Performance Analysis of AMQP Protocol for Patient Health Data in IoT Case

Dickri Gumelar[1], Andrian Rakhmatsyah[1] and Rahmat Yasirandi[1]
[1]*School of Computing, Telkom University, Telekomunikasi Street No.1, Bandung, Indonnesia*

Keywords: Classification, Health Data, Publisher, Subscriber, Protocol.

Abstract: Data classification is broadly defined as the process of organizing data based on relevant categories so that it can be used and protected more efficiently. Classification of data divided into several classes, one of which is health data, which needs are essential for data exchange because health data has confidentiality and a large amount for transfer. Electrocardiogram (ECG) requires timeliness in exchanging data. Currently, the MQTT Protocol is the most commonly implemented protocol in the IoT area, due to the compatibility of MQTT with lightweight data to run on it. As for health data, it is far from light data classification. There should be more resilient protocols for use in large scale data or continuous data. In this study, a comparison between MQTT and other protocols to process health data has conducted. AMQP is an open-source protocol that provides features suitable for the high requirements for exchanging data on ECG data. This study describes the performance between the MQTT protocol and AMQP protocol that is principled in the publisher-subscriber, by comparing the time delay and throughput to measure the data transmission in real-time. This results in the AMQP protocol delay is less than 1 second per transfer, and the throughput gets an average output of 15209265.86 Bit per second for ten attempts, while the MQTT protocol gets 46 seconds per transfer and throughput gets an average output of 17592975.29 Bit per second. Proving that in the case of health data with ECG data as the dataset used, another protocol besides MQTT, which is the AMQP protocol is better in terms of exchanging data in large and continuous capacity.

1 INTRODUCTION

There have been many studies using the protocol with the principle of publisher and subscriber. A protocol is a set of rules and guidelines for communication between data in every step and process of communication to successfully exchanging various messages. The use of protocols does not have a clear standard. However, if one chooses, it will have a significant influence on the exchange of data that occurs, and if the use of the protocol is not following the requirements will result in fatal events in the ongoing exchange of data. One of the most popular protocols is the MQTT protocol. MQTT is commonly used for data exchange in the world; almost all sectors use it, such as industry, security, business (Kodali and Gorantla, 2017)(Perrone et al., 2017) (van der Westhuizen and Hancke, 2018)(Pereira et al., 2019)(Imane et al., 2018) and health data is no exception (Sarierao and Prakasarao, 2018)(Terry and Francis, 2007).

Health data is essential because it requires continuous reliability, safety, and availability in the aspect of data queue for a medical recap because the exchange of data in health requires proper treatment for successful treatment (Das and Ari, 2014).

Not all health data have lightweight data exchange to be sent at one time; there are also large and busy data; for example, an electrocardiogram (ECG). An ECG is a recording of the electrical activity of the heart, which provides essential information about the condition of the heart. ECG heartbeat detection is needed to diagnose heart disease in the early stages (Naik, 2017). ECG has extensive data and real-time. Therefore, this study will compare communication protocols other than MQTT. The MQTT protocol is one of a machine to machine communications that is not suitable if it uses in real-time in large amounts of data. Because in the MQTT Protocol, there are deficiencies in terms of data transfer capacity. AMQP is an open-source protocol that provides features suitable for high requirements. The use of appropriate protocols is one of the crucial things in the case studies raised (Jaikar and Iyer, 2018). AMQP has succeeded in becoming a superior communication protocol in the banking sector. The banking sector requires excellent communication for high-size activi-

Gumelar, D., Rakhmatsyah, A. and Yasirandi, R.
Performance Analysis of AMQP Protocol for Patient Health Data in IoT Case.
DOI: 10.5220/0009907001430150
In *Proceedings of the International Conferences on Information System and Technology (CONRIST 2019)*, pages 143-150
ISBN: 978-989-758-453-4

ties with essential and sensitive data (Anusha et al., 2017). The similarities of the health sector, making AMQP a feasible protocol to be the proposed protocol and compared with MQTT. Therefore, the AMQP Protocol has a vital role in the reliability, security provision, and high level of data interoperability.

2 RELATED WORK

2.1 Comparison of AMQP with Similar Protocols

Protocol able to run according to the needs that exist in the exchange of data on the IoT (Lee et al., 2018). but this is only simulated using architecture without any significant results through research. Furthermore, research on a comparison of existing protocols, such as MQTT, CoAP, AMQP, and XMPP, shown significant results and can be a reference for its development. Architectural designs built for small scalability using MQTT will be reviewed for extensive and valuable security of data using the AMQP protocol (Yi et al., 2016) in need for information transfer over low bandwidth networks that cannot be relied upon for monitoring. Different requirements make protocol performance produce different comparison results (Pohl et al., 2018). An enormous need appears to integrate the device with an external data source for data utilization, and assessment collected effectively to get the best performance out of the requirements (Chaudhary et al., 2017).

2.2 ECG (Electrocardiogram)

The development of the medical world cannot be separated from technological advancements as a tool for doctors to analyze the function of human organs. In analyzing the function of human organs, doctors still read the results of numbers or graphs from assistive devices. One of the numbers and figures that doctors analyze is the result of an electrocardiogram. An electrocardiogram is a test of human heart function. Electrocardiogram functions to record the electrical activity of the heartbeat.

To some extent, the Electrocardiogram will also identify if there is abnormal blood flow. Currently, the doctor is analyzing the results of electrocardiogram records manually. After investigating, the doctor can determine the state of heart function in humans. The activity of exchanging data on an electrocardiogram requires real-time. So, this problem is appealing to be researched. Electrocardiogram recorded data can turn into training data that can be processed computerized so that a significant data exchange can be seen for large data sizes.

2.3 MQTT (Message Queue Telemetry Transport)

Message Queue Telemetry Transport (MQTT) is a lightweight messaging protocol at the application layer that is suitable for IoT and also machine to machine. It has a 2-byte header size, that is why it is has a very small overhead. MQTT (Kim et al., 2015) follows asymmetrical architecture. MQTT is good for small data exchanges in the IoT world; its role in monitoring systems in health data is crucial. MQTT server can be considered as the broker and MQTT clients as the publisher and subscriber. A broker is a repository of topics published by publishers, and customers subscribe to this topic that has been posted by brokers. There are most brief sessions between the client server defining the connection of the client and the server. Yet logically attached to the subject of interest in the case of a subscription. The user will exchange messages with this subject when a client subscribes to an exciting topic. Using this protocol is useful in situations where small messages requiring less bandwidth are exchanged. In small data exchange cases, MQTT can be the right choice.

2.4 AMQP (Advanced Message Queueing Protocol)

Advanced Message Queueing Protocol (AMQP) is [laced at the application layer for protocol-oriented middleware services in applying the IoT concept (Hong et al., 2018). AMQP protocol designed to allow interoperability between different applications and systems. This AMQP interoperability feature is vital because it provides various platforms to implement. AMQP works using the publish-subscribe concept, devices that perform the Publish process are called Producers, the Subscribe Process are called Consumer. Producer and Consumer-based AMQP with a Server that includes Exchange and Queue messages as a bridge between Producers and Consumers. The producer gave the message to consumers in the form of a topic. Producers can distribute which topics will be sent to Consumers on the Server. In this study, it will be explained that the performance of the AMQP protocol will be better in the health data case study because of the large data requirements, compared to other similar Protocols, which will lead to lower performance.

3 DESIGN AND SCENARIO

3.1 System Planning

The design mechanism for the data queue implementation in sensing nodes consists of hardware, software, and brainware. An explanation of the design of the system mechanism can be seen in Figure 1.

Hardware	PC	PC becomes the Implementation of Making Computer Server and Computer Client, for the Health Data Transmission Process that is connected between 2 Client and 1 Server. Producer as Sender of Data, and Consumer as Receiver of Data, while Server as Monitor and administrator between Producer and Consumer are connected.
Software	Server	Set the Receipt and Request of Data from the Client as Data Traffic
Software	Client	Sending and Requesting Health Data as a Topic on the Server to Access Information provided by the Server computer.
Brainware	User	User gets notifications from hardware and software.

Figure 1: System Mechanism Design.

3.2 Network Architecture

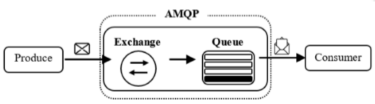

Figure 2: concept exchange Message of AMQP Server.

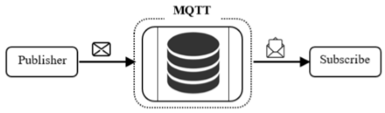

Figure 3: concept exchange Message of MQTT Server.

In the architecture that has been made using two different protocols, namely the AMQP and MQTT Protocol, in testing the protocol will use encrypted ECG data in Excel, the dataset used in this study consists of 11361 ECG data with 202 attributes. These attributes are used by the publisher to send data to the server, and the server will receive it as a topic. The subscriber will request the data that has been sent by the publisher in the messaging protocol. The performance will appear in the server that aims to test the performance of each protocol.

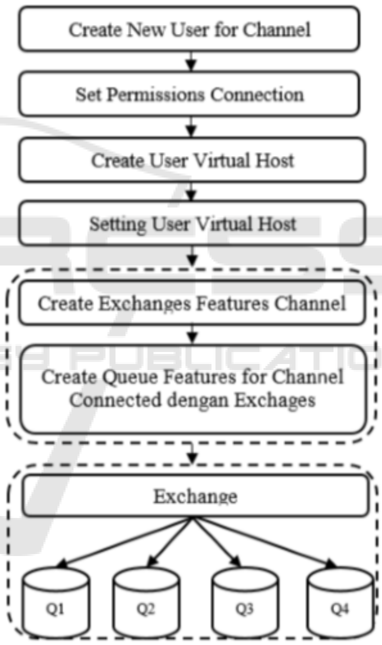

Figure 4: Basic step Manufacture AMQP Server.

In the basic steps of making AMQP server, the initial setup of the server used in the AMQP protocol uses RabbitMQ, which aims to connect between client and server, the initial installation of RabbitMQ has provided the administrator with a username and password in the form of "guest" for server access to be able to see the data exchange performance that occurred. The initial stage of making client access to the server is the administrator to create a channel so that the client gets a username and password, then the channel will set the connection permissions so that the client username can access the server created, then the channel will set the virtual host user to connect to the server, then the user will create an exchange feature that is contained in rabbitMQ with virtual host permissions that have been made before, finally at this stage create a Queue feature to manage each data queue that enters the server by connecting Exchange as a queue regulator if the Queue capacity is Full.

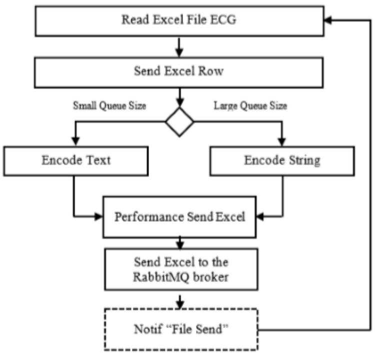

Figure 5: Activity diagram Client Producer AMQP.

In the client producer activity diagram in Figure 5 the first step is to prepare an Excel file electrocardiogram (ECG) data that has been encrypted with a string length in the cell, then the client producer will read each cell with every row sent with different data types according to size, then each row of the data is encoded to release the initial delivery performance. After all the data that has been sent to the server, the process is complete and the producer will send a notification that the file has been sent. The client producer will always run as long as the process is not stopped (Loo ping Forever).

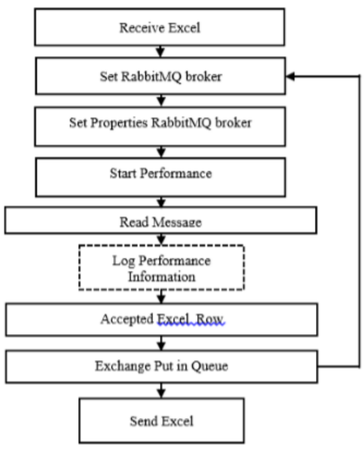

Figure 6: Activity diagram Server AMQP

In the initial stages of the AMQP Server activity diagram in Figure 6, the server receives data in the form of each row in excel sent by the client producer, the data received will be processed and entered into a broker that has been given access by the administrator, then the server will start reading data that has been sent by client producer and output performance data transfer performed by the client, then the server distributes the data that has been read to be entered into the Queue which is managed by the exchange feature and sent to the client consumer who requests the data with the same username access.

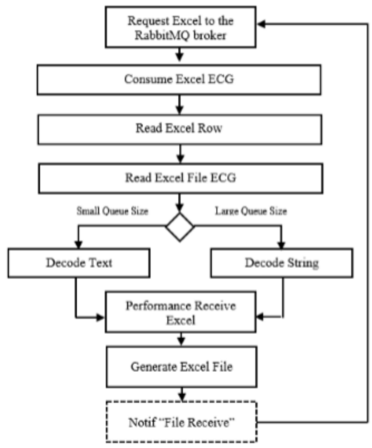

Figure 7: Activity diagram Client Consumer AMQP.

In the initial stages of the activity diagram in Figure 7 for the AMQP consumer client, the client will request data to the server from the same username given by the administrator, then the server will send the data requested by the consumer client, the server will send each excel Row, then the Client consumer will receive it to be decoded so that can create a new file for the results of the receipt that has been done, if the receipt has been completed, then the client consume will give a notification "Receive file" on the running system, but the consumer client that is created always runs as long as the process is not stopped (Looping Forever). Based on the sequences diagram above, the first step is to prepare the server for the initial step to see the exchange of data associated with Wireshark, then the user will send ECG data that has been previously encrypted in the form of Excel. Producer will read the file with the length of the row that will be sent to the server and will delivery performance appears from the producer to the server on Wireshark, the consumer will send a request to the server to get the ECG data that has been sent by the Producer, the request will issue a performance result on the Wireshark because the ECG data is sent to the Consumer, this process will run continuously because it will analyze real-time protocol performance

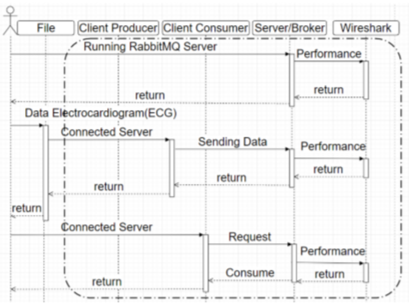

Figure 8: Sequences Diagram.

3.3 Testing

In this study, the test was conducted to see the robustness of the protocol of data exchange that took place in real-time by testing delay and throughput as the most critical aspects of the exchange that occurred.

3.3.1 Delay Testing

Testing is done to get the time needed from publisher to subscriber. In this context, the source is publisher, whose destination is subscriber, from publisher to server, then server to subscriber. In sending data, it will take time to travel from the source to a specific destination to get the ECG Data delay performance that arrives.

3.3.2 Throughput Testing

This test is carried out to see the actual data sent per unit time, the realistic speed of a network, which will be carried out from publisher to subscriber in sending ECG data that is done continuously. The sending will be done to show the results to compare the two performances.

4 RESULT AND DISCUSSION

4.1 Result

Justification in research is using extensive data because the research that has conducted is using small data. In the form of a string length and the results of the test do not show a significant difference from the comparison of protocols carried out, therefore in this study using extensive data. The implementation uses encrypted ECG data in excel form with the protocol used is the AMQP protocol and the MQTT protocol, using the same data exchange concept with the publisher as the sender and the subscriber as the recipient. The test is carried out ten times with one time testing 10 ECG data transfers continuously and will be analyzed to see the results of the performance between protocols.

4.2 Result of Publisher Subscriber Delay

$$Packet1 = (timearrived - timesend) \tag{1}$$

$$Experiment1 = \frac{Packet1 + Packet2 + .. + Packet10}{N} \tag{2}$$

$$AverageDelay = \frac{Experiment1 + .. + Experiment10}{N} \tag{3}$$

Experiment	Time
1	36.739 second
2	42.725 second
3	36.757 second
4	39.316 second
5	41.271 second
6	33.01 second
7	32.121 second
8	43.078 second
9	80.113 second
10	35.597 second
Average	42.0727 second

Figure 9: Average Delay MQTT.

$$AverageDelay/packet = \frac{Delay}{Amountofpacket} \tag{4}$$

In the tests carried out on the MQTT Protocol(Figure 10), the output of the delay results from publisher to subscriber shows the following results. In tests conducted on the AMQP protocol(Figure 11), the output of the delay conducted from producer to consumer shows the following results.

Packet	Delay	Average Delay
773574	36.379	0.00004702717516359130
793402	42.725	0.00005385038101744130
785614	36.575	0.00004655594223117210
781294	39.316	0.00005032164588490380
793277	41.271	0.00005202596318814240
767780	33.01	0.00004299408684779490
772733	32.121	0.00004156804484860880
787549	43.078	0.00005469881874016730
807986	80.113	0.00009915147044627010
772062	35.597	0.00004610640078128440

Figure 10: Average Delay per Packet MQTT.

Experiment	Time
1	0.23 second
2	0.278 second
3	0.284 second
4	0.299 second
5	0.338 second
6	0.314 second
7	0.347 second
8	0.31 second
9	0.277 second
10	0.428 second
Average	0.3105 second

Figure 11: Average Delay AMQP.

4.3 Comparison of MQTT Protocol Delays and AMQP Protocol Delays

$$averagedelay = averagedelayx10000000 \quad (5)$$

From the tests conducted and the results of the diagram that came out(Figure 12), the average of delay MQTT shows 42.0727 seconds for each time the publisher sends a file to the subscriber. In contrast, the AMQP Protocol shows the average of each producer sending the file to the consumer; it only takes 0.3105 seconds. It shows that the availability of data directly and continuously AMQP is far superior compared to MQTT. Because at large enough data size, MQTT is not able to accommodate the queues that are carried out continuously by the publisher, whereas AMQP can accommodate every shipment made by the producer.

4.4 Throughput Test Results

Throughput is a measure of how many units of information a system can process in a given amount of time and usually represented as an average and measured in bits per second (bps). The testing is done 10 times file transfer for 1-time testing, and the results are taken from the publisher to the subscriber, testing is done 10 times with a total number of file transfers 100, here are the results of each test:

Packet	Delay	Average Delay
864365	0.23	0.00000026609129245168
794817	0.278	0.00000034976604677555
823694	0.284	0.00000034478823446571
865505	0.299	0.00000034546305336191
803213	0.338	0.00000042080992215016
843363	0.314	0.00000037231891842540
802919	0.347	0.00000043217310837083
842342	0.31	0.00000036802153994458
836321	0.277	0.00000033121253681302
784259	0.428	0.00000054573807887445

Figure 12: Average Delay per Packet AMQP.

MQTT	AMQP
47.02*10(-6)	0.26*10(-6)
53.85*10(-6)	0.35*10(-6)
46.55*10(-6)	0.34*10(-6)
50.32*10(-6)	0.34*10(-6)
52.02*10(-6)	0.42*10(-6)
42.99*10(-6)	0.37*10(-6)
42.56*10(-6)	0.43*10(-6)
41.56*10(-6)	0.36*10(-6)
54.69*10(-6)	0.33*10(-6)
99,15*10(-6)	0.54*10(-6)

Figure 13: Average Time Each Packages.

$$AverageBps = \left(\frac{Bytes}{TimeSpan}\right)x8bit \quad (6)$$

Bytes	Time Span	Average Bps
878409750	469.443	14969395.65
890978479	428.355	16640001.48
888118684	378.296	18781455.45
885566612	394.479	17959214.3
908979790	405.987	17911505.34
887146353	388.831	18252584.86
900751391	383.105	18809493.82
880903186	397.068	17748157.72
904212478	471.042	15356804.33
888252549	364.39	19501139.97

Figure 14: Throughput MQTT.

Bytes	Time Span	Average Bps
918993607	480.111	15313018.98
887306321	507.73	13980758.61
888997369	425.199	16726236.31
892573476	411.779	17340825.56
899413185	414.259	17369098.75
898262662	461.613	15567372.01
891567186	419.674	16995423.8
898039448	537.753	13359880.06
890368958	500.925	14219597.07
885235800	631.159	11220447.46

Figure 15: Throughput AMQP.

In the table above, the MQTT throughput has the last average of 10 trials with an average number of packets 891,331,927.2 Bytes and an average time span of 208.0996 resulting in an average Bytes / s of 17,592,975.29 or 16.1 Mbit / s. While in AMQP, the throughput obtained with 208.0996 with the last average of 14.7 Mbit / s and the average number of packets of 895,075,801.2 Bytes and Time span of 479.0202. From the results, there were differences in the average packet of 3,743,874 Bytes with the AMQP protocol sending more packets, but the average shows that AMQP is smaller issuing throughput.

5 CONCLUSIONS

From this research, it can be concluded that for a bigger capacity and continuous availability, AMQP protocol is better in terms of exchanging data than the MQTT protocol. The results have been obtained by showing the time delay and throughput to measure the data transmission in real-time. Obtain accuracy when sending health data, with the output that the AMQP protocol delay that comes out is less than 1 second per shipment while the MQTT protocol gets 46 seconds for each delivery. For the throughput testing, the AMQP protocol gets an average output of 15209265.86 Bit per second for ten attempts, while the MQTT protocol gets 17592975.29 Bit per second. The outputs on testing health data using similar protocols with AMQP protocol results better than the MQTT protocol; the availability of the AMQP protocol is better for sending extensive data. This research could speed up the data running process, so there will be no data fallacy or data lose. It can be concluded from the initial problem that the AMQP protocol answers to data exchange on health data because it has various needs, and it is done continuously. The development of advanced research that can be done is more massive data in the security and availability aspects, and many implementations can be done in other sectors to see a better performance other than AMQP protocol.

REFERENCES

Anusha, M., Babu, E. S., Reddy, L. S. M., Krishna, A., and Bhagyasree, B. (2017). Performance analysis of data protocols of internet of things: a qualitative review. *International Journal of Pure and Applied Mathematics*, 115(6):37–47.

Chaudhary, A., Peddoju, S. K., and Kadarla, K. (2017). Study of internet-of-things messaging protocols used for exchanging data with external sources. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 666–671. IEEE.

Das, M. K. and Ari, S. (2014). Ecg beats classification using mixture of features. *International scholarly research notices*, 2014.

Hong, X. J., Yang, H. S., and Kim, Y. H. (2018). Performance analysis of restful api and rabbitmq for microservice web application. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 257–259. IEEE.

Imane, S., Tomader, M., and Nabil, H. (2018). Comparison between coap and mqtt in smart healthcare and some threats. In *2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, pages 1–4. IEEE.

Jaikar, S. P. and Iyer, D. (2018). A survey of messaging protocols for iot systems. *Int. J. Adv. Manag. Technol. Eng. Sci*, pages 510–514.

Kodali, R. K. and Gorantla, V. S. K. (2017). Weather tracking system using mqtt and sqlite. In *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 205–208. IEEE.

Naik, N. (2017). Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. In *2017 IEEE international systems engineering symposium (ISSE)*, pages 1–7. IEEE.

Pereira, E., Pinto, R., Reis, J., and Gonçalves, G. (2019). Mqtt-rd: A mqtt based resource discovery for machine to machine communication.

Perrone, G., Vecchio, M., Pecori, R., Giaffreda, R., et al. (2017). The day after mirai: A survey on mqtt security solutions after the largest cyber-attack carried out through an army of iot devices. In *IoTBDS*, pages 246–253.

Pohl, M., Kubela, J., Bosse, S., and Turowski, K. (2018). Performance evaluation of application layer protocols for the internet-of-things. In *2018 Sixth International Conference on Enterprise Systems (ES)*, pages 180–187. IEEE.

Sarierao, B. S. and Prakasarao, A. (2018). Smart healthcare monitoring system using mqtt protocol. In *2018 3rd International Conference for Convergence in Technology (I2CT)*, pages 1–5. IEEE.

Terry, N. P. and Francis, L. P. (2007). Ensuring the privacy and confidentiality of electronic health records. *U. Ill. L. Rev.*, page 681.

van der Westhuizen, H. W. and Hancke, G. P. (2018). Comparison between coap and mqtt-server to business system level. In *2018 Wireless Advanced (WiAd)*, pages 1–5. IEEE.

Yi, D., Binwen, F., Xiaoming, K., and Qianqian, M. (2016). Design and implementation of mobile health monitoring system based on mqtt protocol. In *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pages 1679–1682. IEEE.