Performance Evaluation of M2M Protocols Over Cellular Networks in a Lab Environment

Lars Dürkop*, Björn Czybik*, Jürgen Jasperneite*†
*inIT - Institute Industrial IT, Ostwestfalen-Lippe University of Applied Sciences, D-32657 Lemgo, Germany
Email: {lars.duerkop, bjoern.czybik, juergen.jasperneite}@hs-owl.de
†Fraunhofer IOSB-INA Application Center Industrial Automation, D-32657 Lemgo, Germany
Email: juergen.jasperneite@iosb-ina.fraunhofer.de

***Abstract*—According to the vision of the Internet of Things the seamless and flexible networking of everyday objects will become an important field of application for Internet-based communication. The simple integration of these devices into a communication system often requires wireless technologies, especially when there is no wired infrastructure available. Cellular networks of the third and fourth generation are promising enablers for embedding a variety of different devices into the Internet of Things. However, cellular networks use a completely different approach for data transmission and media access than wired networks like Ethernet. Therefore, it is necessary to investigate the transmission behavior of common protocols for machine-to-machine (M2M) communication with respect to the peculiarities of cellular networks. In this paper, three M2M protocols – CoAP, MQTT and OPC UA – are compared to each other with regard to their transport mechanisms to evaluate the transmission times and analyzing potentials for optimization. For the evaluation a laboratory test environment with cellular network emulators for EDGE, UMTS and LTE is used to analyze the protocols without interference of delays caused by the Internet or by other users allocating resources of the cellular network.**

I. Introduction

The term Internet of Things (IoT) describes the increasing cross-linking of smart devices like sensors and actuators referred to as things. Exemplary IoT scenarios can be found in the fields smart home, e-health and smart grid. The latter describes a grid emerging from the convergence of information and communication technologies (ICT) with existing electrical distribution systems. It will enable intelligent applications like smart metering, realtime pricing or an improved network management for enhanced power quality and optimal distribution of power [1].

By today, the Internet is used mainly in a human-centric manner. In contrast, the IoT is characterized by communication between machines, the so called machine-to-machine (M2M) communication. The interconnected things are thereby often located in relatively isolated areas without access to the wired ICT. Hence, their connection to the Internet is often realized over cellular networks. The main disadvantage of such networks is their unreliability, which is reflected in high packet loss rates. Therefore, in the context of M2M several protocols have been developed specialized on lossy channels. The specific characteristics of these protocols can simplify the design and the operation of M2M applications because a large portion of error handling can be done by the protocols.

This paper will present the OPC Unified Architecture (OPC UA), the Constrained Application Protocol (CoAP) and the Message Queuing Telemetry Transport (MQTT) protocol as representative M2M protocols. Since the M2M endpoints are often very resource constrained devices, the main criteria for this selection was the presence of lightweight implementations. The focus of this paper is on the evaluation of the mentioned protocols over cellular networks for M2M applications with the need for reliable real-time data exchange. An exemplary use case from the Smart Grid domain addresses the demand response management via real-time electricity price control [2]. Another prospective application is the realization of industrial control systems via M2M protocols. By today, the application of M2M communication is limited to maintenance and monitoring. This is due to the high temporal constraints of the underlying technical process. To control such processes remotely, a periodic data exchange with reliable timing behavior is necessary.

The performance of the underlying data-transfer protocols is very important for the design of such applications. Therefore the time necessary for data transfers from a data source (i. e. a smart meter) to the data sink (i. e. the grid operator's control central) is used as Key Performance Indicator (KPI) in this paper. The measurements have been performed in a lab test environment under ideal conditions with no packet loss. By using this method the protocol behavior for cyclic data transfers can be analyzed in detail. In future investigations the protocols must also be evaluated under the presence of packet loss.

This paper is structured as follows: After a summary of related work in section II, the analyzed M2M protocols are described in section III. The results of the performance evaluation are presented in section IV. The paper ends with a conclusion and an outlook to further work in section V.

II. Related work

A characterization of three cellular network standards from the second, the third and the forth generation was performed in [3]. In that paper the round-trip time of UDP packets over cellular networks was measured over a couple of days. It was shown, that the performance is dependent on the time of the day.

A comparison of MQTT and CoAP for smartphones was conducted in [4] for different application scenarios wherein CoAP is showing better results with respect to bandwidth usage and round trip time (RTT). A further comparison of MQTT and CoAP was discussed in [5] and in [6] for wireless sensor

978-1-4799-1866-9/15/$31.00 ©2015 IEEE

networks. The evaluation in [5] showed a better performance of CoAP with respect to bandwidth usage. Furthermore, MQTT was compared with HTTP in [7]. An evaluation of CoAP over Short Message Service using GPRS was discussed in [8], showing that the response time is linearly increasing with respect to the payload size. Further performance evaluations over cellular network (GPRS and UMTS) for different application layer protocols such as HTTP, FTP, POP3 and SMTP were discussed in [9]. The evaluation showed that the protocols need to be optimized for the use in cellular networks because the protocols have much overhead which occupies a huge amount of the available network resources. Furthermore, the retransmission mechanism in TCP occupies further network resource since TCP was designed for congestion in network and not for lossy channels.

In [10] a simulation study for smart meters sending their data using UDP and TCP via UMTS was conducted with respect to different smart meter scenarios. The evaluation outlined that UDP shows better performance than TCP wherein the delay of packets substantially increases with the number of smart meters.

III. Protocols for M2M applications

This section will shortly describe different protocol classes used for M2M communication. From each class a representative protocol is chosen for the performance evaluation in section IV.

A. Protocol classes

There are three major groups of protocols used for M2M communication:

Service-oriented architectures (SOA) are used in industrial automation systems to exchange soft real-time data, for instance between programmable logic controllers and Supervisory, Control and Data Acquisition (SCADA) systems. The Device Profile for Web Services [11] was developed as an SOA implementation specialized in embedded systems. However, as discussed in [12], DPWS is distinguished by a noticeable protocol overhead and a large memory requirement. In contrast, the authors showed that the OPC Unified Architecture (OPC UA) [13] can be scaled down so that it can be implemented on very resource constrained devices. Therefore, OPC UA is used as SOA protocol for the performance evaluation.

Protocols following the **Representational State Transfer (REST)** [14] architecture style which defines constraints to the used components, connectors and data elements. Since the Internet is based on the REST style, an easy integration of sensors and actuators using such protocols into the existing Internet infrastructure is possible. In this work the Constrained Application Protocol (CoAP) [15] is used as an example for REST-based protocols since it is designed especially for constrained networks and environments such as sensor networks.

The design of **message oriented protocols** supports the asynchronous data transfer between distributed systems. Therefore these protocols often use specialized message transfer agents which can buffer messages on their transmission path. The agents act as intermediary between sender and receiver. The Message Queuing Telemetry Transport (MQTT) [16] is a lightweight representative of this protocol class.

The three selected protocols will be described in the following paragraphs.

B. OPC Unified Automation

OPC UA is a platform-independent industrial middleware technology. It is designed to allow interoperability between heterogeneous system components over various types of networks. OPC UA defines methods for both data modeling and transport, whereas the focus of this paper is on the latter. As shown in Fig. 1, there are three variants for transmitting data over the OPC UA communication stack.

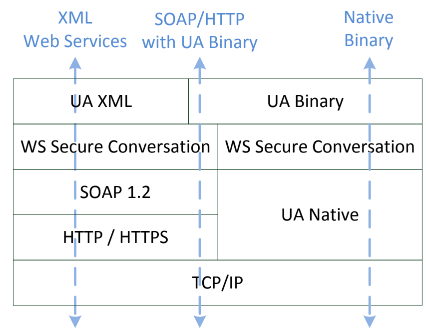

Fig. 1: OPC UA communication stack [17]

OPC UA is based on the server/client communication pattern. Here, a client requests data and a server sends a response containing the data. For transmission OPC UA currently defines two protocol mappings and two encodings. The data can be encoded generically in UA XML or more efficiently in UA Binary. For transport OPC UA can use common web standards like SOAP and HTTP which allow an easy crossing of firewalls. For resource constrained devices the payload can be directly integrated into TCP by the UA Native protocol. It is also possible to transport UA Binary by SOAP and HTTP.

Nevertheless, even the combination of UA Binary and UA Native causes a significant overhead. Each *ReadResponse* message, which transports the OPC UA payload from the server to the client, can contain the values of up to 10 variables. As analyzed in [18], the message overhead is at least 60 bytes. Each variable results in a further overhead of 18 bytes.

C. Constrained Application Protocol

CoAP is an active Internet draft of the Internet Enginnering Task Force (IETF), developed by the Working Group for Constrained RESTful Environments (CoRE) [19]. The aim is to develop a lightweight protocol for sensor and actuator integration into the existing Internet architecture. Since the Internet architecture follows the REST architectural style, CoAP follows this arhitecture style as well. One of the main used protocols in the Internet is the HTTP protocol using the methods GET, PUT, POST and DELETE for resource manipulation. CoAP uses the same method semantics as HTTP for resource manipulation to enable the aforementioned easy integration. However, HTTP uses TCP as transport protocol with a huge protocol overhead and connection management to enable a reliable transport. Instead of TCP, CoAP uses UDP as transport protocol to have less overhead and interaction between endpoints because of short-living connections of sensors (which might be in a sleep mode most of the time). Furthermore, CoAP has an optimized 4 byte fixed protocol header, a token field with a maximum of 8 bytes and an option field with variable length. Since CoAP uses the unreliable transport mechanism of UDP it has an own reliable transport service if messages need to be confirmed. For the purpose of firmware updates CoAP defines a blockwise transfer [20] with a maximum block size of 1024 bytes where each block must be acknowledged. Furthermore, a CoAP client can discover a CoAP server and subscribe to resources of the CoAP server via a discovery protocol using a discovery repository specified in [21]. For the publication of resources to the subscribed CoAP clients, UDP multicast communication is used. (Using TCP as transport protocol for group communication via multicast is not possible because for each subscriber a communication channel needs to be established.) A further advantage of UDP is the asynchronous communication between endpoints.

D. Message Queuing Telemetry Transport

MQTT has been designed for devices with limited processing power and memory capabilities. It is based on the publish/subscribe communication pattern. Two MQTT devices do not interact directly with each other, but via a so-called broker. The design of a MQTT system is shown in Fig. 2.

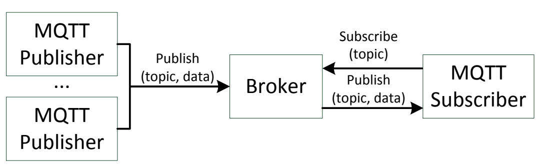

Fig. 2: MQTT communication

The publisher (client) publishes messages to the broker and assign a topic to each message. A subscriber (server) subscribes to a specific topic. When the broker receives a publishing, it forwards the message to the servers subscribed to the topic of the message. A typical MQTT system consists of many distributed small devices like sensors functioning as publishers. Their data is aggregated by a central broker and consumed by a subscriber.

Like OPC UA MQTT is based on TCP, but in contrast its overhead is considerably smaller: the MQTT protocol header comprises only two bytes and no additional transport protocols like SOAP or HTTP are used. As mentioned in section II, the error handling of TCP is not well suited for cellular networks. Therefore MQTT defines own methods for message reliability categorized in three Quality of Service (QoS) classes. In QoS class 0, the connection reliability only depends on TCP – no additional methods for quality checking are defined. In QoS class 1 each message is retransmitted until it is acknowledged by the receiver. As a result, it is possible that messages are received multiple times. This is avoided in QoS class 2, which ensures the unique transmission of each message. In [22] the handshake mechanisms used in each QoS class and their impact on the connection quality are described in detail.

IV. Performance Evaluation

In this section the performance evaluation of CoAP, MQTT and OPC UA is discussed. First of all the measurement setup is explained and afterwards the obtained results are presented.

A. Test setup

The core component of the test setup for the performance evaluation of CoAP, MQTT and OPC UA over cellular networks is an Anritsu MD8475A emulator for GSM (2G), UMTS (3G) and LTE (4G). In this lab testing environment reproducible measurements can be performed. Influences occurring in real cellular networks due to varying receive/transmit conditions, cell utilization, path loss or interferences can be excluded.

The test setup is used to measure the transmission time between a data source and the corresponding data sink in relation to the length of the payload. The latter varies between 0 and 10,000 bytes with a step size of 100 bytes. The measurements are repeated periodically 100 times per payload length. In the sense of M2M communication the data sources represent distributed devices like smart meters, and the data sinks act as data integration points like control applications. For the measurements the data source of the examined protocol is connected via a mobile router to the radio interface of the cellular network emulator. The data sink is connected to the Ethernet interface of the emulator. The setup for each protocol is shown in Fig. 3 and Fig. 4.

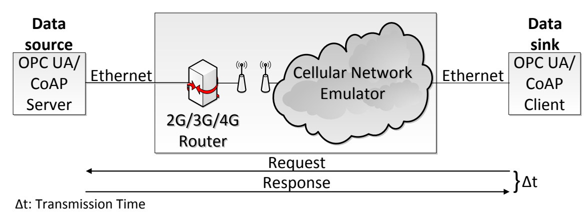

Fig. 3: Test setup for OPC UA and CoAP

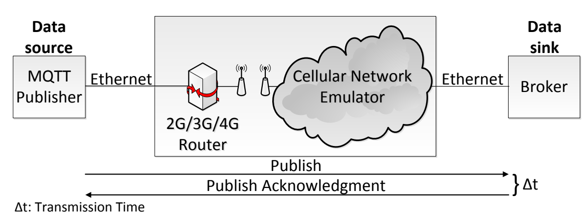

Fig. 4: Test setup for MQTT

In table I the most important emulator parameters for the three different cellular network standards are given.

B. Analysis

The evaluation results for all protocols are shown in Fig. 9, Fig. 10 and Fig. 11 [1]. In the following, some noticeable issues are mentioned.

[1]In this paper the definition 1 kbyte = 1000 bytes is used.

TABLE I: Cellular network emulator settings

EDGE		**UMTS**	
DL Slots	3	Duplex Mode	FDD
UL Slots	1	DL Packet Window Size	512 Bytes
Coding Scheme	MCS-9	UL Packet Windows Size	256 Bytes
		DL Packet Rate	384 kBytes/s
		UL Packet Rate	64 kByte/s
LTE			
Duplex Mode	FDD		
DL Bandwith	5 MHz	DL: Downlink	
UL Bandwith	5 MHz	UL: Uplink	
Scheduling Mode	Static	FDD: Frequency Division Multiplex	

– **OPC UA has the lowest transmission time.** This fact is worth mentioning because OPC UA has the largest overhead of all measured protocols. To explain this matter, the data exchange of the protocols must be analyzed. The frames sent in OPC UA for one data exchange are shown in Fig. 5. The OPC UA connection establishment occurs only once at the beginning of the measurement and is not shown. Note, that there is only one pure TCP frame. The TCP acknowledgment for the second ReadResponse is included in the ReadRequest frame of the following data transfer. For an OPC UA payload of 2500 bytes, the total amount of 2959 bytes is sent (including IP layer and MAC addresses). The transmission time Δt_1 over EDGE is about 840 ms.

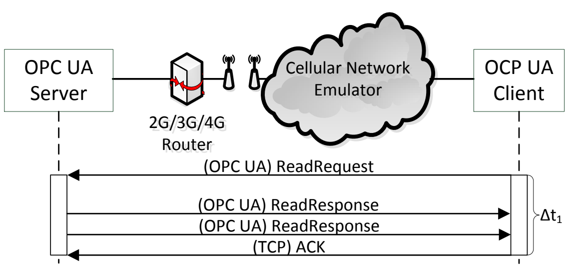

Fig. 5: OPC UA data exchange (Payload 2500 bytes)

For comparison, Fig. 6 shows the frames sent in MQTT for one data exchange (QoS class 0). In contrast to OPC UA, for each data exchange a dedicated TCP connection is established and released. In addition, the publisher establishes an own connection to the broker on the MQTT layer by the Connect and Connect Acknowledge (Connack) frames. The bytes sent and the time needed for each section of the data transfer (MQTT payload 2500 bytes over EDGE) are shown in table II.

TABLE II: Analysis of MQTT data exchange

Section	Bytes	Time [ms]	Remark
Δt_1	186	305	TCP Connection Establishment
Δt_2	150	278	MQTT Connection Establishment
Δt_3	2729	517	MQTT Payload
Δt_4	116	105	MQTT Connection Termination
Δt_5	114	107	TCP Connection Termination
Sum	**3195**	**1312**	

The transmission time of CoAP and OPC UA messages are similar up to 1024 bytes because both protocols need the same number of frames for transmitting their payload. After 1024 bytes, CoAP shows worse performance in comparison

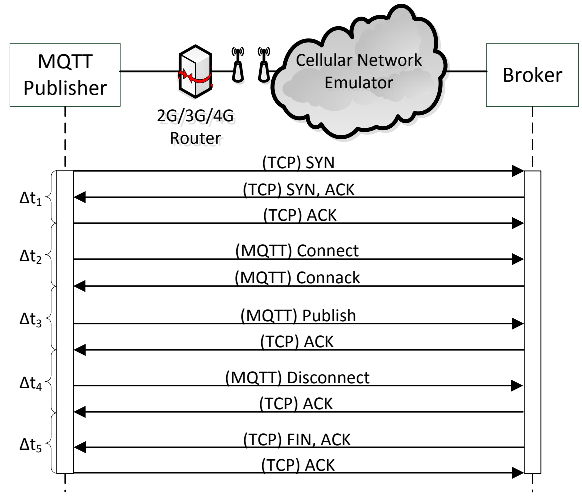

Fig. 6: MQTT data exchange (Payload 2500 bytes)

to OPC UA in all measurements since each packet needs to be acknowledged in CoAP whereas in OPC UA no explicitly acknowledgment is necessary.

The stepwise increase of the transmission time in CoAP is reasonable because of the blockwise transfer of payloads greater than 1024 bytes. Each block must be acknowledged and the next block is only sent if the client has requested the next block as depicted in Fig. 7. For the transmission of a 2500 bytes payload the transmitted data sizes and the transmission time is given in table III.

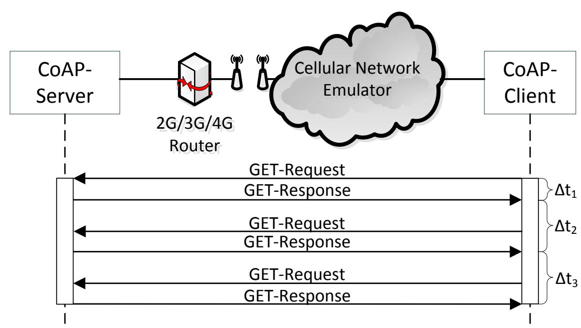

Fig. 7: CoAP data exchange (Payload 2500 bytes)

TABLE III: Analysis of CoAP data exchange

Section	Bytes	Time [ms]	Remark
Δt_1	1137	623	CoAP Payload 1/3
Δt_2	1137	618	CoAP Payload 2/3
Δt_3	565	534	CoAP Payload 3/3
Sum	**2839**	**1775**	

– **There are periodic spikes in the transmission time.** This phenomenon occurs in the measurements of the TCP-based protocols OPC UA and MQTT over EDGE and UMTS. LTE is not affected. To illustrate this point, an excerpt of the OPC UA over EDGE test result is shown in Fig. 8. At specific payload lengths the spikes occur in very short intervals. This leads to the increased median of the transmission time visible in Fig. 9 at payload lengths 3900 bytes (OPC UA), 8500 bytes (OPC UA) and 6800 bytes (MQTT). The reason of the delays probably lies in the interaction of TCP and the cellular network protocols. A detailed analysis is part of future work.

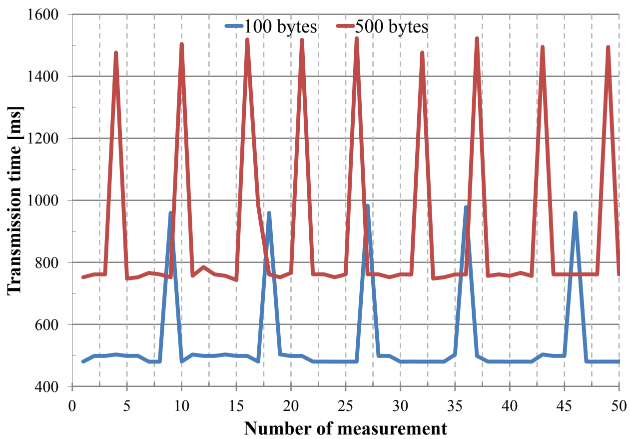

Fig. 8: Transmission time for different payload sizes (OPC UA over EDGE)

– **In contrast to CoAP, the transmission times of MQTT and OPC UA over LTE remain largely constant (see Fig. 11).** In LTE the data transmitted over the air interface is divided into so-called transport blocks. The length of the blocks is determined by the LTE base station (respectively the cellular network emulator in the used test setup). If the length of one transport block is larger than the length of one IP packet sent by the mobile device, LTE concatenates the IP packets until the transport block size is reached. After the concatenation the data is transmitted over the air interface.

This process leads to the observed constant transmission time. As the transport block size determined by the emulator is fixed, several TCP frames of the MQTT publisher/OPC UA server are concatenated and are sent in the same transport block. Unused transport block capacity is wasted. This transmission behavior is possible since in TCP successive frames can be sent without waiting for a TCP acknowledgment (depending on the TCP window size). When the transport block capacity is fully utilized, the data must be transmitted in a subsequent transport block. This leads to the increases of transmission time of OPC UA at 4000 bytes and MQTT at 8100 bytes.

In contrast to the TCP-based protocols MQTT and OPC UA, CoAP uses UDP as transport protocol. As mentioned, CoAP segments large payloads into blocks of 1024 byte. Each block must be acknowledged by the recipient. Since CoAP does not support receive windows as in TCP, the blocks cannot be concatenated by LTE and each block must be sent separately over the air interface. Therefore the transmission time of CoAP rises every 1024 bytes – over LTE as well as over EDGE and UMTS.

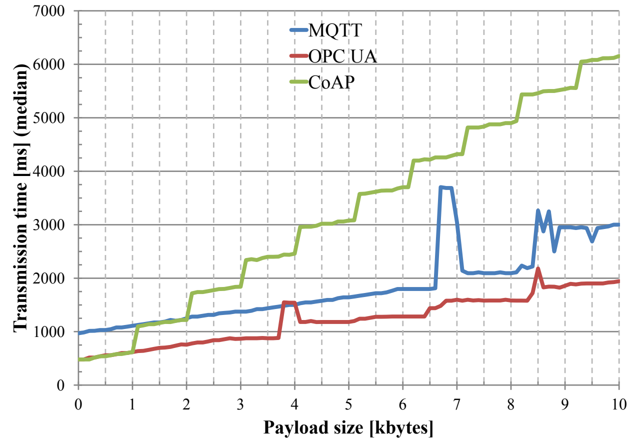

Fig. 9: EDGE measurement results

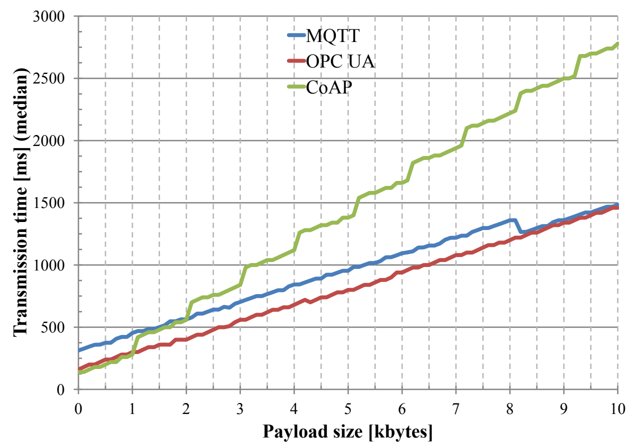

Fig. 10: UMTS measurement results

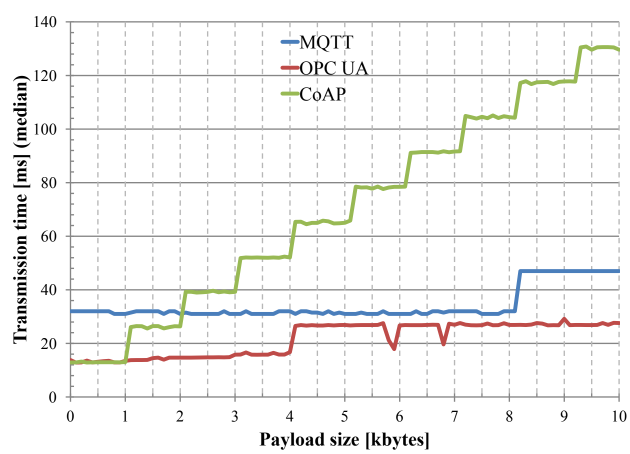

Fig. 11: LTE measurement results

V. Conclusion

This paper has evaluated three prospective protocols for realizing future real-time smart grid applications. The focus was on measurements of the transmission time for cyclic data exchange over the cellular network standards EDGE, UMTS and LTE in a laboratory environment. It has been shown that OPC UA achieved the best test results – although OPC UA has the largest protocol overhead of all evaluated candidates. This is due to the fact that OPC UA has the most suitable protocol design for cyclic data transfer. Especially in the case of LTE the transmission time depends not only on the total amount of data, but also on the exact sequencing of data transfer. This has been clearly observed in the evaluation of CoAP. Its implementation of reliable data exchange is not suitable for the transmission of large payloads over cellular networks. Protocols based on TCP achieve a better performance due to TCP-features like windowing.

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

[1] T. Khalifa, K. Naik, M. Alsabaan, and A. Nayak, “A transport control protocol suite for smart metering infrastructure,” in *International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, Apr 2011.

[2] L. P. Qian, Y. Zhang, J. Huang, and Y. Wu, “Demand response management via real-time electricity price control in smart grids,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1268 – 1280, 2013.

[3] G. Man Shrestha and J. Jasperneite, “Performance evaluation of cellular communication systems for m2m communication in smart grid applications,” in *Computer Networks*, A. Kwiecień, P. Gaj, and P. Stera, Eds. Springer Berlin Heidelberg, 2012, pp. 352–359.

[4] N. De Caro, W. Colitti, K. Steenhaut, G. Mangino, and G. Reali, “Comparison of two lightweight protocols for smartphone-based sensing,” in *IEEE 20th Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Nov 2013, pp. 1–6.

[5] S. Bandyopadhyay and A. Bhattacharyya, “Lightweight internet protocols for web enablement of sensors using constrained gateway devices,” in *Computing, Networking and Communications (ICNC), 2013 International Conference on*, Jan 2013, pp. 334–340.

[6] E. Davis, A. Calveras, and I. Demirkol, “Improving packet delivery performance of publish/subscribe protocols in wireless sensor networks,” 2013, pp. 648–680.

[7] V. Lampkin, W. Leong, L. Olivera, S. Rawat, N. Subrahmanyam, and R. Xiang, “Building smarter planet solutions with mqtt and ibm websphere mq telemetry,” 2012.

[8] N. Gligoric, T. Dimcic, D. Drajic, S. Krco, I. Dejanovic, N. Chu, and A. Obradovic, “Coap over sms: Performance evaluation for machine to machine communication,” in *Telecommunications Forum (TELFOR), 2012 20th*, Nov 2012, pp. 1–4.

[9] P. Reichl, N. Jordan, J. Fabini, R. Lauster, M. Umlauft, W. Jäger, T. Ziegler, P. Eichinger, G. Pospischil, and W. Wiedermann, “Wireless inter-system quality-of-service: A practical network performance analysis of 3g and beyond,” in *Kommunikation in Verteilten Systemen (KiVS)*, ser. Informatik aktuell, P. Müller, R. Gotzhein, and J. B. Schmitt, Eds. Springer Berlin Heidelberg, 2005, pp. 230–241.

[10] S. Abdul Salam, S. Mahmud, G. M. Khan, and H. Al-Raweshidy, “M2m communication in smart grids: Implementation scenarios and performance analysis,” in *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, April 2012, pp. 142–147.

[11] “Devices profile for web services version 1.1,” Organization for the Advancement of Structured Information Standards, July 2009. [Online]. Available: http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.html

[12] L. Dürkop, J. Imtiaz, H. Trsek, and J. Jasperneite, “Service oriented architecture for the auto-configuration of real-time ethernet systems,” in *3. Jahreskolloquium ”Kommunikation in der Automation” (KommA 2012)*, Lemgo, Nov 2012.

[13] “Opc unified architecture,” OPC Foundation, 2009.

[14] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, University of California, Irvine, 2000.

[15] Z. Shelby, K. Hartke, and C. Bormann, “Constrained application protocol (coap) - draft-ietf-core-coap-18,” Internet Engineering Task Force, June 2013. [Online]. Available: https://tools.ietf.org/html/draft-ietf-core-coap-18

[16] IBM and Eurotech, “Mq telemetry transport (mqtt) v3.1 protocol specification,” August 2010. [Online]. Available: http://www.ibm.com/developerworks/webservices/library/ws-mqtt/ws-mqtt-pdf.pdf

[17] G. Cândido, F. Jammes, J. Barata de Oliveira, and A. W. Colombo, “Soa at device level in the industrial domain: Assessment of opc ua and dpws specifications,” in *8th IEEE International Conference on Industrial Informatics (INDIN)*, Jul 2010.

[18] G. Man Shrestha, J. Imtiaz, and J. Jasperneite, “An optimized opc ua transport profile to bringing bluetooth low energy device into ip networks,” in *IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, Sep. 2013.

[19] C. Bormann and A. McGregor, “Constrained restful environments (core),” Internet Engineering Task Force. [Online]. Available: https://tools.ietf.org/html/draft-ietf-core-coap-18

[20] Z. Shelby and C. Bormann, “Blockwise transfers in coap - draft-ietf-core-block-08,” Internet Engineering Task Force, February 2012. [Online]. Available: http://tools.ietf.org/html/draft-ietf-core-block-08

[21] Z. Shelby, C. Bormann, and S. Krco, “Core resource directory - draft-ietf-core-resource-directory-01,” Internet Engineering Task Force, December 2013. [Online]. Available: http://tools.ietf.org/html/draft-ietf-core-resource-directory-01

[22] S. Lee, H. Kim, D.-k. Hong, and H. Ju, “Correlation analysis of mqtt loss and delay according to qos level,” in *International Conference on Information Networking (ICOIN)*, Jan 2013.