Overview of Programming and Experimental Environment

Hui Chen [a]

[a]CUNY Brooklyn College

September 4, 2024

Outline

1. Developing Network Applications

2. Introduction to Socket API

3. Experimental Environment

4. Example Experiment

Outline

1. Developing Network Applications
2. Introduction to Socket API
3. Experimental Environment
4. Example Experiment

Nearly all computer systems implement their network protocols as part of the operating system.

The interface that the OS provides to its networking subsystem is often called the network application programming interface (API).

Outline

1 Developing Network Applications

2 Introduction to Socket API

3 Experimental Environment

4 Example Experiment

Socket API

SOCKET API is a network API is supported in virtually all popular operating systems. The API has bindings in multiple languages, e.g.,

- Java socket library.
  However, its support of low level programming (data link and network layers) are limited.
- C socket library.
  Typically, representing what the Operating Systems offers
- Python socket library.
  Consiered, a wrapper around C socket library.

Socket API

The main abstraction of the socket interface, not surprisingly, is the socket.

- can be understood as the point where a local application process attaches to the network.

Socket API defines a list of operations

- for creating a socket,
- attaching the socket to the network,
- sending/receiving messages through the socket, and
- closing the socket (in order to release resources).

Socket API: C Binding

```c
int socket(int domain, int type, int protocol);

int bind(int socket, struct sockaddr *address, int addr_len);
int listen(int socket, int backlog);
int accept(int socket, struct sockaddr *address, int *addr_len);
int connect(int socket, struct sockaddr *address, int addr_len);

int send(int socket, char *message, int msg_len, int flags);
int recv(int socket, char *buffer, int buf_len, int flags);

int close();

```

Socket API: Python Binding

Python defines the Socket class in the socket package.

https://docs.python.org/3/library/socket.html

Socket API: Java Binding

Java also defines a Socket class.

https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/net/Socket.html

Example Programs

Example programs:

https://github.com/huichen-cs/NetworkClasses/tree/master/socket

We shall primarily use Python in this class.

Outline

1. Developing Network Applications

2. Introduction to Socket API

3. Experimental Environment

4. Example Experiment

Virtual Machine

Use the instructor's Oracle VM VirtualBox virtual machines image

- Debian Linux 10
- Small footprint

About Virtual Machines

The instructor uses Oracle VM VirtualBox, free and open source software.

https://www.virtualbox.org/wiki/Downloads

If you are using a Mac with ARM CPU (e.g., M1 or M2), check out the Test Builds instead:
https://www.virtualbox.org/wiki/Testbuilds

Outline

1. Developing Network Applications

2. Introduction to Socket API

3. Experimental Environment

4. Example Experiment

Examining a Web Application in Python

We shall primarily use Python. The application consits of a server and a client.

- https://flask.palletsprojects.com/. Use it to create the server
- https://docs.python.org/3/library/socket.html. Use it cto create the client.
- https://scapy.net/. Use it to inspect network traffic.

A Flask “Hello, World“ Program

- https://github.com/pallets/flask

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello, World!"
```

Listing: hello.py

Running Flask “Hello, World“ Program

```
#!/bin/bash

env FLASK_APP=hello.py flask run
```

Listing: runhello.sh

```
@echo off
set FLASK_APP=hello.py
flask run
```

Listing: runhello.cmd

What just happened?

A Web Client Program

```
from http import client as httpclient

def main():
  conn = httpclient.HTTPConnection('127.0.0.1:5000')

  conn.request('GET', '/')
  response = conn.getresponse()
  print(response.status, response.reason)
  received = response.read()
  print(received)

  conn.close()

if __name__ == "__main__":
  # execute only if run as a script
  main()
```

Listing: helloclient.py

Running the Web Client Program

```
@echo off
python hellclient.py
```

Listing: runclient.cmd

```
#!/bin/bash

python helloclient.py
```

Listing: runclient.sh

Summary

- Creating network applications: using Socket API
- Experiment environment: using virtual machines