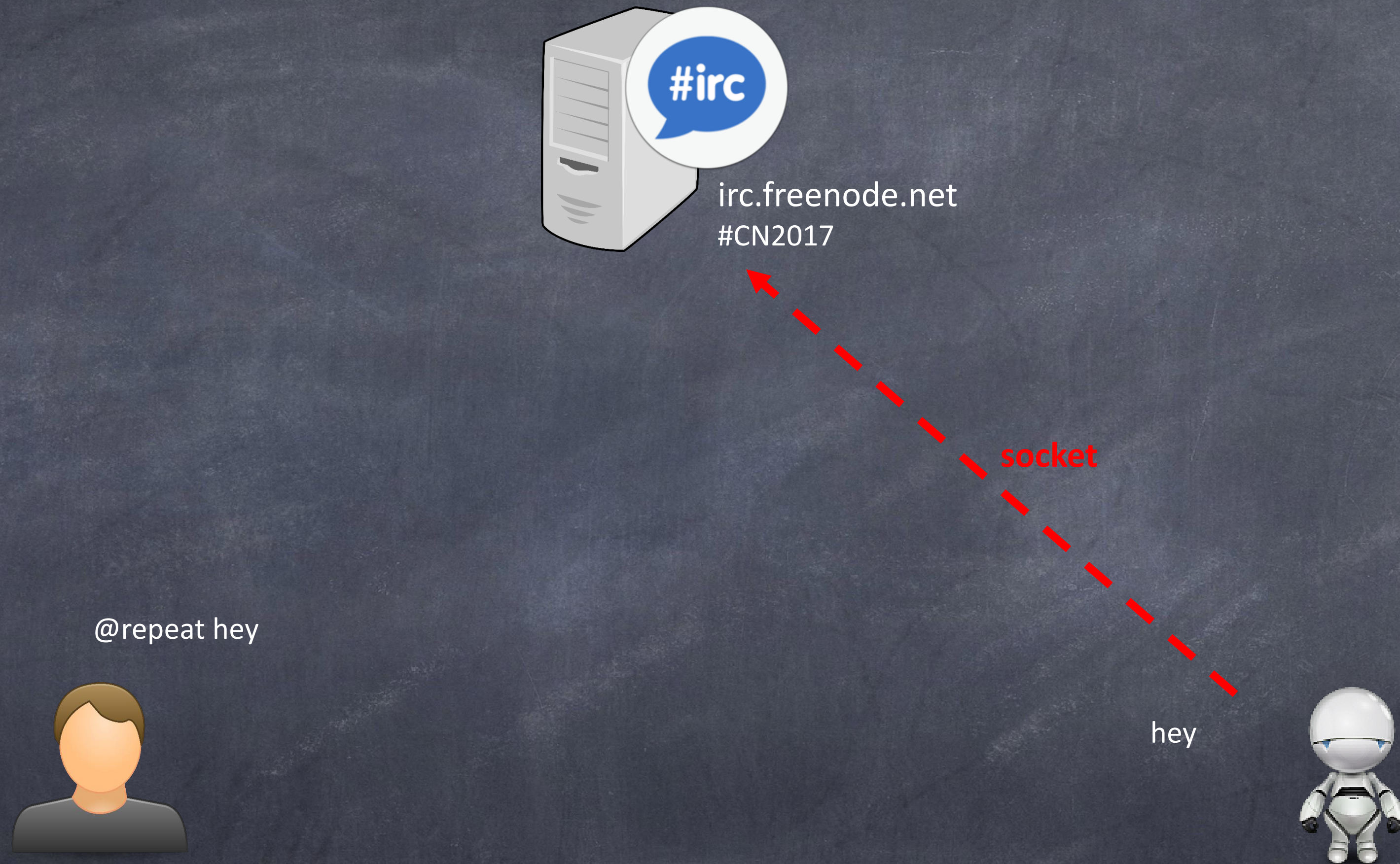
(**irc.freenode.net**) connect irc.freenode.org

→ Step 4. IRC commands ...

Useful Commands

- ★ **CONNECT** <target server> [<port>]
- ★ **INVITE** <nickname> <channel>
- ★ **JOIN** <channel> [<keys>]
- ★ **NAMES** [<channels>]
- ★ **NICK** <nickname>
- ★ **PRIVMSG** <username/channel> <message>
- ★ **USER** <username>

[\[Click\] Reference Site](#)



IRC Client

```
import socket # Include library

IRCsocket = socket.socket( socket.AF_INET, socket.SOCK_STREAM )
IRCsocket.connect( ( irc.freenode.net, 6667 ) )

... .. [????? ^_^ ?????] ... ..

Msg = 'JOIN #CN_DEMO ILoveTA \r\n'
IRCsocket.send( bytes( Msg ) )

... .. [????? ^_^ ?????] ... ..

while True :
    IRCMsg = IRCsocket.recv( 4096 )
    print IRCMsg

    ... .. [????? ^_^ ?????] ... ..
```

20 point
GET ?

Hints

★ IRC will check your robot is “Alive or Not”

→ PING, PONG

使用者收到 PING 時要回覆 PONG，證明在線上

Grading Policy

Language: c/c++/python

(a) Implementation (80%)

(b) Report (10%)

(c) Demo (10%)

Due Date : 23:59, October 18, 2017

Penalty for late submission is “**20%** per day”. **NO**
accept after 23:59, October 20, 2017.

How to Submit

- (a) Please compress all of your file into an archive.
(Format: rar/zip) EX: hw1_rxxxxxxxxx.rar
- (b) Email to ntu.cnta@gmail.com before due date.
Email subject: [CN2017] Homework1_rxxxxxxxxx

Demo Time(Implementation)

- (a) Connect to Channel & Introduction Message (30%)
- (b) 'Repeat' Message (10%)
- (c) Hexadecimal & Decimal Converter (15%)
- (d) Valid IP Address Calculator (20%)
- (e) Help (5%)

