
Sachin Pawaskar, University of Nebraska at Omaha
Johannes Hofer, University of Nebraska at Omaha

Available at: https://works.bepress.com/sachin-pawaskar/12/
Impact of the Application Layer Protocol on Energy Consumption, 4G Utilization and Performance

How the Application Layer Protocol strengthens Green IoT

Johannes Hofer
Information Systems & Quantitative Analysis
University of Nebraska at Omaha
Omaha, United States
johanneshofer@unomaha.edu

Sachin Pawaskar
Information Systems & Quantitative Analysis
University of Nebraska at Omaha
Omaha, United States
spawaskar@unomaha.edu

Abstract—The IoT communication over our data highways is growing tremendously. In future many device and machine manufacturers will transmit data from customer end-devices to data centers to improve and change the products to the customer’s needs. The focus of the next decades will be on reducing energy consumption and data overhead in order to keep utilization of 4G/5G networks low. This paper analyzes two different communication approaches often used in IoT scenarios. The benchmark compares the protocol MQTT against the RESTful approach which is based on HTTP(s) in terms of energy consumption and performance. The results of this paper highlight that MQTT has strong advantages compared to REST. This is also due to the fact that these application layer protocols were specialized for IoT communication while HTTP was extended to address new requirements.

Keywords—MQTT, REST, Application Layer Protocol, IoT Communication Protocol, Green IoT, Energy Consumption, Performance, 4G Utilization

I. INTRODUCTION

The Internet of Things (IoT) is becoming an enabler in a variety of different domains. It allows us to be more convenient, efficient or to minimize risks. In today’s life sensors and actors are ubiquitous. We are using them daily and sometimes without even knowing it [1]. For example, an average car built in 2018 has approximately 100 to 200 sensors onboard and the number of sensors will increase significantly in the next few years [2]. The field of applications for IoT are endless. There is a big push for smart ecosystems in many directions like in the named example for smart cars. In future we are confronted with smart homes, smart buildings and even smart cities which are driven by IoT. Thus, the number IoT devices will increase exponentially in the next few years. Figure 1 depicts that in 2025 the number of devices will approach the 200 billion mark [3].

A. Cloud Computing

Cloud computing is a fairly established concept which delivers resources on-demand. For instance, it allows startups to operate their services with the rapid elasticity and pay-as-you-go model. This allows growth without big investments into infrastructure. Cloud computing contributed to the success of IoT startups like Nest, which was bought by Google in 2014 [4].

The tremendous number of sensors that are used by IoT applications deliver the data to a central processing unit. In most cases this is a cloud data center which stores and analyzes the data. This implies that the number of servers in the data centers is growing with the increasing number of IoT devices. EMC predicts that from 2013 to 2020 the data of our digital universe will double every two years. In 2020 it will reach the 44 zettabytes (44 trillion gigabytes) mark. Especially emerging markets like China, India and Russia will catch up...
and produce more data in 2017 than the mature European and American markets [5].

B. Infrastructure

Two thirds of the world population have no permanent internet connection. Some telecommunication companies in rural areas and organizations like Google and Facebook addressed this problem and initiated projects with balloons and drones which should bring internet to the emerging markets [6].

The technology behind this projects is mostly LTE. For example, Googles project loon uses a relay of balloons which provides an LTE connection up to 10Mbps for people living in rural areas [7].

Emerging countries skip landlines and thus go often straight for cellphone internet. This makes the markets heavily dependent on 4G / 5G networks. In future these markets have to focus on protocols that are reliable even under poor connectivity circumstances. In addition, the protocols should have less overhead to not overload the fragile system.

C. Energy

The increasing number of data centers results in an increasing energy consumption. Energy in data centers is used for operating servers and for cooling. For instance, the global energy consumption for data centers from 2011 to 2012 increased by 63% [8]. In 2014 approximately 1.62% of the world’s energy consumption was consumed by data centers. The rising need of power creates new challenges, which can be turned into opportunities. Mattin Grao Txapartegy, Technology and Market Analyst at Yole mentions that there is a possibility to decrease the yearly energy consumption by 12.4% by just implementing new technologies [9].

Energy supply is not unlimited thus the energy prices are rising. The biggest operational cost aspect of a data center is energy cost. Big companies are in the meantime building their new data centers in the Midwest like Google did in Council Bluffs, Iowa. Energy costs are lower in the Midwest compared to other regions in the US [10].

With the growing number of IoT devices and data centers the efficient use of energy becomes an important factor. Thus, the way of communicating between IoT devices and data centers will change. The network traffic of the future will contain a lot of small data packets. Focusing on topics like protocol overhead and resource utilization will therefore be a key topic to deal with the increasing empathy on energy.

D. IoT Protocols

IoT application layer protocols are improving the efficiency of the communication. Not only will the data center in terms of energy consumption profit from these protocols. IoT devices itself will also be benefitting from the specialized protocols. The communication will be faster due to the fact that less overhead needs to be transmitted. Batteries of wearables will last longer and the requirements to the computing unit will shrink.

Many IoT devices are connected via 4G/LTE. It is not a secret that many 4G nodes are at certain times overloaded and slower. Especially in conurbations it is a challenge to provide enough resources for all connected devices. The capacities of the telecommunication industry therefore also need to growth with the demand caused by IoT applications. IoT application layer protocols should thus have a positive impact on the 4G capacities due to the fact of smaller overheads.

II. BACKGROUND

The Internet of Things has different requirements compared to our Internet. Machines are transmitting small data packets in a regular manner, sometimes even real-time and that results in a tremendous data traffic. For instance, connected drive is becoming more and more popular. Vendors are interested to analyze and visualize their gathered data to gain insights and develop new business models. Imagine if customers could enter their car and brush up their engine power for one weekend by just simply downloading a new set of vehicle parameters.

A. New requirements of IoT

People demand fast 4G networks. This is a challenge, especially in the conurbations. Consuming demanding services like Netflix or Amazon Prime Video plus the upcoming IoT traffic will create a bottleneck. Many IoT devices are geographically distributed and mobile internet is the only way of connecting devices in an affordable way to the internet. In many cases mobile internet cannot be avoided. The only option is to make the way of communication more efficient [11].

The way to go is to reduce the resource usage. This can be done by reducing the protocol overhead and if there is no need for real-time communication compressing the transmitted data. Many IoT solutions are in addition in the need of having a stable and reliable communication even under poor connectivity circumstances. The solution for these requirements are IoT light-weight protocols [12].

The compromise is to use established standards like TCP/IP to be compatible with many devices and vendors. Thus, light-weight protocols like MQTT or CoAP are sitting on the application layer. MQTT tries to organize and manage itself in a very simple way to not use a lot of resources. In addition, this should lead to a minimized energy consumption. MQTT was created by IBM for satellite communication along oil pipelines. It was invented in 1999 for the purpose of monitoring sensors and steering actors that were placed geographically distributed [13]. In January 2018 the MQTT 5 standard was released by the OASIS technical committee [14].

MQTT is an IoT light-weight protocol which is easy to implement and supported among a variety of different platforms. MQTT has three actors in its system: the broker,
subscribers and publishers. Compared to HTTP which uses the request-response approach, MQTT uses a publish-subscribe mechanism [15]. MQTT therefore is also able to support multi- cast communication [16].

**Persistent sessions** allow the broker to know in which topics a client is interested without defining it. If the client has subscribed to a topic previously, the broker knows it and will deliver the data after the successful connection establishment. To enable persistent sessions the client has to set the clean session flag in the connection packet to false [17].

**Retained messages** are delivered even if the client is offline right now. As soon as the client connects to the broker the data will be transmitted. A retained message can be created by activating the retained message flag in the header. The receiving client is able to identify a retained message by the retained message flag [18].

**Quality of Service** allows MQTT to set the priority of a message. Messages with the priority 0 will be sent with the fire-and-forget principle. This means there is no further investigation if the message arrived at the broker or the subscriber. Priority 1 ensure that the message arrives at least once and the priority 2 means it arrives exactly once [13].

**B. REST**

The REST architectural design pattern uses HTTP standard functionalities to transmit data. HTTP was designed as a protocol that uses the client/server principle and a request/response approach. For IoT applications a bidirectional real-time communication approach is in many cases needed. When HTTP was invented it was not designed for bidirectional real-time communication. Over the time HTTP was extended and adapted and offers nowadays two approaches to address this requirement:

- Websockets
- Long Polling

Some people may think REST is a protocol, but that is not true. REST is an architectural design pattern developed by Roy Fielding in his dissertation in 2000. REST was created as an architectural model that fits to the functionality of the world wide web [19].

RESTful web services are using HTTP and its basic functionalities to deliver an easy to deploy data communication over the web. It uses a resource based approach where clients request and post data to the server. The server sends its response as an answer. The payload of REST is standardized by MIME (Multipurpose Internet Mail Extensions) types [20]. One of the most popular MIME type representation is JSON (JavaScript Object Notation).

As already mentioned REST uses an approach where the client must trigger a request to get a response [20]. If the server has new data for the client it must wait till the client picks the data up. For real-time applications this would theoretically mean the client has to ask the server for new data every few milliseconds. This would lead in a lot of unnecessary traffic and would equal a DOS (Denial of Service) attack if a lot of clients do this.

The solution was to develop a different approach where the server keeps the request open till it has the desired data, then the server sends the response including the data. After receiving the response, the client opens again a request immediately. Some adjustments had to be made to use this approach, for example the timeouts had to be extended in order to give the server enough time to wait for the data and answer the request. This approach is called long polling.

### III. MODEL DEVELOPMENT

This section focuses on the development of a model that should reflect the major aspects influencing 4G utilization, energy consumption and performance. The basic expectation is that MQTT performs better in all named areas compared to REST, and that should be proven with help of this research model.

Some hypotheses have to be viewed from two different point of views, because the variable itself can react different depending if the device is receiving or sending. Thus, some variables were monitored on the receiving (R) and sending (S) device.

The basic statement of the research model is that MQTT indirectly impacts the energy consumption of the processing device. This is due to lower resource utilization on random access memory and processing power. MQTT saves connection meta data on the server side, thus the protocol overhead should be smaller. This results again in less energy consumption and in addition in a higher throughput. The throughput was defined as measure that indicates how many transactions can be done in a certain amount of time. Response time is an extraneous variable which cannot be fully controlled. Response time is dependent on processing times within the network and aspects like electromagnetic interferences. Figure 2 summarizes all hypotheses in one visualization.

![Proposed Research Model](image-url)
The following hypotheses assume that MQTT does a better job than REST. Thus, the hypotheses are always formulated in relation to REST.

LTE traffic is rising, thus it is important to keep the packets and meta data as small as possible. MQTT addresses this need by avoiding to send connection meta data with each and every packet.

**H1:** MQTT does not include the connection meta data in each packet, thus the protocol overhead is smaller.

If the overhead of a packet is small, the frame that is submitted on the medium (LAN) is proportional smaller too.

**H2:** The reduced protocol overhead of MQTT leads to less bytes that are going to be transmitted on the medium.

If the number of bytes transmitted on the medium is smaller, more data can be packed into packets and submitted in a certain amount of time.

**H3:** Less bytes transmitted on the medium leads to the expectation that more data can be transmitted in a specified timeframe.

MQTT features like persistent sessions, retained messages, last will and testament are reducing the resource usage. In addition, MQTT libraries are very simple and small in order to be compatible with low-cost microprocessors.

**H4(R):** Receiving device: The smaller libraries and more IoT-aligned functionalities of MQTT are leading to less memory utilization.

**H5(S):** Sending device: The smaller libraries and more IoT-aligned functionalities of MQTT are leading to less memory utilization.

**H6(R):** Receiving device: The simple structure of MQTT utilizes less processing power.

**H7(S):** Sending device: The simple structure of MQTT utilizes less processing power.

The combination of the previous hypotheses is leading to a reduced energy consumption. Less bytes needs to be transmitted to submit the same amount of data while using less resources.

**H8(R):** Receiving device: MQTT leads to a better energy consumption compared to RESTful data transmissions

**H9(S):** Sending device: MQTT leads to a better energy consumption compared to RESTful data transmissions

### IV. RESEARCH METHOD

To compare the two application layer protocols the authors decided to first conduct an experiment to gather the data once for the MQTT and once for the RESTful scenario. In the next step the data was analyzed and compared. Benchmarking is the preferred used instrument for comparison in the Information Technology and was thus also used for this project.

In the experiment a derivation of the pre- and post-observation approach was used. This allowed the authors to differentiate the idle status of the IoT device to the actual transmitting or receiving process. The values captured during the pre- and post-observation were averaged and could then be compared with each other. Figure 3 shows how the differences of a pre- and post-observation look like in an Excel chart.

![Example Graph for Power Consumption](image)

**Fig. 3:** Example Graph for Power Consumption

The authors decided to use Raspberry Pi’s for the experiment. The Raspberry Pi is very popular and common used multi-purpose computer that supports variety of different interfaces and protocols. Figure 4 depicts the devices, which were connected with each other over a category 6 LAN cable. The RESTful experiment used the Tornado and the MQTT experiment used the PAHO framework. Both libraries were used in python scripts which were written for this experiment. The MQTT broker was a Mosquitto MQTT server. For both experiments TLS encryption was used.

![Used devices](image)

**Fig. 4:** Used devices

Figure 5 shows that an oscilloscope was used to measure the energy consumption on the sending and receiving IoT device. The packets and frames were captured with Wireshark. The CPU and RAM utilization was measured with a script on the device itself. This is not an ideal circumstance because the measurement itself influences the measured values. But it affects both experiments the same way and thus the values stay comparable. The throughput was measured by simply counting the transmitted data packets within a 60 seconds time window.
The experiment was conducted three times. There were no big deviations between each instance of the experiment, thus the authors decided to average the results of the three experiments. This is more representative and reduces noise.

To investigate if the protocols differ from each other depending on how much data is sent, the authors decided to transmit six messages for each protocol with different message sizes.

**TABLE I. MESSAGE SIZES**

<table>
<thead>
<tr>
<th>Message No.</th>
<th>Message Sizes Represented</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>typical sensor value</td>
<td>6</td>
</tr>
<tr>
<td>#2</td>
<td>~ one sentence</td>
<td>100</td>
</tr>
<tr>
<td>#3</td>
<td>~ one MS Word paragraph (plain text)</td>
<td>500</td>
</tr>
<tr>
<td>#4</td>
<td>~ two MS Word paragraphs (plain text)</td>
<td>1000</td>
</tr>
<tr>
<td>#5</td>
<td>~ 22 MS Word pages (plain text)</td>
<td>100,000</td>
</tr>
<tr>
<td>#6</td>
<td>~ 216 MS Word pages (plain text)</td>
<td>1,000,000</td>
</tr>
</tbody>
</table>

V. RESULTS

This section shows the results of the experiment. The focus of the key results is on energy consumption, overhead and performance.

Due to the fact that transmissions always take a certain time, it is not possible to simply take the energy consumption in Ampere or Watts. The energy consumption has to be put in relation to the transmission time. The authors decided to use Ampere seconds which can easily converted into Watthours with the constant of 5 Volts with the formula

\[ P \text{ (Watts)} = U \text{ (Voltage)} \times I \text{ (Ampere)} \]  

(1)

Figure 6 and 7 show the results for hypothesis 8 and 9. The messages were sent and received within one connection. What is clearly visible is that MQTT needs significantly less energy during the connecting and disconnecting process due to the fact that it does not need to send its meta data every time (the broker already knew the clients in this scenario).

During the transmission of the messages MQTT has a significant lower energy consumption. This is true for the sending as well as the receiving device.

\[ \text{Overhead} = \text{Sniffed Bytes (Frame)} - \text{Message Size} \]  

(2)

Due to the fact that all six messages were sent during one connection the overhead is presented for a transmission of 1075 Kilobytes (sum of all messages sizes). Figure 8 shows the difference of the overhead for both protocols. The overhead from MQTT was in this case 36% lower.
Fig. 8. Overhead comparison for 1075 KB

The throughput is closely related to the overhead but also dependent from the response time. The expectation that less overhead paired with short processing times leads to the possibility to send more messages per minute could be proven. MQTT allows to send 3.85 times the number of 6-byte messages compared to REST. For 1,000,000-byte messages the ratio is 1:2.5. This implies that MQTT communication for larger messages is a little bit more ineffective. However, it is more efficient than REST. Figure 9 visualizes the messages transmitted in 60 seconds for both protocols.

Fig. 9. Number of messages transmitted in 60 seconds

MQTT uses significantly less resources. The Raspberry Pi Version 3 is a rather powerful IoT device. It has a 64bit Quad Core 1.2GHz processor and 1GB RAM. [21] Therefore, the resource consumption in percent is a rather small number. MQTT utilizes 34% less CPU and 4% less RAM resources while sending. During receiving MQTT utilizes 10% less CPU and 4% less RAM.

VI. CONCLUSION

MQTT was used as an example for an IoT application layer protocol. This experiment underlines that protocols like MQTT and CoAP have strengths that cannot be outperformed by HTTP REST web services. This research could prove that it makes sense for organizations to plan the setup of their infrastructure with IoT application layer protocols. It has a significant impact on their data centers and on the 4G and 5G infrastructure plus the related transmission speed.

A. Impact on Data Centers

Data center infrastructure is growing with its connected clients. IoT application layer protocols are more efficient due to the fact that they have less overhead. This also simplifies the processing during the receiving and sending process. In the end that means that data centers need less servers because more clients can be handled by one server. In, addition to the cost and power savings of having less servers, IoT application layer protocols consume significant less energy to transmit the same amount of data.

B. Impact on 4G / 5G Infrastructure

The capacity of the 4G infrastructure is in many big cities already on the edge. More and more devices are added to the network every month. IoT application layer protocols reduce the network traffic and should therefore improve the situation in the congested infrastructure. The proportion of data to overhead is better in MQTT and the throughput is up to 3.85 times significant higher compared to REST. In addition, the QoS of MQTT can handle the reliability of the transmission in areas where the connection quality is poor.

The findings show that IoT application layer protocols were designed for the special purpose of M2M communication. This kind of protocols fulfill the needs and requirements of common IoT devices while REST and other approaches are failing to deliver that functionalities. In, addition to the functionality advantage, IoT application layer protocols use less computing resources and less energy.

VII. FUTURE WORK

This paper addressed two communication protocols in the IoT landscape. There is the opportunity to do way more research in this area. MQTT is one example for a light-weight protocol. An interesting idea would be to compare CoAP,
MQTT and commonly used web services like REST and SOAP. The outcome of the research could underline why it is important to count on IoT light-weight protocols in future.

Future research directions could be to investigate the nodes in between of the communication. How are they affected by using IoT protocols compared to commonly used approaches? Another interesting aspect would be to compare the performance of web sockets to IoT application layer protocols.

Another research direction that is targeted by the authors is the investigation of the impact of IoT application layer protocols in terms of performance and energy consumption on the IoT devices itself and resulting opportunities for the IoT market.

REFERENCES


