Cloud Providers Services Selection for IoT-Sensors Data Processing

Anton Shykhmat[1] and Zenoviy Veres[2]

Abstract—The Internet of Things (IoT) enables the creation of networks between devices, humans, applications, and the Internet, creating new ecosystems with higher productivity, better energy efficiency, and higher profitability. Nodes in the networks should be able to communicate and exchange data. For this purpose, data protocols are used. This paper provides an overview of two existing data protocols: MQTT and HTTP, compares amount of billed traffic produced by each protocol, protocols cost efficiency, and provides decision tree to select the best fit protocol for particular use cases.

Index Terms— IoT, data streaming protocols, HTTP, MQTT, AWS, GCP, IoT Core.

I. INTRODUCTION

Internet of Things (IoT) is widely adopted within every aspect of our life. IoT enables the creation of networks between devices, people, and applications on the Internet, resulting in ecosystems with higher productivity, better energy efficiency, and greater profitability. Devices help to recognize the state of affairs, which gives them the advantage of anticipating a person's needs based on information gathered by context. COVID-19 increased the remote work demand. It raises tasks to collect, process, store and figure out the insights from the received data. Being able to manage a massive amount of devices within the system is a complex task by itself. An increased number of devices adds extra price to build a robust solution to receive the telemetry data, check their state, and discover disconnected/failed ones proactively [1]. Cloud is commonly considered as a basis to build the solution for the IoT field. The most straightforward (and most challenging at the same time) approach is to use cloud computation capabilities and set up all required components on your own. However, AWS and GCP cloud provides provide Cloud IoT Core [2] modules to set up, manage, and ingest telemetry to the cloud. Both IoT Core solution supports data ingest using two widely adopted protocols in the IoT field: MQTT and HTTP.

The rest of the paper is organized as follows. Section II provides and overview of the following IoT data protocols: MQTT and HTTP. Section III compares amount of billed traffic produced by each protocol and expenses associated with this traffic. Section IV concludes the paper by providing decision tree to select the best-fit protocol for particular use cases.

II. PROTOCOLS OVERVIEW

A. MQTT

MQTT is a messaging protocol for the Internet of Things (IoT) developed and managed by the OASIS MQTT Technical Committee. It is lightweight, open, simple, and designed to be easy to implement. These characteristics make it ideal for use in many situations, including constrained environments such as for communication in Machine to Machine (M2M) and Internet of Things (IoT) contexts where a small code footprint is required and/or network bandwidth is at a premium [3].

MQTT provides the ability to have the device connected indefinitely even if it does not transmit any data. The broker keeps track of connected devices using the keepalive feature. The Keep-Alive is a time interval measured in seconds. It is expressed as a 16-bit word; it is the maximum time interval permitted to elapse between the point at which the Client finishes transmitting one Control Packet and the point it starts sending the next. It is the responsibility of the Client to ensure that the interval between Control Packets being sent does not exceed the Keep Alive value. In the absence of sending any other Control Packets, the Client MUST send a PINGREQ Packet.

When a connection is lost, the broker could issue the client’s Last Will and Testament (LWT) message. The message could be used as a trigger to notify the user about the issue and proactively figure out the disconnect reason. However, disconnect could occur due to connectivity issues and could be a false positive. As a consequence, a more robust approach is required to figure out the device’s failure.

The MQTT specification describes three Quality of Service (QoS) levels [3]:

- QoS 0, delivered at most once
- QoS 1, delivered at least once
- QoS 2, delivered exactly once

Please note, Cloud IoT Core does not support QoS 2. Publishing QoS 2 messages close the connection.

According to the GCP documentation, Cloud IoT Core limits the max inactivity period with idle time set to 20 minutes: "A client connection will automatically be terminated if the client doesn't send any messages for 20 minutes, even if the keep-alive interval is longer. If a keep-alive value isn't supplied, the default idle timeout of 20 minutes still takes effect". [4]

Lviv Polytechnic National University, Ukraine

https://doi.org/10.17758/EARES12.DIR0423102

B. HTTP

HTTP was invented as a World Wide Web component to transfer documents. It is most familiar to us as an enabling technology that allows web browsers to work. Servers contain resources identified by the URLs to which HTTP clients can usually make requests. HTTP is a "connectionless" protocol: devices do not maintain a connection to Cloud IoT Core with the HTTP bridge. Instead, they send requests and receive responses. Cloud IoT Core supports HTTP 1.1 only (not 2.0). HTTP bridge could be used to send the device state to the IoT Core regularly.

REST is an architectural style for building web services based on the HTTP protocol. Services that support this style are called RESTful services. Such services do not store the client's state, making their usage fast, reliable, and scalable. In response to requests made to a resource URI, RESTful services often respond in HTML, JSON, or XML formats (but are not limited to these). RESTful services most often use the following 4 HTTP methods [5]:

1. *GET:* to retrieve resource information only and do not modify it
2. *POST:* to create new resources
3. *PUT:* to update existing resources
4. *DELETE:* to delete a current resource

III. BILLED TRAFFIC COMPARISON

According to the tests performed to deliver 1K messages over MQTT and HTTP - MQTT was shown 6 times faster on the task of posting consistent time-valuable data and is more efficient from a power consumption point of view [6]

Set of scenarios were evaluated to compare the billed traffic by GCP and AWS that represents the most common patterns [7] for data transmission in the IoT field:

1. 1Kb message payload, data is transmitted every minute
2. 1Kb message payload, data is transmitted every 5 mins
3. 1Kb message payload, data is transmitted every 10 mins
4. 1Kb message payload, data is transmitted every 15 mins
5. 1Kb message payload, data is transmitted every 20 mins
6. 1Kb message payload, data is transmitted every 30 mins
7. 1Kb message payload, data is transmitted every hour
8. 1Kb message payload, data is transmitted every 2 hours
9. 1Kb message payload, data is transmitted every 3 hours
10. 1Kb message payload, data is transmitted every 6 hours

For the MQTT bridge, PINGREQ or data message should be delivered at least once per 20 minutes to maintain the connection opened [8]. In GCP, PINGREQ are charged in the same way, as data messages. AWS does not charge for PINGREQ messages, but it charges for total connection time. The GCP minimum billed message is 1Kb, even if the message itself is only a few bytes, for AWS minimum billed message is 5Kb. For HTTP bridge, both GCP and AWS bill every request and response with data transmission. The minimum billed message size is also 1Kb for GCP and 5Kb for AWS. Let’s compare HTTP vs. MQTT bridges with PINGREQ message is transmitted from each device every 20 mins for 10k, 100k, 1M, 10M, and 100M devices. The calculations assume that each device connects/re-connects to the MQTT Bridge only once per day (Connection is billed as 1Kb message by GCP, for AWS it is billed as the size of the message, so let’s assume it also 1Kb). More detailed calculations are presented below for scenarios: 1, 5, 6 and 7.

A. Scenario 1: 1Kb Message Payload, Data is Transmitted Every Minute

The data transmission frequency adds a crucial amount of traffic. PINGREQ messages are approximately 5% of all traffic and are relatively small, and their contribution could be ignored for calculations. HTTP bridge is used almost 2x more messages since it is billed for each request and response separately. As a consequence, the HTTP bridge is not applicable for the scenario with high message transmission frequency.

TABLE I: Billed Traffic for MQTT and HTTP Bridges for Scenario 1 with PINGREQ Message Every 20 Minutes

Devices count	PINGREQ messages traffic Mb/month	Connection’s traffic Mb/month	Telemetry messages traffic, Mb/month	Total AWS MQTT traffic Mb/month	Total GCP MQTT traffic Mb/month	Total HTTP traffic Mb/month
10K	21.6K	300	432.3K	432.6K	453.9K	864K
100K	216K	3K	4.323M	4.326M	4.539M	8.64M
1M	2.16M	30K	43.23M	43.26M	45.39M	86.4M
10M	21.6M	300K	432.3M	432.6M	453.9M	864M
100M	216M	3M	4.323B	4.326B	4.539B	8.64B

https://doi.org/10.17758/EARES12.DIR0423102

B. Scenario 5: 1Kb Message Payload, Data is Transmitted Every 20 Minutes

TABLE II: Billed Traffic for MQTT and HTTP Bridges for Scenario 5 with PINGREQ Message Every 20 Minutes

Devices count	PINGREQ messages traffic Mb/month	Connection’s traffic Mb/month	Telemetry messages traffic, Mb/month	Total AWS MQTT traffic Mb/month	Total GCP MQTT traffic Mb/month	Total HTTP traffic Mb/month
10K	21.6K	300	21.6K	21.9K	43.5K	43.2K
100K	216K	3K	216K	219K	435K	432K
1M	2.16M	30K	2.16M	2.19M	4.35M	4.32M
10M	21.6M	300K	21.6M	21.9M	43.5M	43.2M
100M	216M	3M	216M	219M	435M	432M

As it is presented in Table 2, HTTP Bridge billed traffic is comparable to MQTT in GCP (for AWS MQTT billed traffic is still twice smaller). The difference between bridges is caused by the assumption of MQTT connection/reconnection frequency. This assumption adds approximately 0.7% of extra traffic. The main option to decrease the billed traffic cost is to extend the PINGREQ message time from 20 minutes up to the maximum possible value for MQTT according to the specification - 18 hours []. It will decrease the PINGREQ traffic 54 times per device, as stated in Table 3.

TABLE III: Billed Traffic for MQTT and HTTP Bridges for Scenario 5 with PINGREQ Message Every 18 hours

Devices count	PINGREQ messages traffic Mb/month	Connection’s traffic Mb/month	Telemetry messages traffic, Mb/month	Total AWS MQTT traffic Mb/month	Total GCP MQTT traffic Mb/month	Total HTTP traffic Mb/month
10K	400	300	21.6K	21.9K	22.3K	43.2K
100K	4K	3K	216K	219K	223K	432K
1M	40K	30K	2.16M	2.19M	2.23M	4.32M
10M	400K	300K	21.6M	21.9M	22.3M	43.2M
100M	4M	3M	216M	219M	223M	432M

Increasing the PINGREQ time saves almost 49% of billed traffic for the MQTT bridge in GCP and has no changes for AWS. Since each HTTP transmission is billed as two 1Kb messages compared to 1Kb messages for MQTT, the MQTT bridge is a cheaper and preferred approach to transmit the data for scenarios with frequent data transmissions (less than 20 minutes) and stable network connections. The connection’s traffic shows that MQTT is preferable over HTTP bridge even the reconnect needs to happen every 20 minutes – the connect traffic will be equal to the telemetry messages traffic.

C. Scenario 6: 1Kb Message Payload, Data Is Transmitted Every 30 Minutes

This scenario is evaluated with the following assumptions:

1. PINGREQ message is delivered every 20 minutes for the MQTT bridge
2. The connection/re-connection is made only once per day for the MQTT bridge

TABLE IV: Billed Traffic for MQTT and HTTP Bridges for Scenario 6

Devices count	PINGREQ messages traffic Mb/month	Connection’s traffic Mb/month	Telemetry messages traffic, Mb/month	Total AWS MQTT traffic Mb/month	Total GCP MQTT traffic Mb/month	Total HTTP traffic Mb/month
10K	21.6K	300	14.4K	14.7K	36.3K	28.8K
100K	216K	3K	144K	147K	363K	288K
1M	2.16M	30K	1.44M	1.47M	3.63M	2.88M
10M	21.6M	300K	14.4M	14.7M	36.3M	28.8M
100M	216M	3M	144M	147M	363M	288M

Thirty minutes telemetry messages interval is when the GCP MQTT bridge requires 26% more billed traffic than the HTTP ones, but AWS MQTT bridge requires 48.95% less billed traffic than the HTTP.

D. Scenario 7: 1Kb Message Payload, Data Is Transmitted Every Hour

This scenario is evaluated with the following assumptions:

1. PINGREQ message is delivered every 20 minutes for the MQTT bridge
2. The connection/re-connection is made only once per day for the MQTT bridge

As presented in Table 5, MQTT PINGREQ messages add more than 74% of billed traffic to the GCP MQTT bridge. This contribution increases when the message delivery is done less frequently: PINGREQ messages add 83% of billed traffic if telemetry message is delivered every second hour, and it adds 90% of traffic for scenario 10 (telemetry message is delivered every 6 hours).

TABLE V: Billed Traffic for MQTT and HTTP Bridges for Scenario 7

Devices count	PINGREQ messages traffic Mb/month	Connection’s traffic Mb/month	Telemetry messages traffic, Mb/month	Total AWS MQTT traffic Mb/month	Total GCP MQTT traffic Mb/month	Total HTTP traffic Mb/month
10K	21.6K	300	7.2K	7.5K	2.91K	14.4K
100K	216K	3K	72K	75K	291K	144K
1M	2.16M	30K	720M	750K	2.91M	1.44M
10M	21.6M	300K	7.2M	7.5M	29.1M	14.4M
100M	216M	3M	72M	75M	291M	144M

E. Expenses comparison

The billed traffic size is not the only criterion to select the solution for connecting IoT devices to the cloud. Expenses are also a crucial point for a business that constraints architecture design. As mentioned earlier, AWS does not charge for PINGREQ messages, but it charges for device connection time. Table 6 represents expenses calculation for different transmission scenarios using GCP MQTT bridge with PINGREQ frequency 20 mins, Table 7 represents expenses calculation for AWS MQTT bridge when device is connected to MQTT bridge for the whole day, Table 8 represents expenses calculation for GCP HTTP bridge, and Table 9 represents expenses calculation for AWS HTTP bridge (scenarios 6-10)

TABLE VI: Traffic Expenses for GCP MQTT Bridge, 1Kb Message Payload

Devices count	Data transmission every				
	30 mins	1 hour	2 hours	3 hours	6 hours
10K	$162,23	$129,83	$113,63	$108,23	$102,82
100K	$1 350,50	$1 206,5	$1 134,50	$1 092,39	$1 038,38
1M	$7 884,50	$6 444,5	$5 724,50	$5 484,50	$5 244,5
10M	$150 848,88	$118 448,88	$102 248,88	$96 848,88	$91 448,88
100M	$1 620 998,88	$1 296 998,88	$1 134 998,88	$1 080 998,88	$1 026 998,87

TABLE VII: Traffic Expenses for AWS MQTT Bridge, 1Kb Message Payload

Devices count	Data transmission every				
	30 mins	1 hour	2 hours	3 hours	6 hours
10K	$49,64	$42,34	$38,69	$37,47	$36,26
100K	$496,40	$423,4	$386,9	$374,7	$362,60
1M	$4 872	$4 234	$3 869	$3 747	$3 622,6
10M	$45 960	$40 850	$38 160	$37 184	$36 216
100M	$453 300	$402 200	$376 650	$368 110	$359 640

TABLE VIII: Traffic Expenses for GCP HTTP Bridge, 1Kb Message Payload

Devices count	Data transmission every				
	30 mins	1 hour	2 hours	3 hours	6 hours
10K	$128,48	$63,68	$31,28	$20,47	$9,67
100K	$1 200,50	$646,88	$322,88	$214,87	$106,87
1M	$6 384,50	$3 504,50	$2 064,50	$1 584,50	$1 078,88
10M	$117 098,88	$52 298,88	$19 898,88	$10 224,50	$5 424,50
100M	$1 283 498,88	$635 498,88	$311 498,88	$203 498,88	$95 498,88

TABLE IX: Traffic Expenses for AWS HTTP Bridge, 1Kb Message Payload

Devices count	Data transmission every				
	30 mins	1 hour	2 hours	3 hours	6 hours
10K	$29,2	$14,6	$7,3	$4,87	$2,43
100K	$292	$146	$73	$48,7	$24,3
1M	$2 536	$1 368	$730	$487	$243
10M	$21 140	$10 920	$5 810	$4 096	$2 144
100M	$205 100	$102 900	$51 800	$34 790	$17 710

The HTTP bridge could be used for data transmission for 10k up to 1M devices when it happens every 30 minutes or less often. The price comparison shows a massive increase between 1M and 10M devices in GCP. While the device count increased by 10, the GCP MQTT expenses increased 18x for 6 hours of data transmission and almost 20x for 30 mins transmission frequency. The HTTP bridge price is also growing 10x between 1M and 10M devices. For AWS situation is different. Price per message decrease when you produce more traffic, and as we can see HTTP protocol is cheaper, even it produces more billable traffic. Also, we may notice, that AWS MQTT bridge is 3x cheaper than GCP MQTT bridge, and AWS HTTP bridge is 6x cheaper than GCP HTTP bridge.

As mentioned in the Scenario 5 section, the GCP MQTT could be configured to deliver a PINGREQ message every 18 hours to save traffic. The expenses for such conditions (scenarios 1-5) are represented below:

TABLE X: Traffic Expenses for GCP MQTT Bridge, 1Kb Message Payload with PINGREQ Message Every 18 hours

Devices count	Data transmission every				
	1 min	5 mins	10 mins	15 mins	20 mins
10K	$1 489,90	$390,83	$196,42	$131,63	$99,23
100K	$9 278,50	$2 366,50	$1 502,50	$1 214,50	$1 002,37
1M	$182 213,88	$26 693,88	$9 404,50	$6 524,50	$5 084,50
10M	$1 934 648,87	$379 448,88	$185 048,88	$120 248,87	$87 848,88
100M	$19 458 998,88	$3 906 998,87	$1 962 998,87	$1 314 998,88	$990 998,87

As of AWS, we can decrease charges, if we do not need to connect device to MQTT bridge for the whole day, but only for specific working hours. The expenses for 8 connection hours per day (scenarios 1-5) are represented in Table 11 (for MQTT) and in Table 12 (for HTTP)

TABLE XI: Traffic Expenses for AWS MQTT Bridge, 1Kb Message Payload, 8 Hours per Day

Devices count	Data transmission every				
	1 min	5 mins	10 mins	15 mins	20 mins
10K	$157,68	$40,88	$26.28	$21,41	$18,98
100K	$1 484,8	$408,8	$262,8	$214,1	$189,8
1M	$12 088	$3 704	$2 536	$2 141	$1 898
10M	$114 580	$32 820	$22 600	$19 191	$17 490
100M	$1 139 500	$321 900	$219 700	$185 610	$168 600

TABLE XII: Traffic Expenses for AWS HTTP Bridge, 1Kb Message Payload, 8 Hours per Day

Devices count	Data transmission every				
	1 min	5 mins	10 mins	15 mins	20 mins
10K	$292	$58,4	$29,2	$19,47	$14,6
100K	$2 536	$584	$292	$194,7	$146
1M	$21 140	$4 788	$2 536	$1 757	$1 368
10M	$205 100	$41 580	$21 140	$14 329	$10 920
100M	$2 044 700	$409 500	$205 100	$136 990	$102 900

Overall AWS IoT Core is 5 times cheaper than GCP IoT Core for both MQTT and HTTP protocols, and MQTT bridge in AWS is about 45% cheaper than HTTP bridge in AWS for frequent data transmission scenarios (<5 mins).

IV. CONCLUSIONS

The data transmission depends on the business requirements. Set of use cases and more detailed approach for bridge selection is described above. The major driver for the proper selection – the telemetry delivery frequency and devices count. As mentioned earlier, GCP IoT Core is more expensive comparing to AWS IoT Core for all evaluated scenarios, so it is not recommended to use. For almost all cases usage of AWS IoT Core MQTT bridge is applicable for frequent data delivery. The device count increase (>10M) with high frequency of data delivery (approx. every 1 minute) push the solution to use either standalone MQTT broker or figure out some other TCP-based protocol, e.g. CoAP [10]. If data transmission is going to happen less frequently than every 10 minutes, HTTP-bridge could be a solution up to 100M devices. Considering IoT Use Case Adoption Report 2021 [9], MQTT is a good choice for remote asset monitoring, vehicle fleet management, location tracking and on-site track & trace use cases. HTTP is a good choice for IoT-based process automation and predictive maintenance use cases. The decision table with recommended options for telemetry delivery as a summary of the research is depicted below:

TABLE XIII: Protocols Selection Decision Tree

Devices count	Messages sent		Data transmission every		Decision
	24 hours per day	8 hours per day	<10mins	>10mins	
<10K	Y	N	Y	N	AWS MQTT
<10K	Y	N	N	Y	AWS HTTP
<10K	N	Y	Y	N	AWS MQTT
<10K	N	Y	N	Y	AWS HTTP
<100K	Y	N	Y	N	AWS MQTT
<100K	Y	N	N	Y	AWS HTTP
<100K	N	Y	Y	N	AWS MQTT
<100K	N	Y	N	Y	AWS HTTP
<1M	Y	N	Y	N	AWS MQTT
<1M	Y	N	N	Y	AWS HTTP
<1M	N	Y	Y	N	AWS MQTT
<1M	N	Y	N	Y	AWS HTTP
<10M	Y	N	Y	N	AWS MQTT
<10M	Y	N	N	Y	AWS HTTP
<10M	N	Y	Y	N	AWS MQTT
<10M	N	Y	N	Y	AWS HTTP
<100M	Y	N	Y	N	Custom MQTT broker or CoAP
<100M	Y	N	N	Y	AWS HTTP
<100M	N	Y	Y	N	Custom MQTT broker or CoAP
<100M	N	Y	N	Y	AWS HTTP
>100M	Y	N	Y	N	Custom MQTT broker or CoAP
>100M	Y	N	N	Y	AWS HTTP
>100M	N	Y	Y	N	Custom MQTT broker or CoAP
>100M	N	Y	N	Y	AWS HTTP

REFERENCES

[1] IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things / D.Hanes, G. Salgueiro, P. Grossetete, R. Barton, J. Henry. – Indianapolis, Indiana: Cisco Press, 2017

[2] Paola Pierleoni, Roberto Concetti, Alberto Belli, and Lorenzo Palma. Amazon, google and microsoft solutions for iot: architectures and a performance comparison. IEEE access, 2019 https://doi.org/10.1109/ACCESS.2019.2961511

[3] S. Misra, A. Mukherjee, A. Roy. “Introduction to IoT”. Cambridge: Cambridge University Press, 2021 https://doi.org/10.1017/9781108913560

[4] Google Cloud Platform. Publishing over the MQTT bridge. Access mode: https://cloud.google.com/iot/docs/how-tos/mqtt-bridge#idle_time_limit

[5] N.M.Shaikh, Y.Ingle. “Application of Restful APIs in IOT”: A Review. Haryana: iJRASET, 2021

[6] R.A.Atmoko, R.Riantini, M.K.Hasin. “IoT real time data acquisition using MQTT protocol”. Journal of Physics Conference Series, Vol 853, 2017, pp. 1-6 https://doi.org/10.1088/1742-6596/853/1/012003

[7] M. Amadeo, C. Campolo, A. Iera and A. Molinaro, "Information Centric Networking in IoT scenarios: The case of a smart home," 2015 IEEE International Conference on Communications (ICC), London, UK, 2015, pp. 648-653

[8] E. Longo, A. Redondi, M. Cesana, A. Arcia-Moret, Andrés & Manzoni. MQTT-ST: a Spanning Tree Protocol for Distributed MQTT Brokers. 2020 IEEE International Conference on Communications, 2020 https://doi.org/10.1109/ICC40277.2020.9149046

[9] IoT Use Case Adoption Report 2021. Access mode: https://iot-analytics.com/product/iot-use-case-adoption-report-2021-2/

[10] Alabbas Alhaj, “Constraint application protocol (CoAP) for the IoT”. – Frankf. Univ. Appl. Sci., 2018, p. 1