CSC 591/791, ECE592/792 Spring 2022

Homework Assignment # 2

Team members:

Anand Kumar Chowdadenahalli Ramesh (achowda)

Ella Wells (ekwells)

Fu-Jen Yen (fyen)

Lee Shyu (lshyu)

Shao-Yo Chao (schao2)

Contributions:

	Anand Kumar Chowdadenahalli Ramesh (achowda)	Ella Wells (ekwells)	Fu-Jen Yen (fyen)	Lee Shyu (lshyu)	Shao-Yo Chao (schao2)
MQTT QoS 1 Publisher			100%		
MQTT QoS 1 Subscriber			100%		
MQTT QoS 2 Publisher				100%	
MQTT QoS 2 Subscriber				100%	
CoAP Server					100%
CoAP Client					100%
HTTP Client Raspberry Pi	100%				
HTTP Server Windows 11 PC		100%			

MQTT QoS 1

	Throughput (in kilo bits per second)	
File Size	Average	Standard Deviation
100 Bytes	6.35	1.29
10 Kilobytes	593.42	131.26
1 Megabytes	19865.39	3985.42
10 Megabytes	27419.76	2871.02

We conducted MQTT QoS1 experiments on a mac and two PCs. As the transfer file size gets bigger, the throughput also gets higher. MQTT QoS1 had the highest throughput among all experiments when transferring big files. MQTT QoS1 had a similar or slightly better performance in transferring small files. Since QoS1 is an "at least once" mechanism, it can perform faster under a good internet environment than QoS2.

MQTT QoS 2

	Throughput (in kilo bits per second)	
File Size	Average	Standard Deviation
100 Bytes	5.11615	1.856675
10 Kilobytes	218.649149	184.009108
1 Megabytes	1072.685002	1913.718765
10 Megabytes	1428.905548	7850.584237

As for transferring through MQTT with QoS 2, the throughput seems to be affected by the header in the packets and the time broker check if the packets have been received. When sending files of 100B, the throughput dropped significantly compared to throughput when sending files of 10kB. When transferring 100B files, the header needs 95% of the size of the original file though the header only needs 0.9277% of the size of the original file transferring 10kB files.

CoAP

In the CoAP experiment, we set a Windows 10 PC as a server and a Linux virtual machine as a client. We recorded the transferring time between client requesting and receiving the data. Moreover, we use Wireshark to analyze packets and compute the total application layer data transferred from sender to receiver per file divided by the file size. We can find out that CoAP has a more balanced throughput than HTTP and MQTT, it also supports observing resources, it is best suited to a state transfer model. On the other hand, CoAP usually has smaller header size than HTTP, and has better performance when the data size is small enough. As a result, CoAP is suited to the IoT system.

	Throughput (in kilo bits per second)	
File Size	Average	Standard Deviation
100 Bytes	65.74788	31.86086
10 Kilobytes	1290.544	518.1117
1 Megabytes	3557.479	304.1437
10 Megabytes	2390.946	554.212

HTTP

We conducted two different experiments to compare transfer times using HTTP. The first setup used a Raspberry Pi 3 as the HTTP client and a Windows 10 PC as the HTTP server. The second setup also used a Windows 10 PC as the HTTP server, with the client being a Windows 11 PC. The transfer times were similar for 100 B and 10 KB files sizes, however, the 1 MB and 10 MB files arrived at the Windows 11 PC client faster than the Raspberry Pi client.

HTTP average throughput for different file sizes is captured in Table 1. Based on the outcome from this experiment, the HTTP is more throughput efficient while transferring larger amounts of data.

Table 1 HTTP throughput comparison for different file sizes

	Throughput (in kilo bits per second)	
File Size	Average	Standard Deviation
100 Bytes	5.644	2.476
10 Kilobytes	413.447	104.789
1 Megabytes	7883.271	2169.797
10 Megabytes	14020.215	1996.28

Furthermore, the throughput is higher for larger amounts of data for a PC to PC transfer compared to a PC to

Raspberry Pi transfer. The Raspberry Pi3 client is constrained by the following compared to a PC

- Driver implementation and buffer limitations on raspberry Pi
- Power efficient hardware on raspberry Pi trading off performance compared to a PC

Table 2 HTTP transfer time measured with Raspberry Pi3 Client

File Size	Avg Time for transfer(seconds)	Standard Deviation(seconds)
100 Bytes	0.018	0.01
10 Kilobytes	0.025	0.015
1 Megabytes	0.3047	0.0524
10 Megabytes	2.286	0.126

Table 3 HTTP transfer time measured with a Windows PC as client

File Size	Avg Time for transfer(seconds)	Standard Deviation(seconds)
100 Bytes	0.018	0.014
10 Kilobytes	0.021	0.019
1 Megabytes	0.133	0.051
10 Megabytes	0.736	0.122

Summary

Transmission of large payload such as 1MB and 10MB, MQTT QoS 1 has the best performance among the four protocol. It has 27419 kilo bits/sec of throughput and approximately 2 times more throughput than the second fast protocol under the same circumstances. However, transmission of small payload such as 100B and 10kB, CoAP perform better than the rest. It has 66 kilo bit/sec of throughput when sending files of 100B. As for MQTT QoS 2, which has the least throughput under every condition.

Setup of HTTP

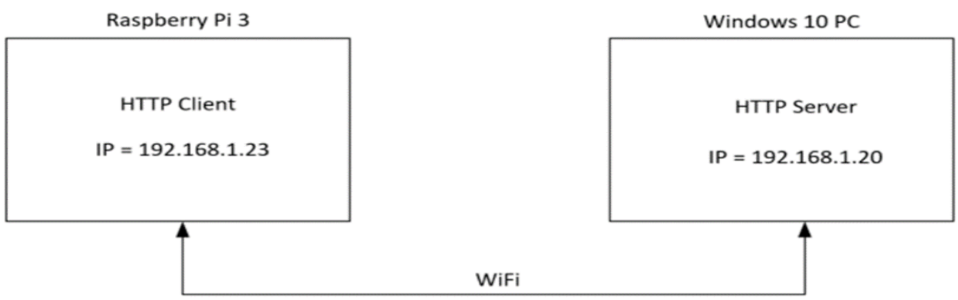

Figure 1 HTTP Experiment Setup Block Diagram

The experiment setup is shown in Figure 1

- A windows 10 http server hosted the files in this experiment
- A Raspberry Pi3 was used as a http client in this experiment
- The HTTP server and the client were assigned static IP addresses 192.168.1.23 and 192.168.1.20 respectively shown in the block diagram
- The server and the client both had Python 3.7.4 installed
- To deploy the server, python http server library was used that comes pre-installed with Python 3.7.4 as shown in Figure 2.

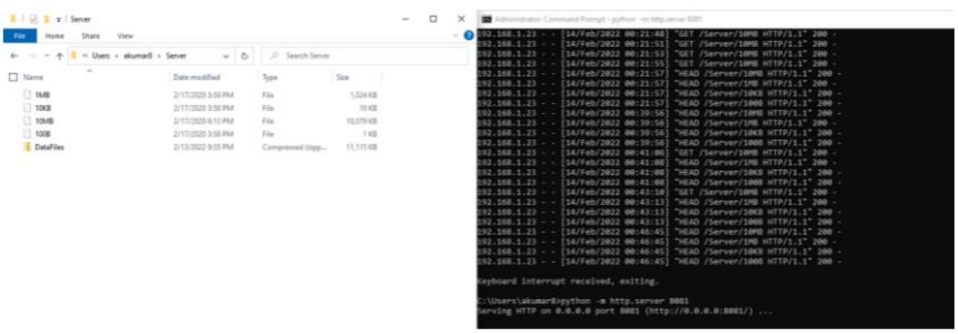

Figure 2 Deployment of HTTP server using Python http library