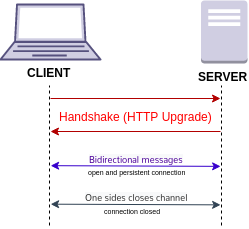
Lab   
WebSocket

รหัส 67040249114 ชื่อ-สกุล นายศุภวิชญ์ หาญมนตรี

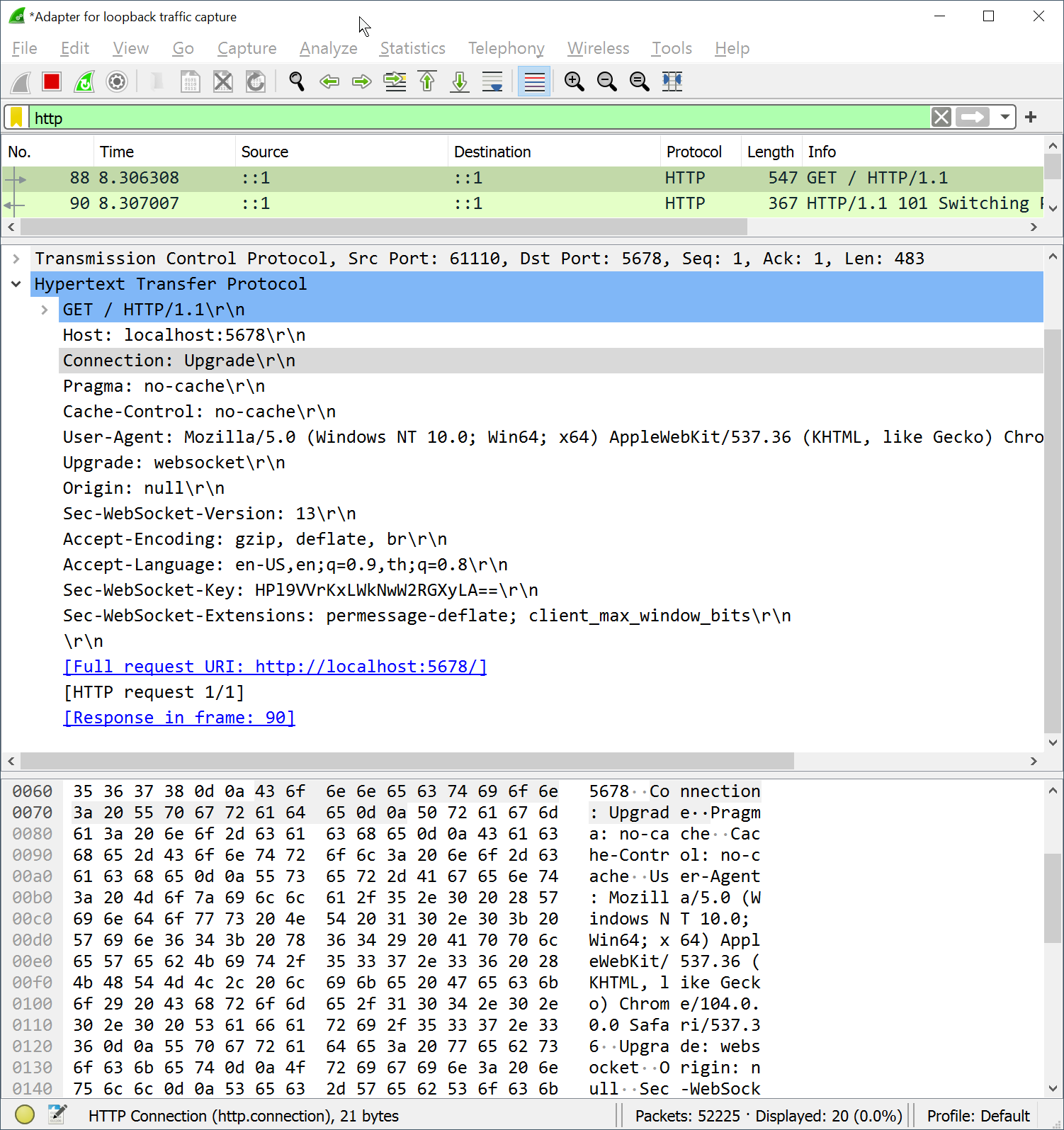
**WebSocket**

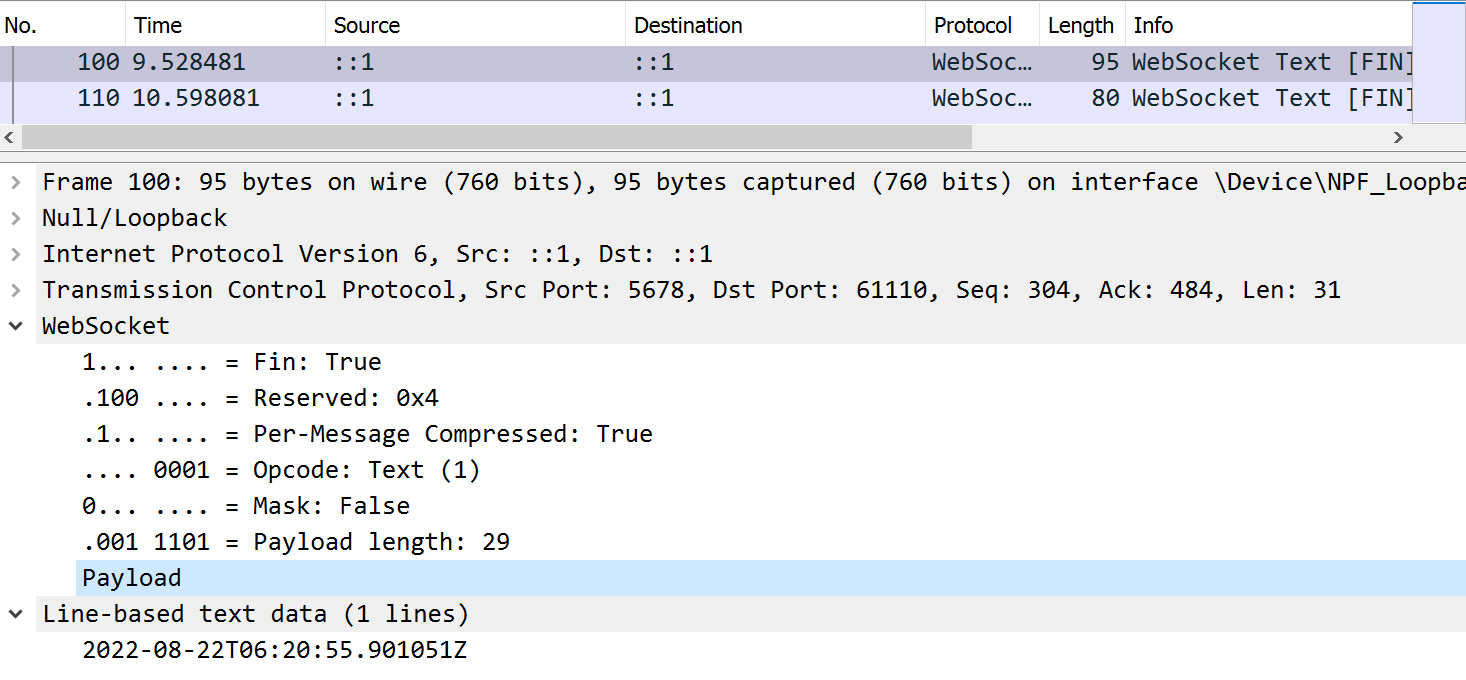


Source: <https://en.wikipedia.org/wiki/WebSocket>

WebSocket เป็น Protocol ทำงานอยู่บน Socket ที่เป็น Connection แบบ TCP (Transmission Control Protocol) คล้ายกับ HTTP แต่รองรับการสื่อสารแบบสองทิศทาง ผู้รับและผู้ส่งสือสารได้ในเวลาเดียวกันดังรูป (Full Duplex หรือ Bidirectional Communication)   
 WebSocket เป็นมาตรฐานที่ถูกกำหนดโดย IETF (Internet Engineering Task Force) RFC6455 (The WebSocket Protocol) สร้างขึ้นในปี 2011 ในปัจจุบันนิยมนำมาใช้งานค่อนข้างเยอะ เพราะเหมาะกับระบบที่ต้องการการอัพเดทข้อมูลแบบ Realtime เช่น ระบบ Chat, ระบบ Notification, Game, และอื่น ๆ

ในการเชื่อมต่อนนั้น การเชื่อมต่อไปยัง WebSocket จะใช้ URI เป็น **ws** และ **wss** คล้าย ๆ กับ http และ https โดยที่ ws เป็นการเชื่อมต่อแบบ Non-secure แต่ wss เป็นการเชื่อมต่อแบบ Secure คือ (WebSocket ที่ทำงานอยู่บน Transport Layer Security) การเชื่อมต่อเช่น ws://localhost/chat หรือ wss://csit.udru.ac.th.com/chat เมื่อเริ่มเชื่อมต่อ WebSocket จะส่ง Request แรก เป็น HTTP ไปยัง WebSocket Server โดย HTTP ใช้ในการเริ่มต้น หรือ Handshake เพื่อ Upgrade จาก HTTP ไปเป็น WebSocket จากนั้นจึงเริ่มกระบวนการสื่อสารด้วย WebSocket จนยกเลิกการเชื่อมต่อ (Close Connection) ดังรูป จะเห็นว่า Wireshark ดักจับได้เป็น HTTP มี header Connection: Upgrade และรายละเอียดของ WebSocket





**Python WebSocket 1**

ตัวอย่างการเขียน WebSocket เพื่อให้ Server กระจ่ายเวลาทุกๆ 1 วินาทีให้กับ Client ที่เข้ามา

ตัวอย่างโค้ด WebSocket Server (show\_time.py)

#!/usr/bin/env *python*

import asyncio

import datetime

import random

import websockets

CONNECTIONS = *set*()

*async* *def* register(websocket):

CONNECTIONS.add(websocket)

try:

await websocket.wait\_closed()

finally:

CONNECTIONS.remove(websocket)

*async* *def* show\_time():

while *True*:

message = datetime.datetime.utcnow().isoformat() + "Z"

websockets.broadcast(CONNECTIONS, message)

await asyncio.sleep(random.random() \* 2 + 1)

*async* *def* main():

async with websockets.serve(register, "localhost", 5678):

await show\_time()

if \_\_name\_\_ == "\_\_main\_\_":

asyncio.run(main())

ตัวอย่างโค้ด html แสดงผล (show\_time.html)

<!*DOCTYPE* *html*>

<html lang="en">

<head>

<title>WebSocket demo</title>

</head>

<body>

<script src="show\_time.js"></script>

</body>

</html>

ตัวอย่างโค้ด javascript เพื่อ เรียกใช้ WebSocket client มาแสดงผลใน html (show\_time.js)

*window*.addEventListener("DOMContentLoaded", () *=>* {

*const* messages = *document*.createElement("ul");

*document*.body.appendChild(messages);

*const* websocket = new WebSocket("ws://localhost:5678/");

websocket.onmessage = ({ data }) *=>* {

*const* message = *document*.createElement("li");

*const* content = *document*.createTextNode(data);

message.appendChild(content);

messages.appendChild(message);

};

});

**Python WebSocket 2**

ตัวอย่างการเขียน WebSocket เพื่อให้ Server สามารถรับคำสั่งจาก Client มาประมวลผลเพื่ออัปเดตสถานะแอปพลิเคชัน และเผยแพร่สถานะที่อัปเดตไปยัง Client ที่เชื่อมต่อทั้งหมด

ตัวอย่างโค้ด WebSocket Server (counter.py)

#!/usr/bin/env *python*

import asyncio

import json

import logging

import websockets

logging.basicConfig()

USERS = *set*()

VALUE = 0

*def* users\_event():

return json.dumps({"type": "users", "count": *len*(USERS)})

*def* value\_event():

return json.dumps({"type": "value", "value": VALUE})

*async* *def* counter(websocket):

global USERS, VALUE

try:

# Register user

USERS.add(websocket)

websockets.broadcast(USERS, users\_event())

# Send current state to user

await websocket.send(value\_event())

# Manage state changes

async for message in websocket:

event = json.loads(message)

if event["action"] == "minus":

VALUE -= 1

websockets.broadcast(USERS, value\_event())

elif event["action"] == "plus":

VALUE += 1

websockets.broadcast(USERS, value\_event())

else:

logging.error("unsupported event: %s", event)

finally:

# Unregister user

USERS.remove(websocket)

websockets.broadcast(USERS, users\_event())

*async* *def* main():

async with websockets.serve(counter, "localhost", 6789):

await asyncio.Future() # run forever

if \_\_name\_\_ == "\_\_main\_\_":

asyncio.run(main())

ตัวอย่างโค้ด html แสดงผล (counter.html)

<!*DOCTYPE* *html*>

<html lang="en">

<head>

<title>WebSocket demo</title>

<link href="counter.css" rel="stylesheet">

</head>

<body>

<div class="buttons">

<div class="minus button">-</div>

<div class="value">?</div>

<div class="plus button">+</div>

</div>

<div class="state">

<span class="users">?</span> online

</div>

<script src="counter.js"></script>

</body>

</html>

ตัวอย่างโค้ด css (counter.css)

body {

*font-family*: "Courier New", *sans-serif*;

*text-align*: *center*;

}

.buttons {

*font-size*: 4*em*;

*display*: *flex*;

*justify-content*: *center*;

}

.button, .value {

*line-height*: 1;

*padding*: 2*rem*;

*margin*: 2*rem*;

*border*: *medium* *solid*;

*min-height*: 1*em*;

*min-width*: 1*em*;

}

.button {

*cursor*: *pointer*;

*user-select*: *none*;

}

.minus {

*color*: *red*;

}

.plus {

*color*: *green*;

}

.value {

*min-width*: 2*em*;

}

.state {

*font-size*: 2*em*;

}

ตัวอย่างโค้ด javascript เพื่อ เรียกใช้ WebSocket client มาแสดงผลใน html (counter.js)

*window*.addEventListener("DOMContentLoaded", () *=>* {

*const* websocket = new WebSocket("ws://localhost:6789/");

*document*.querySelector(".minus").addEventListener("click", () *=>* {

websocket.send(*JSON*.*stringify*({ action: "minus" }));

});

*document*.querySelector(".plus").addEventListener("click", () *=>* {

websocket.send(*JSON*.*stringify*({ action: "plus" }));

});

websocket.onmessage = ({ data }) *=>* {

*const* event = *JSON*.*parse*(data);

switch (event.type) {

case "value":

*document*.querySelector(".value").textContent = event.value;

break;

case "users":

*const* users = `${event.count} user${event.count == 1 ? "" : "s"}`;

*document*.querySelector(".users").textContent = users;

break;

default:

*console*.*error*("unsupported event", event);

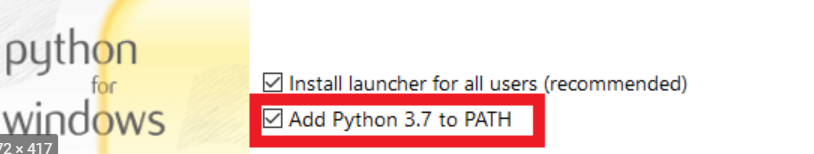
}

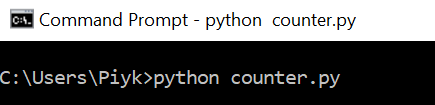
};

});

**การรัน**

1. กรณียังไม่มี python ให้ติดตั้ง <https://www.python.org/downloads/> เลือก Add to path



1. ติดตั้ง WebSocket โดยพิมพ์ pip install websockets   
   
2. รันโค้ดโดยใช้ cmd ****
3. เปิดไฟล์ counter.html ด้วย chrome
4. **ผลการทำงานตามโค้ดตัวอย่าง**

ผลการรัน show\_time.py

A screenshot of a computer

AI-generated content may be incorrect.

ผลการรัน show\_time.html

A screenshot of a computer

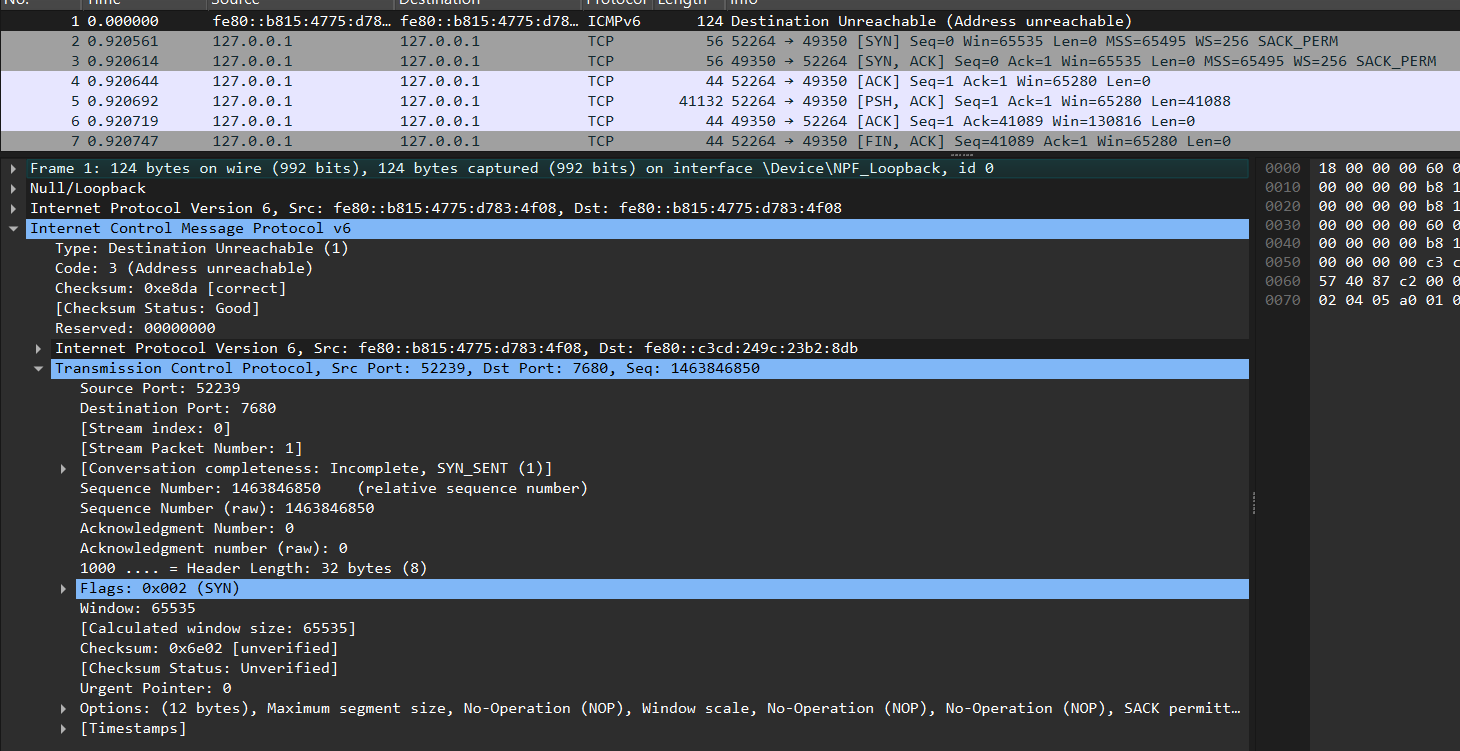
AI-generated content may be incorrect.

ผลการรัน counter.py A screen shot of a computer

AI-generated content may be incorrect.

ผลการรัน counter.html A screenshot of a computer

AI-generated content may be incorrect.



1. **ให้ปรับเปลี่ยนโค้ด หรือเขียนใหม่ โดยใช้ websocket เขียนเป็น app อะไร ก็ได้**

เว็บนับการคลิกแล้วเปลี่ยนรูปและเล่นเสียง เมื่อคลิกถึง 1500 รอบรูปจะมีการเปลี่ยนแปลง

