International Research Journal of Engineering Science, Technology and Innovation Vol. 11(1) pp. 1-8, January, 2025
Available online http://www.interesjournals.org/IRJESTI
Copyright ©2025 International Research Journals

Review Article

Analyzing Data Usage and Cost Considerations among Cloud Service Providers for IoT Sensor Data Processing

Anton Shykhmat* and Zenoviy Veres

Department of Computerized Automatic Systems, Lviv Polytechnic National University, Ukraine

*Corresponding Author's E-mail: anton.o.shykhmat@lpnu.ua

Received: 16-September-2023; Manuscript No: irjesti-24-114040; **Editor assigned:** 19-September-2023; Pre-QC No: irjesti-24-114040 (PQ); **Reviewed:** 03-October-2023; QC No: irjesti-24-114040; **Revised:** 10-January-2025; Manuscript No: irjesti-24-114040 (R); **Published:** 17-January-2025, DOI: 10.14303/2315-5663.2025.118

Abstract

The Internet of Things (IoT) enables the creation of networks among devices, people, applications, and the internet, thereby establishing new ecosystems with higher productivity, improved energy efficiency, and increased profitability. Nodes in these networks should have the ability to communicate and exchange data. To achieve this, data transfer protocols are employed; however, the choice of a specific protocol for a given use case is not always straightforward. This article provides an overview of two existing data transfer protocols, MQTT and HTTP, comparing the amount of billable traffic generated by each protocol and the efficiency of protocol expenditures. The research revealed that in comparison to AWS IoT Core, GCP IoT Core is more expensive for all assessed scenarios and is not recommended for use. For scenarios involving frequent data transmission, the optimal solution is to use the MQTT bridge provided by AWS IoT Core. If the number of connected devices exceeds 10 million with high data transmission frequency every 1 minute, considering the use of a standalone MQTT broker or another TCP-based protocol like CoAP is advisable. In cases of less frequent data transmission (every 10 minutes or less), an HTTP bridge may be a suitable solution for up to 100 million devices. As a result of the study, a decision tree has been created to select the best protocol for specific use cases.

Keywords: IoT, Data streaming protocols, HTTP, MQTT, AWS, GCP, IoT core

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 (Hanes D et al., 2017). 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 (Pierleoni P et al., 2019). 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. Cloud is commonly considered as a basis to build the solution for the IoT field (Misra S et al., 2021). 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 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 (Maurya R et al., 2021).

The rest of the paper is organized as follows. Section II provides an 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 (Atmoko RA, et al., 2017).

LITERATURE REVIEW

Protocols overview

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 (Amadeo M, et al., 2015).

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 (Longo E, et al., 2020).

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 (Ali AA, 2018).

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

- 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".

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:

- **GET:** To retrieve resource information only and do not modify it.
- **POST:** To create new resources.
- **PUT:** To update existing resources.
- **DELETE:** To delete a current resource.

DISCUSSION

Billed traffic comparison

According to the tests performed to deliver 1 K 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.

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

- 1 Kb message payload, data is transmitted every minute
- 1 Kb message payload, data is transmitted every 5 mins

- 1 Kb message payload, data is transmitted every 10 mins
- 1 Kb message payload, data is transmitted every 15 mins
- 1 Kb message payload, data is transmitted every 20 mins
- 1 Kb message payload, data is transmitted every 30 mins
- 1 Kb message payload, data is transmitted every hour
- 1 Kb message payload, data is transmitted every 2 hours
- 1 Kb message payload, data is transmitted every 3 hours
- 1 Kb 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. 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 1 Kb, even if the message itself is only a few bytes, for AWS minimum billed message is 5 Kb. For HTTP bridge, both GCP and AWS bill every request and response with data transmission. The minimum billed message size is also 1 Kb for GCP and 5 Kb for AWS. Let's compare HTTP *vs.* MQTT bridges with PINGREQ message is transmitted from each device every 20 mins for 10 k, 100 k, 1 M, 10 M, and 100 M devices. The calculations assume that each device connects/re-connects to the MQTT Bridge only once per day (Connection is billed as 1 Kb message by GCP, for AWS it is billed as the size of the message, so let's assume it also 1 Kb). More detailed calculations are presented below for scenarios: 1, 5, 6 and 7.

Scenario 1: 1 Kb 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 1).

Table 1. 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
10 K	21.6 K	300	432.3 K	432.6 K	453.9 K	864 K
100 K	216 K	3 K	4.323 M	4.326 M	4.539 M	8.64 M
1 M	2.16 M	30 K	43.23 M	43.26 M	45.39 M	86.4 M
10 M	21.6 M	300 K	432.3 M	432.6 M	453.9 M	864 M
100 M	216 M	3 M	4.323 B	4.326 B	4.539 B	8.64 B

Scenario 5: 1 Kb message payload, data is transmitted every 20 minutes

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 2. 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
10 K	21.6 K	300	21.6 K	21.9 K	43.5 K	43.2 K
100 K	216 K	3 K	216 K	219 K	435 K	432 K
1 M	2.16 M	30 K	2.16 M	2.19 M	4.35 M	4.32 M
10 M	21.6 M	300 K	21.6 M	21.9 M	43.5 M	43.2 M
100 M	216 M	3 M	216 M	219 M	435 M	432 M

Table 3. 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
10 K	400	300	21.6 K	21.9 K	22.3 K	43.2 K
100 K	4 K	3 K	216 K	219 K	223 K	432 K
1 M	40 K	30 K	2.16 M	2.19 M	2.23 M	4.32 M
10 M	400 K	300 K	21.6 M	21.9 M	22.3 M	43.2 M
100 M	4 M	3 M	216 M	219 M	223 M	432 M

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 1 Kb messages compared to 1 Kb 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.

Scenario 6: 1 Kb message payload, data is transmitted every 30 minutes

This scenario is evaluated with the following assumptions:

- PINGREQ message is delivered every 20 minutes for the MQTT bridge.
- The connection/re-connection is made only once per day for the MQTT bridge (Table 4).

Table 4. 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
10 K	21.6 K	300	14.4 K	14.7 K	36.3 K	28.8 K
100 K	216 K	3 K	144 K	147 K	363 K	288 K
1 M	2.16 M	30 K	1.44 M	1.47 M	3.63 M	2.88 M
10 M	21.6 M	300 K	14.4 M	14.7 M	36.3 M	28.8 M
100 M	216 M	3 M	144 M	147 M	363 M	288 M

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.

Scenario 7: 1 Kb message payload, data is transmitted every hour

This scenario is evaluated with the following assumptions:

- PINGREQ message is delivered every 20 minutes for the MQTT bridge.
- 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 5. 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
10 K	21.6 K	300	7.2 K	7.5 K	29.1 K	14.4 K
100 K	216 K	3 K	72 K	75 K	291 K	144 K
1 M	2.16 M	30 K	720 M	750 K	2.91 M	1.44 M
10 M	21.6 M	300 K	7.2 M	7.5 M	29.1 M	14.4 M
100 M	216 M	3 M	72 M	75 M	291 M	144 M

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 6. Traffic expenses for GCP MQTT bridge, 1 Kb message payload.

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

Table 7. Traffic expenses for AWS MQTT bridge, 1 Kb message payload.

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

Table 8. Traffic expenses for GCP HTTP bridge, 1 Kb message payload.

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

Table 9. Traffic expenses for AWS HTTP bridge, 1 Kb message payload.

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

The HTTP bridge could be used for data transmission for 10 k up to 1 M devices when it happens every 30 minutes or less often. The price comparison shows a massive increase between 1 M and 10 M 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 1 M and 10 M 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 (Table 10).

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 10. Traffic expenses for GCP MQTT bridge, 1 Kb message payload with PINGREQ message every 18 hours.

	Data transmission every				
Devices count	**1 min**	**5 mins**	**10 mins**	**15 mins**	**20 mins**
10 K	$1 489,90	$390,83	$196,42	$131,63	$99,23
100 K	$9 278,50	$2 366,50	$1 502,50	$1 214,50	$1 002,37
1 M	$182 213,88	$26 693,88	$9 404,50	$6 524,50	$5 084,50
10 M	$1 934 648,87	$379 448,88	$185 048,88	$120 248,87	$87 848,88
100 M	$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 11. Traffic expenses for AWS MQTT bridge, 1 Kb message payload, 8 hours per day.

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

Table 12. Traffic expenses for AWS HTTP bridge, 1 Kb message payload, 8 hours per day.

Devices count	**Data transmission every**				
	1 min	**5 mins**	**10 mins**	**15 mins**	**20 mins**
10 K	$292	$58,4	$29,2	$19,47	$14,6
100 K	$2 536	$584	$292	$194,7	$146
1 M	$21 140	$4 788	$2 536	$1 757	$1 368
10 M	$205 100	$41 580	$21 140	$14 329	$10 920
100 M	$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) (Table 13).

The decision table with recommended options for telemetry delivery as a summary of the research is depicted below:

Table 13. Protocols selection decision tree.

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

CONCLUSION

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 (>10 M) 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. If data transmission is going to happen less frequently than every 10 minutes, HTTP-bridge could be a solution up to 100 M devices. Considering IoT Use Case Adoption Report 2021, MQTT is a good choice for remote asset monitoring, vehicle fleet management, location tracking and on-site track and trace use cases. HTTP is a good choice for IoT-based process automation and predictive maintenance use cases.

REFERENCES

1. Hanes D, Salgueiro G, Grossetete P, Barton R, Henry J (2017). IoT fundamentals: Networking technologies, protocols, and use cases for the internet of things. Cisco Press. Indiana, United States.
2. Pierleoni P, Concetti R, Belli A, Palma L (2019). Amazon, Google and Microsoft solutions for IoT: Architectures and a performance comparison. IEEE access. 8: 5455-5470.
3. Misra S, Mukherjee A, Roy A (2021). "Introduction to IoT". Cambridge University Press. Cambridge, USA.
4. Maurya R, Nambiar KA, Babbe P, Kalokhe JP, Ingle YS, et al (2021). Application of restful apis in iot: A review. Int J Res Appl Sci Eng Technol. 9: 145-151.
5. Atmoko RA, Riantini R, Hasin MK (2017). IoT real time data acquisition using MQTT protocol. J Phys Conf Ser. 853: 012003.

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

7. Longo E, Redondi AE, Cesana M, Arcia-Moret A, Manzoni P (2020). Mqtt-st: a spanning tree protocol for distributed mqtt brokers. IEEE International Conference on Communications. 1-6.

8. Ali AA (2018). Constrained application protocol (CoAP) for the IoT. InIoT Seminar, High Integrity System, Frankfurt University of Applied Science. 1-4.