WHITE PAPER

Solving the complexity of communicating between IoT devices and the enterprise

Overcoming IoT challenges using solutions based on the industry-standard MQTT protocol

Abstract

On the face of it, getting data from an IoT device to the cloud may appear straightforward, but before you begin any IoT project, you need to consider how you'll solve a variety of communication challenges. Something that is so much more than just the connectivity. This paper explores how to overcome the issues of complexity, cost, and availability, by building solutions based on the industry-standard MQTT protocol. It also looks at the low-power benefits of MQTT-SN for IoT applications.

White paper: Solving the complexity of communicating between IoT devices and the enterprise

Author: Nick Hayes, Senior Principal Engineer, Product Center Services,
u-blox AG

Contents

Background / Executive summary

On the surface, designing and building an Internet of Things (IoT) solution might seem simple. You have some IoT devices, the data they collect, a network, and a cloud platform to receive and process that data. But when you look at the detail of implementing all these elements, it's not so simple after all. Before launching any IoT project, there are important questions to address. What connectivity should you use? Which network? How will it scale? Is it secure? What about power consumption and data efficiency? What interface will you use to get data to the enterprise? What will all this cost?

Finding the answers is crucial to any IoT project's success.

Today's IoT device makers need design partners with proficiency in connecting to the cloud via cellular networks. When designing your solution, you want simple connectivity that can work globally, with minimal setup, and that's easy for customers to install. One of the key challenges is around getting data from the IoT device to the cloud-based management platform.

In this white paper, we examine the hidden complexities of communicating between the IoT device and the enterprise, and explain how communication is so much more than just connectivity. We then demonstrate that challenges such as security, efficiency, and availability can be overcome by solutions based on the MQTT protocol.

Introduction

Common characteristics of IoT use cases
Across most IoT use cases, from connected healthcare to industrial monitoring, and from smart cities to asset-tracking, a range of key characteristics emerges:

- **Message size:** The message or payload size is small, typically in the order of 50-500 bytes.
- **Message frequency:** The message frequency ranges from sending data every few minutes, to sending updates daily or even weekly, or in some cases only sending during an exception condition.
- **Battery life:** The battery might need to last a week between charges or may need to operate for many years in the field without ever being recharged or replaced.

These three characteristics form the key goalposts for design success in resource constrained IoT devices.

A low-power solution is more than just LPWAN
Many IoT products operating under these types of constraints will use low power wide area networks (LPWAN) to benefit from the power-efficiency associated with LTE-M or NB-IoT radio technologies, as well as features such as power-save mode (PSM) and extended discontinuous reception (eDRX).

However, the network operators ultimately determine the effectiveness of these features, and their settings can become cumbersome to manage, particularly for IoT devices being deployed around the globe.

As a result, fast, secure communication with the cloud has a major impact on device power consumption, and the choice of protocols becomes a key decision in achieving an ultra-low-power solution. Its a decison that can have signigicantly greater impact and can complement network-operator-controlled LPWAN features.

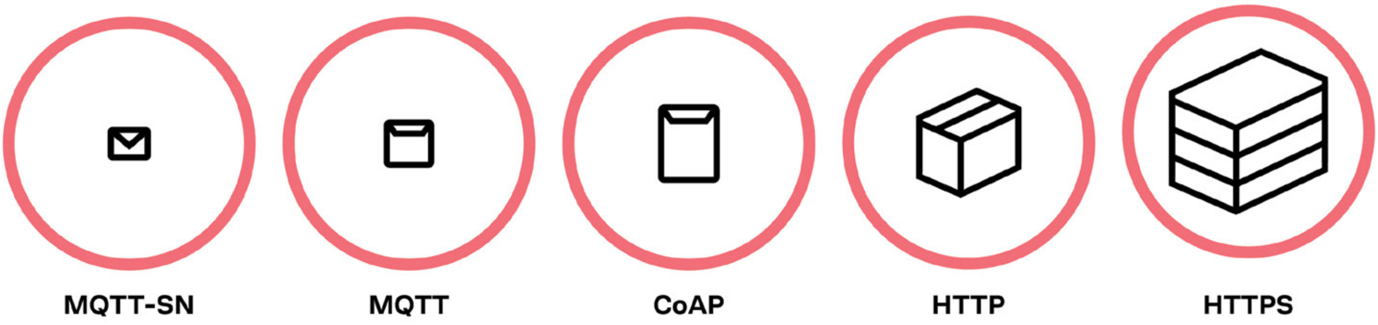
The device-to-enterprise data challenge

Let's consider the hidden complexities of communicating between the IoT device and the enterprise, including around security, efficiency, and availability. We'll then explore ways to overcome these challenges using solutions based on the industry-standard MQTT protocol.

Choosing the right protocol is key
There are two components to consider in any data transmission: the payload (the bytes of data that make up the message itself); and the overhead (the other informational bytes necessary to transmit the payload). Particularly for resource-constrained IoT use cases, keeping careful control of the transmission overhead is crucial in the overall efficiency of the solution.

Think of the communication protocol as the overhead or 'packaging' around the payload or 'message' you want to send. In the figure below, we've illustrated this using familiar parcel packaging. On the far left, the MQTT for Sensor Networks (MQTT-SN) protocol is represented by an A6-sized letter envelope, followed by MQTT as a legal-sized envelope. The Constrained Application Protocol (CoAP) might be represented by a bubble-padded mailer, HTTP by a cardboard box, and lastly on the far right, secured communication protocols such as HTTPS, which would be the size and weight of a wooden palletized box.

By continuing with the packaging analogy, we'll show how choosing the right communication protocol can mean lower message overhead, which overcomes many of the obstacles in the device-to-enterprise data challenge. Let's now look at MQTT specifically.

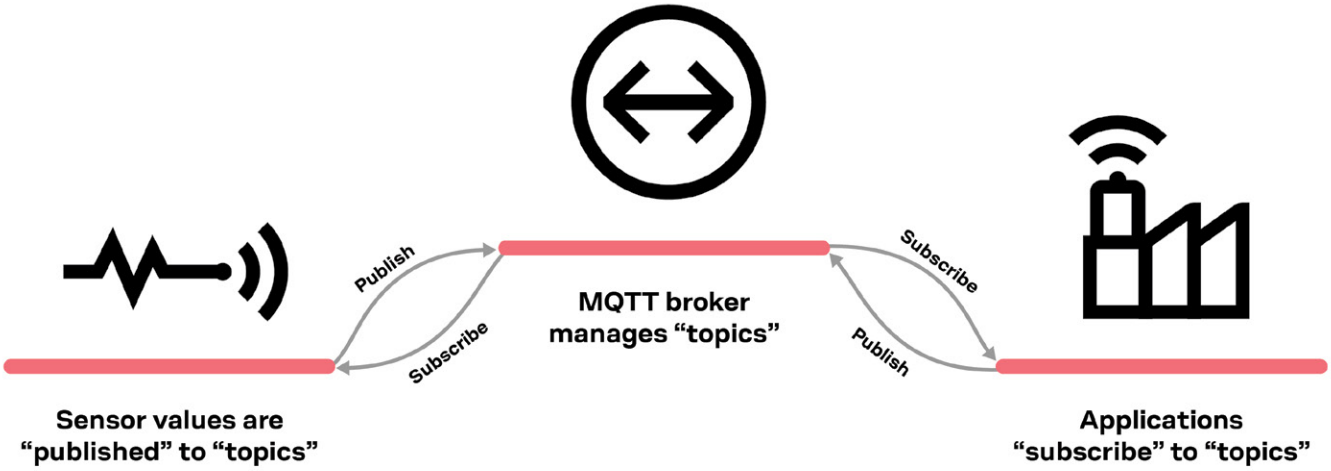


Why MQTT?

MQTT, short for Message Queuing Telemetry Transport, has quickly become one of the most popular protocols for the IoT. It's very straightforward to implement, flexible, lightweight, bi-directional, and has no restrictions on numbers of messages, which means it's scalable.

MQTT is a publish-and-subscribe messaging protocol, designed to create a reliable standard for machine-to-machine (M2M) communication. Users can set the quality of service (QoS) based on the specific requirements of their applications.

Sensors publish their readings to so-called 'topics'. The MQTT broker manages topics. Applications subscribe to the topics as managed by the MQTT broker. The publish-and-subscribe architecture plays an essential role in the efficiency of MQTT. MQTT-SN is further tailored to the needs of power- and bandwidth-constrained devices. MQTT-SN has all the features of regular MQTT, but is not tied to TCP/IP, and introduces an additional quality of service, a connected sleep mode and MQTT topic aliasing, all of which are designed to reduce the amount of data sent over the air, and therefore also the time devices are on-air, and the power consumed.



Connectivity / network choices and power consumption
Many people approach the challenge of selecting the right radio technology and operating network with the assumption that an LPWAN technology is what they need, and then conclude that because NB-IoT has the lowest power consumption, it must be the right choice. However, things are not always as they first seem.

The first question should be around where the IoT device will need to operate. Despite what the coverage maps might have you believe, LTE-M and NB-IoT rollouts are still in their early days, and nationwide coverage is not quite there in most countries or even non-existent in some others. If you have an IoT device that will be moving, or operating in rural locations, relying on LPWAN connectivity alone is likely to leave you with coverage gaps.

NB-IoT is only really suited to stationary applications. In addition, the low data rates provided by NB-IoT mean that if you want to send a payload of 200 bytes or more, or as soon as you add security to your solution, the power savings of NB-IoT are quickly eroded by the long time spent on air during data transmission.

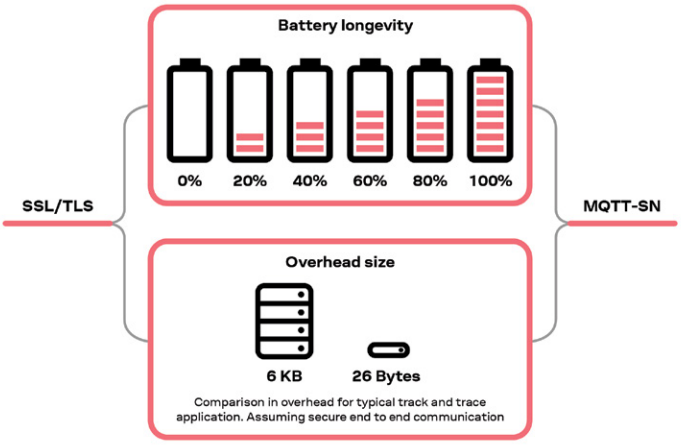
For reliable global connectivity, either an LPWAN module with 2G fallback or an LTE module are the more robust answers. Communication solutions based on the MQTT protocol can greatly contribute to power optimization, making either of these radio technology selections viable options for power-constrained IoT devices.

Security and power consumption

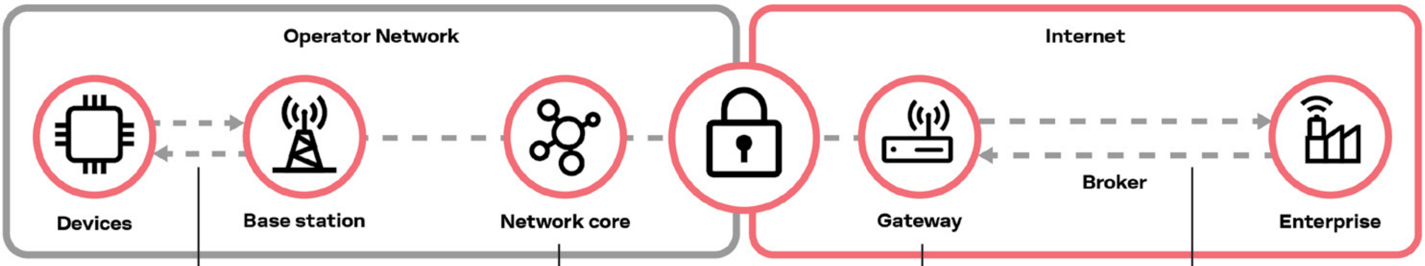
As with the choice of protocol, how you choose to secure your data can have a significant impact on power consumption. In the figure below, we compare the overhead of sending a simple 12-byte ''Hello World'' message using MQTT-SN versus SSL/TLS. A secure end-to-end communication using MQTT-SN requires only 26 bytes of overhead, while SSL/TLS uses 6KB – that's 230x.

With typical payload messages in the order of 50-500 bytes, a 230x overhead penalty for security is beyond inflationary, erasing any gains you might have thought you made in the device-to-enterprise data challenge.

As part of its suite of IoT communication services, u-blox uses a private access point name (APN) to provide secure, end-to-end MQTT-SN communication. Since the APN is private, data is not exposed to the public internet. Authentication is achieved using the SIM identity, a value given directly by the operator core network, and not susceptible to tampering. This means that data can effectively be sent in plain text, with no need to include the overhead associated with additional headers or device identifying data. The data arrives at our MQTT-SN gateway via a secure virtual private network (VPN) and then is sent on to our MQTT broker for delivery to the enterprise. This can be done using a variety of protocols, including MQTT and HTTPS. Of course, this heavier protocol and security transport is managed in the cloud, where there is plenty of processing power.



Comparison in overhead for typical track and trace application. Assuming secure end to end communication



Security between a device and network base station guaranteed by hardware SIM card

Traffic is private and secure inside the operator network

Thingstream MQTT-SN gateway is connected securely to the operator network core via VPN

Enterprises connect to the Thingstream via secure protocols (SSL/TLS)

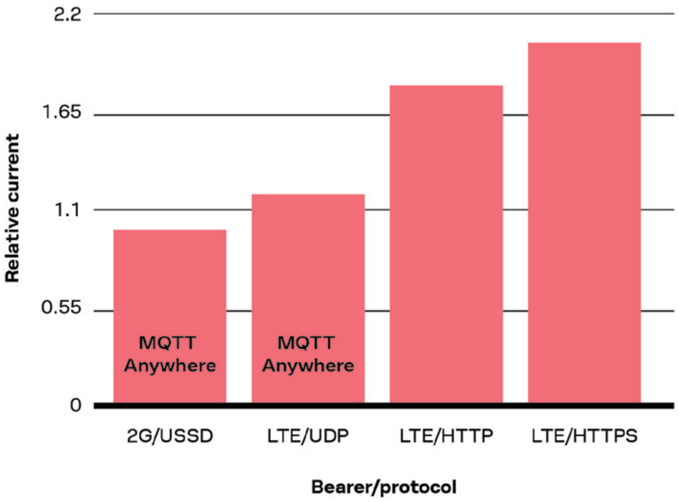
Power consumption and data usage

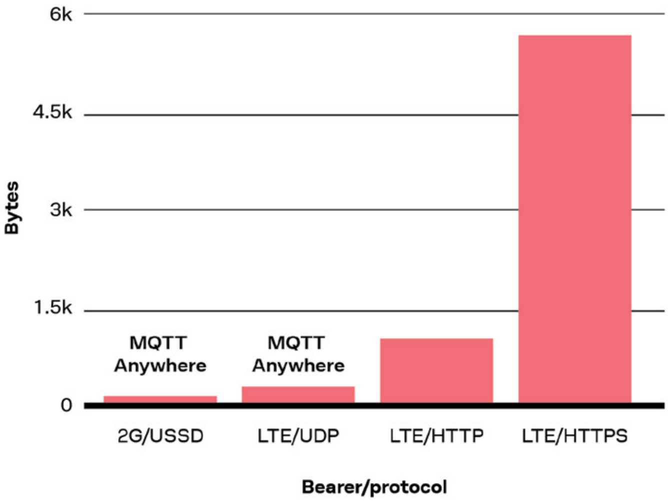
The choice of communication protocol has a major influence on the total number of bytes sent over the air. This in turn impacts the amount of time the IoT device's modem must be connected to the network, and the amount of power required to maintain that connection. MQTT-SN is optimized for power- and bandwidth-constrained environments, providing the lowest overheads and lowest-possible power consumption for such use cases.

We put MQTT-SN to the test by sending a 12-byte "Hello World" message using various bearer protocols. In the figure below, the right-hand bar chart shows the total bytes sent via MQTT-SN over 2G and LTE, and via HTTP and HTTPS over LTE. The left bar chart shows the relative power consumed to send the same 12-byte message.

The results show that protocol choice is important for two reasons:

1. The overhead introduced by the protocol directly impacts the power consumed during message transmission.
2. If the selected protocol is highly inflationary, data usage is dramatically multiplied by the overhead versus the payload, where the actual value of the data lies.

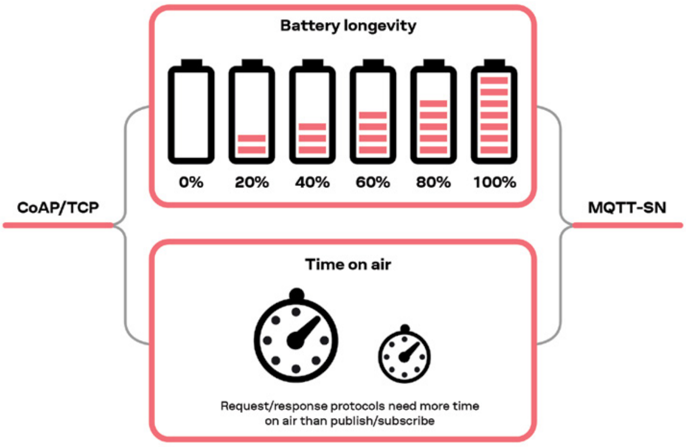




Bearer/protocol	Relative power consumption
MQTT Anywhere 2G/USSD	1.0
MQTT Anywhere LTE Cat-1/UDP	1.21
HTTP/LTE Cat-1	1.81
HTTPS/LTE Cat-1	2.05

Bearer/protocol	Payload (bytes)	Total transferred (bytes)	Inflation factor
MQTT Anywhere 2G/USSD	12	26	2.17
MQTT Anywhere LTE Cat-1/UDP	12	34	2.83
HTTP/LTE Cat-1	12	1034	86.17
HTTPS/LTE Cat-1	12	5676	473

The role of the MQTT client and MQTT broker
Continuing toward solving the device-to-enterprise data challenge, once you have your data out of the IoT device, the next question is how to deliver it to the enterprise. At this point, we revisit the publish-and-subscribe architecture of the MQTT protocol, and the roles of the MQTT client and MQTT broker.


Battery longevity
0%
20%
40%
60%
80%
100%
CoAP/TCP
MQTT-SN
Time on air
Request/response protocols need more time on air than publish/subscribe

Protocol selection impacts the time spent on air and impacts battery longevity. Request/response protocols such as CoAP and TCP need more time on the air than publish-and-subscribe protocols such as MQTT and MQTT-SN. Herein lies the potency of the MQTT broker.

> “Don’t think client and server. Think client and broker.”

MQTT broker:

- Acts like a traffic signal, rather than as storage and distribution.
- Handles authentication of client broker communication.
- Manages connections, sessions, and subscriptions.
- Is responsible for receiving all published messages and sending the messages to all subscribed clients.
- Queues messages for subscribed clients and delivers them according to agreed quality of service.

MQTT client:

- Can be any device (from a microcontroller to a full-fledged server) that runs an MQTT library and connects to an MQTT broker over a network.
- Can publish and subscribe.
- Clients receive messages by subscribing to a topic on the MQTT broker. Messages are published to an MQTT topic, not sent to an individual device. Think of topics as shared email boxes.

Cost
When considering the cost of communication, many IoT device makers look at the cost of a traditional cellular data offering from a network operator, and the data cost per MB. On the surface, you might begin to think about message size, security inflation, and roaming fees. When you look more closely, however, you’ll typically discover many other costs, such as minimum-term line-rental fees, access to a SIM management platform, device certificate management fees, and data/protocol management.

The u-blox IoT Communication-as-a-Service suite has a differentiated approach to pricing. There are pricing plans based on the number of MQTT messages sent (not the cost per MB), with simple, predictable, fixed monthly rates. There are no minimum contract periods. You’re free to activate and deactivate devices whenever you need to, and only pay if a device is activated. The package includes global roaming connectivity, access to the Thingstream IoT service delivery platform, as well as our powerful enterprise-grade MQTT broker, and advanced Data Flow Manager that makes network-edge intelligence, programming, and data management easy.

In this way, you access IoT communications as-a-service, and benefit from value over-and-above what most network operators offer with their own connectivity. The additional value is when the common characteristics of IoT use cases are ideally suited with a solution that solves the problem of device-to-cloud communication, a challenge that traditional cellular connectivity does not fully answer.

IoT Communication-as-a-Service

Getting data from IoT devices anywhere in the world to the enterprise is simplified through MQTT Anywhere, the u-blox IoT Communication-as-a-Service offering, delivered via the Thingstream IoT service delivery platform.

MQTT Anywhere enables long-life, low-power devices and offers:

- Access to 600+ cellular networks in 190 countries, to support global rollouts of IoT devices
- MQTT-SN gateway, enabling efficient, connectionless LPWA communication
- Simple, low-cost, predictable pricing, based on the number of MQTT messages
- An enterprise-grade IoT cloud platform, using auto-scaling technology
- A reliable, high-performance MQTT broker that enables efficient client communication
- Advanced Data Flow Manager, making network-edge intelligence, programming, and data management easy

Summary

By exploring the complexities of communicating between IoT devices and the enterprise, we've shown how communication is so much more than connectivity alone. IoT connectivity is only one component of an overall low-power solution. The choice of protocol is a key decision that directly impacts the overhead of each message, and therefore the energy required to send it. MQTT is an industry-standard messaging protocol shown to be ideally suited for power-constrained IoT use cases. u-blox offers a suite of IoT communications solutions using MQTT. The goal of these IoT Communication-as-a-Service offerings is to simplify your IoT project, from the cost of ownership to the data journey between the IoT device and the enterprise. This provides you with a comprehensive, scalable, end-to-end solution with predictable costs, helping you solve the device-to-enterprise data challenge.

Find out more:
https://www.u-blox.com/en/iot-communication-service

Contact us:
https://www.u-blox.com/en/contact-u-blox-services

About the author

Nick Hayes, Senior Principal Engineer, Product Center Services, u-blox

Nick Hayes is Product Owner for the u-blox IoT Communication-as-a-Service portfolio and the Thingstream IoT service delivery platform. He has over 15 years of experience in communications, starting out with the provision of services to enable email and calendar synchronization for mobile phones in the pre-smartphone era and sub-data services to bring internet-based applications to people with basic handsets and no internet data plan. As mobile technology matured, and smartphones became commonplace, the need for low-cost, reliable, standards-based services grew and attention turned to instant messaging and social networking. Nick was part of the team that built a successful cloud-based white-label chat service widely used by mobile network operators (MNOs) around the world. This platform, capable of processing billions of messages per week, became the foundation for Thingstream, which now underpins the delivery of all u-blox services.

About u-blox

u-blox (SIX:UBXN) is a global provider of leading positioning and wireless communication technologies for the automotive, industrial, and consumer markets. Their solutions let people, vehicles, and machines determine their precise position and communicate wirelessly over cellular and short range networks. With a broad portfolio of chips, modules, and a growing ecosystem of product supporting data services, u-blox is uniquely positioned to empower its customers to develop innovative solutions for the Internet of Things, quickly and cost effectively. With headquarters in Thalwil, Switzerland, the company is globally present with offices in Europe, Asia, and the USA.

u-blox or third parties may hold intellectual property rights in the products, names, logos and designs included in this document. Copying, reproduction, modification or disclosure to third parties of this document or any part thereof is only permitted with the express written permission of u-blox.

The information contained herein is provided “as is” and u-blox assumes no liability for its use. No warranty, either express or implied, is given, including but not limited to, with respect to the accuracy, correctness, reliability and fitness for a particular purpose of the information. This document may be revised by u-blox at any time without notice. For the most recent documents, visit www.u-blox.com.

u-blox AG
Zuercherstrasse 68
8800 Thalwil
Switzerland

www.u-blox.com