

Socket Programming Assignment: UDP-based Reliable File Transfer

จงเขียน client program และ server program ด้วยภาษา Python เพื่อใช้ส่งไฟล์จากฝั่ง client ไปยังฝั่ง server โดยกำหนดให้ใช้ UDP เป็น Transport Layer Protocol ซึ่งผู้เรียนจำเป็นต้องพัฒนาแนวทางการส่งข้อมูลแบบเชื่อถือได้ (Reliable Delivery) ขึ้นมาเอง

Requirements and Limitations

1. โปรแกรมต้องสามารถนำส่งข้อมูลแบบเชื่อถือได้จาก client process ไปยัง server process เพื่อให้ฝั่ง server ได้ข้อมูลไฟล์ถูกต้องครบถ้วน ถึงแม้ว่าเครือข่ายที่เชื่อมต่อจะมีปัญหา packet duplication หรือ packet loss
2. โปรแกรมต้องสามารถนำส่ง binary file ได้
3. โปรแกรมต้องรองรับการระบุหมายเลข IP address และ port ของ server ผ่าน command-line arguments
4. โปรแกรมต้องสามารถทำงานได้บนระบบปฏิบัติการ Linux
5. โปรแกรมต้องพัฒนาด้วยภาษา Python เวอร์ชัน 3.8+ หรือเวอร์ชันสูงกว่าเท่านั้น
6. โปรแกรมต้องสามารถทำงานได้จาก source files ที่ผู้เรียนพัฒนา โดยไม่จำเป็นต้อง download หรือติดตั้ง package หรือ software อื่นใดเพิ่มเติมนอกเหนือจาก libraries หรือ modules ที่มีให้อยู่แล้วจากการติดตั้งตัวแปลภาษา Python มาตรฐานบนระบบปฏิบัติการ Linux
7. ในการทำงานของโปรแกรมจะต้องใช้เพียง User Datagram Protocol เป็น Transport Layer Protocol เท่านั้น
8. ในการทำงานของโปรแกรมฝั่ง server เพื่อรับไฟล์ในแต่ละครั้ง จะต้องใช้ socket เพียง socket เดียวเท่านั้น
9. จำนวน source files ทั้งหมดของโปรแกรมนรวมทั้งฝั่ง client และ ฝั่ง server แล้ว ต้องมีจำนวนไม่เกิน 5 ไฟล์
10. จำนวนบรรทัดของ code จาก source files ที่ผู้เรียนพัฒนารวมกันทุกไฟล์ ต้องมีจำนวนไม่เกิน 2000 บรรทัด

Program Usage

1. ไฟล์สำหรับเริ่มการทำงานของโปรแกรมฝั่ง server ให้ตั้งชื่อว่า “urft_server.py” กำหนดให้ใช้คำสั่งต่อไปนี้ในการเริ่มการทำงานของโปรแกรมฝั่ง server โดยหลังจากที่ทำงานแล้ว จะต้องไม่มีการรับ keyboard input ใดๆ เพิ่มอีก

```
python urft_server.py <server_ip> <server_port>
```

- <server_ip> คือ หมายเลข IP address ที่ socket ของ server process ใช้งาน
- <server_port> คือ หมายเลข port ที่ socket ของ server process ใช้งาน

2. ไฟล์สำหรับเริ่มการทำงานของโปรแกรมฝั่ง client ให้ตั้งชื่อว่า “urft_client.py” กำหนดให้ใช้คำสั่งต่อไปนี้ในการเริ่มการทำงานของโปรแกรมฝั่ง client โดยหลังจากที่ทำงานแล้ว จะต้องไม่มีการรับ keyboard input ใดๆ เพิ่มอีก

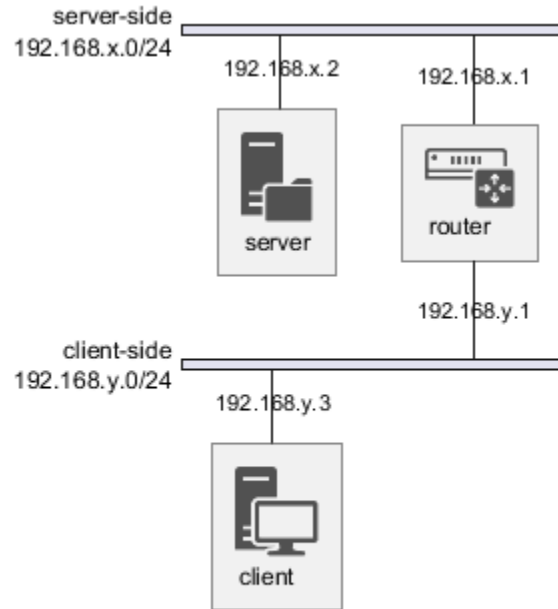
```
python urft_client.py <file_path> <server_ip> <server_port>
```

- <file_path> คือ path ของไฟล์ที่ client process ต้องการส่งให้ server process
- <server_ip> คือ หมายเลข IP address ที่ socket ของ server process ใช้งาน
- <server_port> คือ หมายเลข port ที่ socket ของ server process ใช้งาน

3. หลังจากเริ่มทำงาน ฝั่ง server จะเป็นฝ่ายรอฝั่ง client ติดต่อและส่งข้อมูลไฟล์มาให้ โดยฝั่ง client จะต้องแจ้งชื่อไฟล์ให้ฝั่ง server ทราบก่อนส่งข้อมูลในไฟล์ เพื่อให้ฝั่ง server บันทึกข้อมูลที่รับลงในไฟล์ตามชื่อที่ client แจ้ง
4. กำหนดให้การทำงานของโปรแกรมฝั่ง server ในแต่ละครั้ง รับการติดต่อจาก client เพียงรายเดียว
5. หลังจากโปรแกรมฝั่ง client ส่งไฟล์ตามระบุสำเร็จ จะต้องจบการทำงานและกลับมาที่ shell prompt โดยไม่มี error ใดๆ ในทำนองเดียวกัน หลังจากโปรแกรมฝั่ง server รับข้อมูลและเขียนไฟล์เสร็จสิ้น จะต้องจบการทำงานและกลับมาที่ shell prompt โดยไม่มี error ใดๆ

Program Evaluation and Marking

การทดสอบการทำงานจะใช้วิธีการนำโปรแกรมฝั่ง **client** และ **server** ไปทำงานบน host ซึ่งอยู่กันคนละ **subnet** โดยมี **router** ทำหน้าที่เชื่อมต่อและส่งต่อ **packet** ระหว่าง **subnets** นอกจากนี้ **router** จะยังทำหน้าที่เลียนแบบสภาพเครือข่ายด้านต่างๆ ทั้ง **network latency**, **packet loss**, **packet duplication** และ **packet re-ordering** โดยมีการเชื่อมต่อดังภาพ



การเลียนแบบสภาพเครือข่ายด้านรวมถึงเงื่อนไขในการทดสอบอื่นๆ และคะแนน เป็นไปตามตารางต่อไปนี้ โดยผู้เรียนจะมีสิทธิ์ได้คะแนนจากการทดสอบหมายเลขใดนั้น โปรแกรมของผู้เรียนจำเป็นต้องผ่านการทดสอบหมายเลขต่ำกว่าทั้งหมดแล้วเท่านั้น เช่น ผู้เรียนมีสิทธิ์จะได้คะแนนจากการทดสอบที่ 4 ก็ต่อเมื่อผ่านการทดสอบที่ 1, 2 และ 3 แล้วเท่านั้น

No.	File Size	Transport Protocol	RTT (ms)	Client-to-Server Link Condition	Server-to-Client Link Condition	Time Limit (seconds)	Points
1	1 MiB	UDP	10	-	-	30	4
2	1 MiB	UDP	10	Packet Duplication: 2%	-	30	3
3	1 MiB	UDP	10	Packet Loss: 2%	-	30	3
4	1 MiB	UDP	10	-	Packet Duplication: 5%	30	3
5	1 MiB	UDP	10	-	Packet Loss: 5%	30	3
6	1 MiB	UDP	250	-	-	60	2
7	1 MiB	UDP	250	Packet Re-ordering: 2%	-	90	2

Submission

- ให้ส่ง **source files** ทั้งหมดใน Assignment ที่ผู้สอนประกาศไว้บน **Microsoft Teams** ของรายวิชาภายในเวลาที่กำหนดไว้ โดยให้ส่ง **source files** ภาษา **Python** ทั้งหมดใน Assignment โดยตรงแบบไม่ต้องมีการบีบอัดใดๆ
- ให้ตั้งชื่อไฟล์โปรแกรมฝั่ง **client** และฝั่ง **server** ตามชื่อที่กำหนดไว้ในวิธีใช้งาน ส่วนไฟล์อื่นๆ ที่ผู้เรียนมีการใช้งานเพิ่มเติมสามารถตั้งชื่ออย่างไรก็ได้ แต่ต้องสามารถอ้างอิงหรือใช้งานได้ในระบบปฏิบัติการ **Linux**
- การตรวจจะเป็นการใช้ **script** เพื่อทำการตรวจโดยระบบอัตโนมัติตาม **test cases** ที่ได้รับไว้ โดยจะตรวจสอบจากขนาดและค่า **md5sum** ของไฟล์ที่ส่งสำเร็จว่ามีค่าตรงกับค่าของไฟล์ต้นฉบับหรือไม่
- เนื่องจากเป็นการใช้ระบบอัตโนมัติในการตรวจ จึงขอให้ผู้เรียนปฏิบัติตามคำอธิบายการส่งงานอย่างเคร่งครัด ข้อผิดพลาดใดๆ ที่เกิดจากการที่ผู้เรียนไม่ปฏิบัติตามข้อกำหนดการส่ง และส่งผลให้ระบบอัตโนมัติไม่สามารถตรวจงานได้ จะถือว่าส่งผลให้คะแนนส่วนนั้นของผู้เรียน มีคะแนนเป็นศูนย์
- ผู้เรียนสามารถนำโปรแกรมที่พัฒนามาแล้วมาทดสอบได้ในช่วงคาบรายวิชาปฏิบัติ โดยผู้สอนจะทดลองนำโปรแกรมที่ผู้เรียนพัฒนาไปทดลองทำงานบนระบบ **virtual machines** ที่ผู้สอนเตรียมไว้