Communication design considerations for M2M applications

Hans-Peter Ott, Stephan Joest
Ericsson Device and Application Verification Services (EDAVS)
40549 Düsseldorf
device-verification@ericsson.de, s.joest@ericsson.com

Lars Dürkop, Jürgen Jasperneite
inIT - Institute Industrial IT
Ostwestfalen-Lippe University of Applied Sciences
32657 Lemgo
{lars.duerkop, juergen.jasperneite}@hs-owl.de

Abstract: M2M communication is the foundation for the Internet of Things (IoT). In many cases IoT devices are connected to the Internet over cellular networks. But M2M applications and protocols are often not adapted to the specific properties of the radio link. So this paper will give hints and suggestions about the facts which should be considered carefully when design M2M applications and protocols.

1 Introduction

According to industry estimations there will be globally 50 billion connected devices by 2020. This projection builds on the proposition that anything that can benefit from being connected will be connected, and this is the foundation for the so-called Networked Society.

The Networked Society of the future will fundamentally change the way people innovate, collaborate, produce, govern and sustain. It will drive industrial and societal transformation towards new frontiers.

This development can be seen already today while mobile data traffic grows rapidly and exceeds by far traffic volumes generated by mobile telephony (see figure 1). Not only the rapidly increasing usage of smartphones and tablets is a driving force for this development. Another key component for the emerging Networked Society is Machine-to-machine (M2M). M2M refers to solutions allowing communication between sensors (which record temperature, pressure and humidity, among other things), assets (such as cars, smart meters, vending machines and consumer electronics) and information systems. The integration of sensors and information systems also allows automatic execution of processes based on data collection, data analysis and remote interaction.

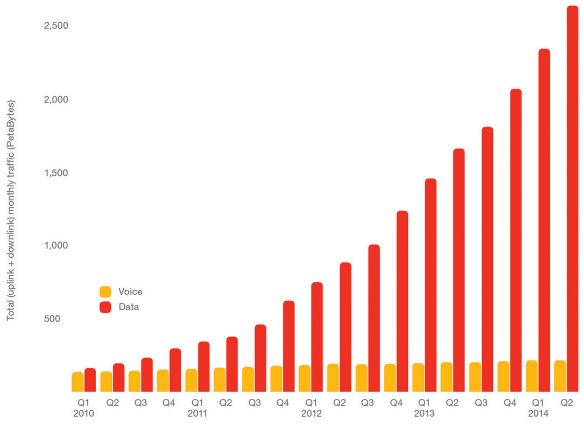

Figure 1: Rapid growth in mobile datavolume

2 M2M: Opportunities and challenges

Already now, M2M communications is one of the fastest growth areas in the ICT industry. Berg Insight, a dedicated M2M/Internet of Things (IoT) market research firm, has estimated that the compound annual growth rate of the number of cellular connected devices will be approximately 23 percent between 2013 and 2019, and that there will be almost 50 billion cellular connected devices at the end of that period (see figure 2). The firm also estimates that global cellular M2M network revenues will grow with a compound annual growth rate of around 27 percent and reach almost EUR 22.5 billion by 2019.

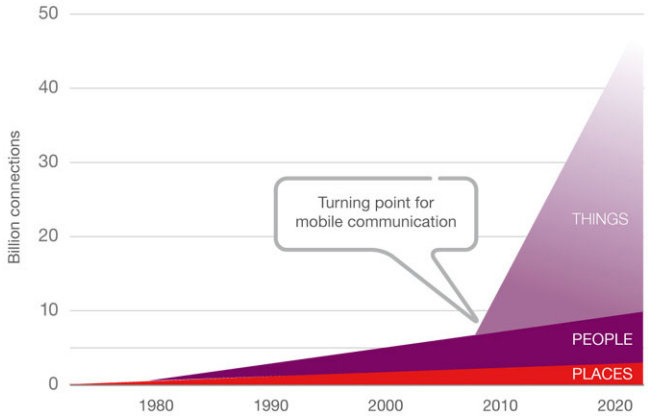

Figure 2: 50 billion connected devices are expected by 2020

The M2M application space is very wide. Improvements in traffic safety and traffic management are two examples. Transforming the electricity grid to a smart grid, driven by new requirements like energy efficiency, electrical vehicles, and consumer energy awareness are other examples. Manufacturing, agriculture, water management and environmental monitoring are other usage areas. To grasp these M2M opportunities it will become increasingly important to secure the delivery of high-quality flawless services as well as to achieve a positive network perception and user experience.

This challenge is demanding: The proliferation of M2M devices and applications will inevitably lead to a further increase in network complexity. The performance of M2M solutions can be affected by any mobile network and its characteristics like loss of coverage, lesser bandwidth and latency. On the other hand, improper M2M solutions can affect the performance of networks by unnecessary signal loading, poor ping-to-alive methodology. On both cases, this could create negative effects like congested networks, leading to unsatisfied customers, missed revenue, delaying mission critical processes and loosing opportunities etc.

What if a smart meter cannot be accessed remotely and the only solution is to send out field service to reset it? The cost for the company is then determined by the number of affected devices, time and cost for field service/hour, cost for transport/km, average distance to affected unit in km, trouble shooting cost and probably the loss of future business opportunities.

Other examples are the impact of applications - also those which are executed in M2M environments, for example Java embedded devices, etc. - onto mobile networks.

Analytics of M2M and M2M application behavior have shown that while developers have their own environment and services under control, and can manage the usage of mobile data, most often aspects such as the operational communications state of the device and the impact onto demand on the mobile network side is not understood.

These analytics can only be done when an inspection of the overall communication takes place, by accessing both radio and core network nodes during operation of the device. A number of essential key performance indicators (KPI) can be monitored and evaluated, such as measurements of energy consumption behavior, state changes of the mobile device in conjunction with the actual mobile data transfer, which then can be correlated to number and size of signaling messages in the mobile network and give valuable insights into the lifecycle and use cases.

Especially large-volume scenarios are of importance when it comes to mass deployment of solutions. ”Signaling storms” in mobile networks caused by a larger number of devices should or even must be avoided as they

do not only have an impact on the mobile network itself but also on the solution, e.g. when connection attempts fail, bandwidth cannot be allocated as requested, etc. - the overall experience can easily lead to a failure in execution and harm the brand reputation. Thus, "network friendly" solutions certainly have a competitive advantage in deployments. This underlines the importance of thorough verification and testing before the launch of new M2M devices and applications.

3 Managing the verification of M2M devices and applications

In order to avoid such severe problems, the M2M solution including devices, chipsets and applications needs not only a thorough verification on system level. Because executing a M2M test at enterprises and vendor setups, be it in a productive or in a research and development environment, one can simulate, trace and log of what happens in a device-to-device or device-to-backend communication, but one cannot observe what impact this communication does have on the operator network: how much payload is generated between the Radio Network and the Core Network, how many signaling messages are created for each packet data protocol establishment or drop, what happens on the network during idling times, how much does a broadcast effect the network signaling congestion etc. Only a verification procedure in a real network can give access to those parameters. That's why all of the leading operators today demand a qualified pre-test before launching new products and solutions on their networks.

This complete verification has crucial advantages. It will:

- Accelerate the process of launching M2M solutions.
- Secure expectations on functionality, stability and performance.
- Secure an efficient and effective testing process on international and operator specific network configurations with the complete M2M solution.
- Provide a cost efficient solution for player within the M2M business. The goal is secure quality using the fewest possible tests and resources to assure quality.
- Reduce harm to networks due to improper devices and applications.

These M2M Solution Verification Services are one central part of the so-called Device and Application Verification Service which Ericsson is performing for smartphone/tablet/chipset producers and for the M2M industry. Verification testing addresses different challenges which are preconditions for a successful commercial launch of M2M solutions, such as:

- End-User Experience
- Device Selection
- Device Behavior & KPIs
- Lifecycle Management of Apps & Devices
- International Use (Roaming)
- Cost

The technical verification is executed on different levels such as:

M2M Interoperability Testing The goal of the interoperability testing is to verify the interworking between the M2M solution and the radio access network and ensure interoperability and no harm to network. This test is focusing on radio and protocol layer and the signaling between the M2M device and the network.

M2M Application Testing The goal of the application testing is to identify and verify critical functionalities (e.g. ongoing transaction, time critical transmission, etc.) which are crucial for enterprise customers and operators. If the M2M application is implemented in a non-efficient way, then these factors e.g. reconnection to the network after loss of coverage, reduced bandwidth, aggressive behavior on network etc., might cause problems.

M2M Antenna Performance Testing The goal is to verify the M2M device's antenna performance regarding supported frequency bands and radio access technologies. Having poor antenna performance decreases the coverage area at the far side of the cell. Poor antenna performance may also drain the device battery quickly.

M2M Radio Performance Testing The goal is to verify the device radio performance in all applicable frequency bands. Poor radio frequency performance leads more signaling load towards the network and thus capacity limitation might occur in network. Point to note that the embedded mobile radio module in the M2M device has already been tested for radio performance according to 3GPP, but there is no guarantee that the performance has not been affected after integrating the module in the M2M device.

4 Influences of application and protocol design

As mentioned before the design of applications and protocols must be adapted to the specific characteristics of cellular networks. In this chapter design examples and their effects are shown.

4.1 Application design

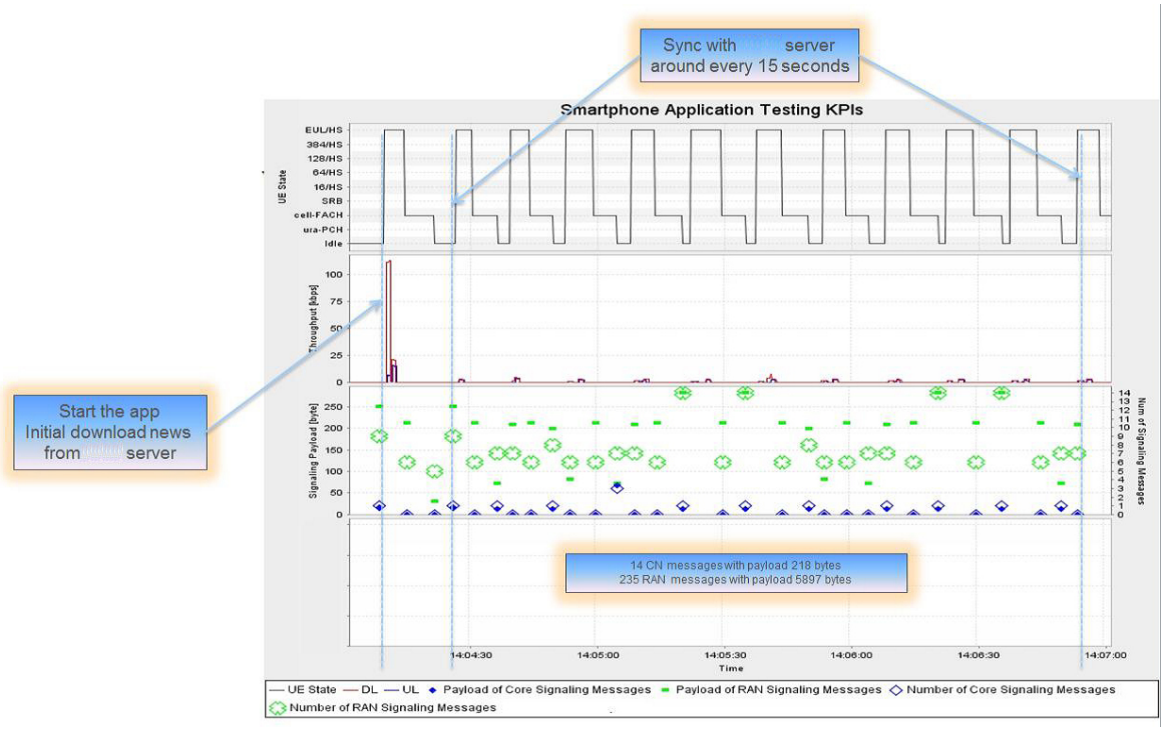

Figure 3: Example for non-optimized application behavior

Case 1 As an example of an interaction figure 3 shows the measuring results of a smartphone standard news reader app. This showcase demonstrates some areas of a typical behavior. The app shown here had been an early release, adapted to fit the needs, but not yet adapted for optimized behavior in mobile networks. The graphs show data consumption in the red-purple area: high initial data load at the beginning of the session, yet very little mobile data update throughout the lifetime of the app.

Yet while the app seemed to be idling, additional triggers were set in the device to create other operational modes (in this case, High Speed (HSPA) mode) – see the top graph component – which were absolutely unnecessary in the frequency they were triggered, compared to the ratio of content updates.

Higher so-called "UE States" (UE, User Equipment Device) do not only lead to higher battery consumption, but also to a request of resources both in the device and in the network which are allocated but not used. As a result, the battery drain is essential and can lead to a largely reduced operational availability in mobile environments.

Additionally, in the communication between radio and core network nodes, signaling payload and also number of signaling messages (visualized by the green graph elements) have a substantial –and certainly unnecessary– volume. This signaling also leads to a battery drain and performance degradation on both device and network side.

Meanwhile this app has been corrected in its behavior as the developers executed multiple series of test sessions in several releases of the app.

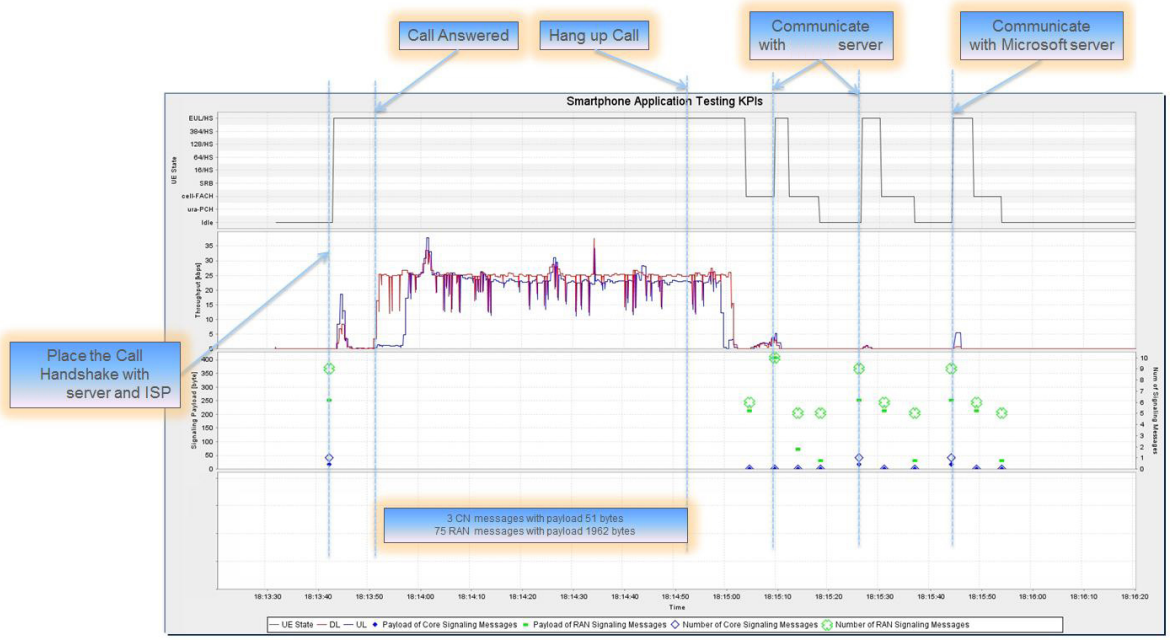

Figure 4: Example for optimized application behavior

Case 2 In figure 4 an optimized behavior of another app is shown: During the actual main operational cycle no signaling is created and thus the combination of device and application behaves as a ”friendly citizen in the network”. User experience is far better and also battery lifetime can be extended.

Additional parameters can be checked – especially on the M2M side, when typically user interaction is limited or not existing at all. Preventive or countermeasures should be applied at an early stage of solution development. Furthermore, in M2M deployment scenarios, the deployed volumes play a different role: a few hundred device and app combinations do not really play a big role and their impact can be neglected but large-scale rollout volumes can have a substantial impact. Imagine a scenario where a network outage is ongoing and several thousand or more devices roam simultaneously to the next available operator – this ”device aggression” behavior can cause severe disruptions on the mobile network and may lead to consequences which are not really wanted by the service or solution provider – up to a situation where these devices are being barred from connectivity by the Operations & Maintenance team of the respective operator. In those cases, a replacement of devices by a field force may lead to unpredicted cost scenarios which have not been calculated in the initial business case.

It should also made clear that each device behaves differently, and that also firmware updates in the device – while keeping the same app stack deployed – can lead to a different behavior in the mobile network. Several cases have been seen where long-term deployed hardware which received an OS update have suffered from optimization issues and thus lead to largely affected experience and use scenarios. Thus it is advised to always verify the combination of ”device and app” at once to make sure that the behavior is optimized.

From a developer’s perspective, several issues can be addressed beforehand. In deployment scenarios where the application design is based on runtime engines or other environments which are used as a stack, the provider of these classes should be asked whether their enabling software components are already verified against mobile networks. Another typical example is to prevent situations where no connectivity can be established at all, be it that the DNS entries are not correct and servers cannot be reached, be it that coverage or bandwidth is low or even missing, how many updates must be done per minute/hour/day etc., what is essential data, what is nice to have data and could be transmitted at a later stage?

4.2 Protocol design

Especially M2M applications often does not implement their own communication behavior, instead they use specialized protocols for data transmission. Therefore three typical M2M protocols have been analyzed in order to draw conclusions how the protocol design influences their performance. The test environment for the characterization of M2M protocols is shown in Figure 5. The key component is an Anritsu MD8475A cellular network emulator which enables measurements that are not affected by external influences like weather conditions or the cell's load situation.

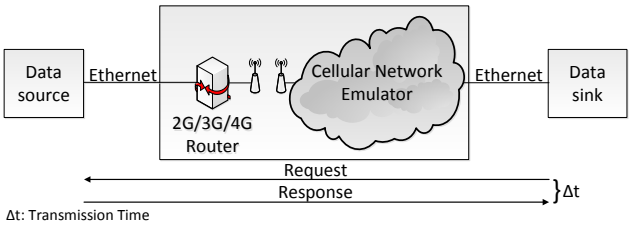

Figure 5: Test environment

In this test setup the transmission time of the M2M protocols Constrained Application Protocol (CoAP) [SHB13], Message Queuing Telemetry Transport (MQTT) [IE10] and OPC Unified Architecture (OPC UA) [Fou09] over LTE has been measured. The results are shown in Figure 6.

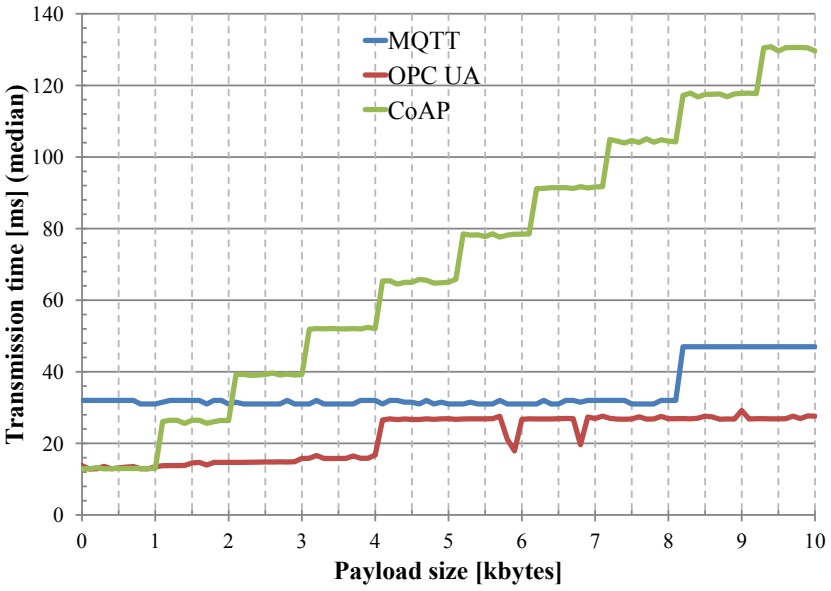

Figure 6: Transmission time of M2M protocols over LTE

It was expected that OPC UA had the largest transmission time because this protocol added significant overhead which resulted in the largest payload of all protocols. Instead it was observed that OPC UA had the smallest transmission time. This is due to the fact that the TCP-based OPC UA sends several successive frames over the radio channel without waiting for the corresponding acknowledgments because of the TCP receive window functionality. As a result, the LTE radio link control layer is able to merge all frames so that they can be transmitted simultaneously in one LTE resource block. In contrast, the UDP-based CoAP expects an acknowledgment for each frame before it continues its transmission. So each packet allocates a new resource block. It must be stated that protocols using UDP and an own message acknowledgment mechanism without windowing are poorly suited for large data transmission (i.e. streaming) over cellular networks.

Furthermore, it has been observed that the used MQTT implementation establishes a dedicated TCP connection for each data packet resulting in a worse performance than OPC UA. This behavior has been corrected, the

improved protocol performance is shown in figure 7.

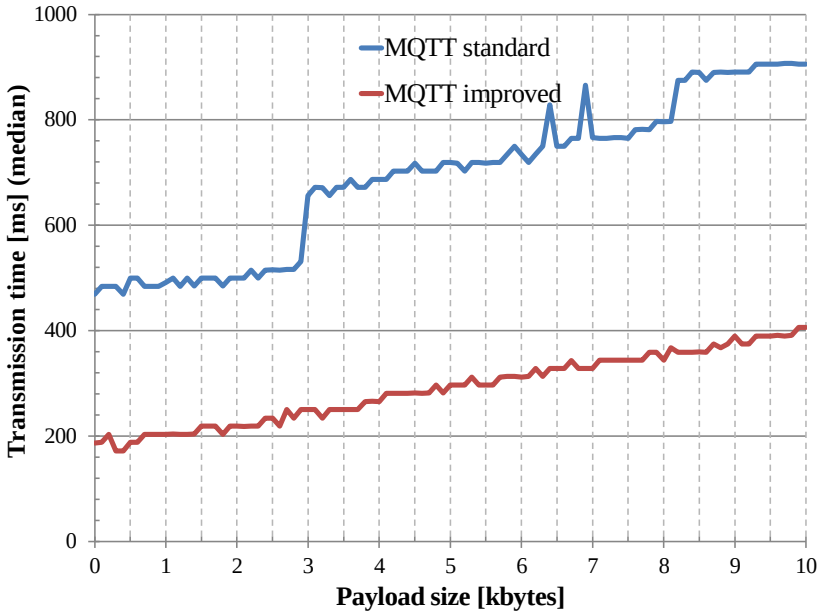

Figure 7: Improved MQTT performance

5 Further reading

The GSM Association, a worldwide federation of cellular network operators, has recently published its IoT Device Connection Efficiency Guidelines [Ass14] which are aimed at M2M service providers, device makers and device application developers. The document contains a comprehensive list of concrete recommendations for the design of M2M device firmware and software.

Acknowledgment

The presented work is part of IGF project 17715 BG ”M2M@Work” which is funded by the German Association for Research ’Forschungsvereinigung beim ZVEI e.V. – FE, Lyoner Str. 9, 60528 Frankfurt/Main’ in the AiF under the program for the support of industrial research and development (IGF) by the Federal Ministry of Economics and Technology based on a decision of the German Bundestag.

References

[Ass14] GSM Association. IoT Device Connection Efficiency Guidelines. http://www.gsma.com/connectedliving/gsma-iot-device-connection-efficiency-guidelines/, October 2014.

[Fou09] OPC Foundation. OPC Unified Architecture. https://opcfoundation.org/about/opc-technologies/opc-ua/, 2009.

[IE10] IBM and Eurotech. MQ Telemetry Transport (MQTT) V3.1 Protocol Specification. http://www.ibm.com/developerworks/webservices/library/ws-mqtt/ws-mqtt-pdf.pdf, August 2010.

[SHB13] Z. Shelby, K. Hartke, and C. Bormann. Constrained Application Protocol (CoAP). https://tools.ietf.org/html/draft-ietf-core-coap-18, June 2013.