An evaluation of the applicability of OPC UA Publish Subscribe on factory automation use cases

Andreas Eckhardt
Engineering Automation Systems
Bosch Rexroth AG
Erbach, Germany
Andreas.Eckhardt@boschrexroth.de

Sebastian Müller
Engineering Automation Systems
Bosch Rexroth AG
Erbach, Germany
Sebastian.Mueller8@boschrexroth.de

Ludwig Leurs
Engineering Automation Systems
Bosch Rexroth AG
Lohr am Main, Germany
Ludwig.Leurs@boschrexroth.de

***Abstract*—In today's factory automation the exchange of data between systems becomes more and more important. The goal is to use one communication technology through all levels of a factory automation. Now the challenge is to find a communication technology that fulfills all the requirements on the different levels. The Server/Client architecture used in OPC UA became the de facto standard for several use cases in the upper levels, but in the lower levels e.g. Control-to-Control and Field level, OPC UA wasn't performant enough to replace today's fieldbus technologies. Therefore the OPC Foundation extended OPC UA by a Publish Subscribe architecture which was released in March 2018 in the OPC UA Specification Part 14. This paper gives an overview of this specification and an evaluation of the applicability of OPC UA communication technologies in combination with Time-Sensitive Networking (TSN) on factory automation use cases.**

Keywords—OPC UA Publish Subscribe; Factory Automation; Communication; Time-Sensitive Networking

I. Introduction

In industrial systems the requirement to exchange data between systems becomes more and more important. Nowadays industrial automation and control systems have several different ways to communicate. A common communication technology used in the upper layers of the automation pyramid is OPC UA. OPC UA is one of the most promising communication protocols in context of Industry 4.0 [1]. Until now OPC UA uses connection-oriented communication based on Service-oriented Architecture (SOA) to exchange data between control systems and the enterprise levels of the automation.

On the field level Ethernet-based fieldbus protocols (e.g. SERCOS, ProfiNET, CAN) are used to achieve the required performance. Especially combining those protocols leads to challenges regarding interoperability. To solve this problem, it would be convenient to have vendor-independent communication solutions through the whole automation pyramid. Therefore the OPC Foundation works on a new specification which is using connectionless mechanisms to exchange data. The specification was released in March 2018 and extends the portfolio of OPC UA with a new communication mechanism called OPC UA Publish Subscribe (OPC UA PubSub) [2].

II. Related Works and Scope of This Paper

When it comes to the OPC UA PubSub specification, there is little to none research published yet. Previous works have only focused on researching the OPC UA Server/Client protocols [3]. However there are papers that cover Publish and Subscribe mechanisms with other middleware approaches like MQTT or AMQP [4, 5].

This paper is organized as follows. In section III the communication used by a factory automation control is being analyzed. The goal is to identify use cases and their communication requirements. In section IV an overview of the content of the OPC UA PubSub specification is given. In section V four different communication options are defined to evaluate and compare their applicability on the previously defined use cases. Section VI concludes the paper and outlines future work.

III. Use Cases of an Industrial Control and Its Requirements

One requirement of Industry 4.0 factories is the vertical and horizontal communication between systems [6]. From the point of view of an industrial control, there are at least four different ways to communicate. Fig. 1 gives an overview of those four ways.

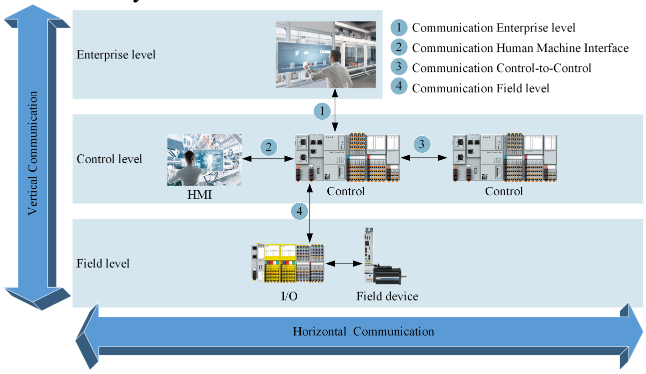

Fig. 1 Overview of the vertical and horizontal communication.

978-1-5386-7108-5/18/$31.00 ©2018 IEEE

A. Communication between controls and enterprise level

In the upper layers of the automation pyramid OPC UA is already a well-established standard. Enterprise applications track the process chain from raw goods to the finished product. They use acyclic communication with a cycle time between seconds and hours. To analyze data, it is very important to have consistent data. Therefore it's more important for enterprise application to have data integrity. The entities can be decoupled in space and also in time.

B. Communication between controls and HMIs

The main use case is to display the state of a machine on an HMI. The communication between the control and HMI is based on acyclic communication with a cycle time typically around 100ms. The entities can be decoupled in space.

C. Communication between controls

The requirements of C2C communication depend a lot on the use case. For this paper when talking about C2C communication, cyclic communication with a latency of maximum 1ms is meant. The machines can be decoupled in space.

D. Communication between controls and field level

The communication between controls and field devices is based on cyclic communication. With a latency smaller than 1ms and a time synchronization accuracy smaller than 1 µs.

E. Requirements

TABLE I summarizes the requirements of the different use cases. *Latency/Cycle Time* is the time that passes between sending and receiving a message. *Time Sync Accuracy* is the timing error on a packet flow between two systems. *Decoupled Communication* describes the requirement to decouple the communication entities in space, time and/or synchronization. If the entities are decoupled in space than they have no connection between them established. Decoupling in time means that the entities don't need to be online at the same time. Consequently the broker can store the data and the subscriber can subscribe to the data even after the publisher went offline. When decoupling in synchronization e.g. a publisher application will not be interrupted for processing publisher data. Subscribers are getting asynchronously notified while working on a concurrent task [7]. *Subscriber* shows the number of subscribers. *Hardware* describes which hardware is usually used in this use case. The requirement *Configuration* indicates if there is only the need to configure the application and/or TSN offline and/or online. *Quality of Service (QoS)* describes whether best effort traffic or scheduled traffic is needed and whether the transmission needs to be reliable in terms of retransmitting lost packages, ordering of packages and error-checking mechanisms.

TABLE I. Use case requirements [8]

Requirements	**Enterprise**	**HMI**	**C2C**	**Field level**
Latency / Cycle Time	1 s – 1 h	100 ms	1 ms	250 µs – 1 ms
Time Sync Accuracy	-	-	<1 ms	<1 µs
Decoupled Communication	(Time), Synchro-nization	Synchro-nization	Synchro-nization	Synchro-nization
Subscriber	>500	2	~100	~500
Hardware	PC	PC / Mobile Devices	High Level Embedded Devices	Embedded Devices
Configuration	Online	Online	Mostly offline	Mostly offline
QoS	Best effort traffic with reliable transmission	Best effort traffic with reliable transmission	Prioritized traffic with unreliable transmission	Scheduled traffic with unreliable transmission

IV. Communication Technologies in OPC UA

This chapter starts with an overview of OPC UA communication with the focus on PubSub. After that four different combinations are defined.

A. Server/Client-oriented communication

In general Server/Client architectures use connection-oriented communication. Clients establish connections to the servers. The client uses service calls over those sessions to the server. Maintaining those sessions in a One-to-Many scenario can allocate plenty of resources on your device which can be a problem especially for embedded devices. This problem can be solved by using OPC UA PubSub.

B. OPC UA PubSub

The OPC UA PubSub specification has several chapters covering different aspects. The following chapters are meant to give an overview over the content of the OPC UA PubSub specification.

1) Broker concept

A broker is a middleware between two entities that want to exchange data. The sender of the data, called publisher, publishes his messages to the topics of a broker. The receiver, called subscriber, can notify the broker that it's interested in a topic. Afterwards the subscriber receives the messages from the broker. A broker can decouple the communication between the entities in space, time and synchronization.

2) Brokerless concept

The brokerless concept doesn't use a broker but solely uses the infrastructure provided by the network to deliver messages between the publisher and the subscriber. One way to do this is to send the messages to a UDP multicast group. It also decouples the communication between the entities but only in space and synchronization. By sending the message directly to the subscriber, it has smaller latencies and is therefore considered to be the more performant concept compared to the broker concept.

3) Transport Protocols

In OPC UA PubSub there are four protocols that can be used to publish messages. All of those protocols have their own characteristics.

a) Advanced Message Queuing Protocol (AMQP)

AMQP is an open standard application layer protocol for Message Oriented Middleware. It's often used in combination with a broker. A publisher sends the message to the broker. A subscriber conveys its interest in the message of the publisher and therefore receives it from the broker.

b) Message Queue Telemetry Transport (MQTT)

MQTT is an open standard application layer protocol for Message Oriented Middleware. It uses the Publish and Subscribe mechanism to deliver messages. It's standardized in OASIS [9] and widely spread over several use cases e.g. in factory automation [10]. It provides three different levels of Quality of Services which are evaluated e.g. in [4].

c) OPC UA UDP

OPC UA UDP uses the User Datagram Protocol (UDP) to transmit data. UDP is a datagram-oriented transport-layer protocol. It doesn't establish connections between the communication entities. Therefore is has less overhead than other transport protocols e.g. TCP. Since UDP doesn't have ordering and error-checking of the received messages, this needs to be done in the application if required.

d) OPC UA Ethernet

OPC UA Ethernet is an Ethernet-based protocol without IP or UDP headers. It solely evaluates the Data-Link layer information. It has its own Ethertype 0xB62C.

Those Transport Protocols must include Message Mappings in their payload. The next chapter gives an overview on those Message Mappings.

4) Message Mappings

In the PubSub specification Message Mappings describe how to structure and encode the messages. There are two of those mappings in the OPC UA PubSub. One is the JavaScript Object Notation (JSON) and the other one is the UA Datagram Protocol (UADP).

a) JSON

JSON is especially popular with web and enterprise software. It's based on a subset of the JavaScript Programming Language. It uses human readable text and is therefore easy to read and write. It relies on the security mechanisms provided by the broker.

b) UADP

UADP focuses on a very efficient structure to fulfill the requirements of cyclic communication e.g. as needed in C2C-communication or on the field level. UADP defines a NetworkMessage which can be included in the payload of a Transport Protocol.

C. Four different options to combine OPC UA

Fig. 2 gives an overview of possible combinations of Transport Protocols and Message Mappings. To evaluate the applicability of OPC UA mechanisms on the previously discussed use cases, four different combinations of OPC UA mechanisms are selected.

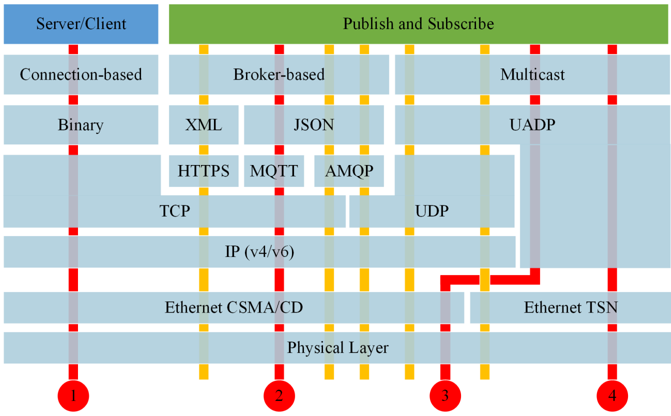

Fig. 2 Overview of possible OPC UA communication technologies.

1) Combination 1: Server/Client with binary encoding

In this combination all the services, e.g. Read/Write/Subscription, provided by the Server/Client architecture can be used. The Subscription service of the Server/Client doesn't have anything to do with the OPC PubSub mechanism. The main difference between those two is, that Subscription from the Server/Client is based on a connection-oriented communication. The services are combined with the OPC UA Binary encoding.

2) Combination 2: PubSub Broker-oriented Communication with JSON encoding

In combination 2 the PubSub broker concept is combined with the JSON Message Mapping. Using a broker between publisher and subscriber gives the additional option to decouple the communication in time, space and synchronization.

3) Combination 3: PubSub Data-Link layer Communication with UADP encoding

Combination 3 uses Data-Link layer communication, which means a connectionless communication between the publisher and the subscriber without any IP-header information. As encoding it takes the UADP Message Mapping, defined in the OPC UA PubSub specification.

4) Combination 4: PubSub TSN with UADP encoding

Combination 4 also uses Data-Link layer communication in combination with Time-Sensitive Networking (TSN) and OPC UA PubSub. The two TSN mechanisms used here are the high accuracy clock synchronization in IEEE 802.1AS-Rev and Time-Aware Shaper (TAS) in IEEE 802.1Qbv.

V. Evaluation and Comparison

Until now the evaluation is based on expert estimates. Fig. 3 has the goal to visualize the estimates on which combination is most suitable for which use case. The fulfillment of the requirements is rated with circles. If a requirement is not fulfilled this is indicated by an empty circle. Vice versa a full circle shows that a requirement is fulfilled completely. A quarter circle means that it already fulfills the requirement but to be able to compare it with the other solutions there is the option to give also half and three quarter of a circle.

Requirements	Combination				
Enterprise	1	2	3	4	**Explanation of the evaluation results**
Latency / Cycle Time	●	●	●	●	All latency and cycle time requirements are fulfilled.
Time Sync Accuracy	●	●	●	●	All time sync accuracy requirements are fulfilled.
Decoupled Communication	◑	●	◑	◑	Server/Client can decouple in synchronization and in time with history service. Broker in time, space and synchronization. The other combinations in space and synchronization.
Number of Subscribers	●	●	●	●	All of the combinations fulfill the requirement. However Servers/Publishers need more resources to manage 500 connections with the Server/Client architecture than with the PubSub architecture.
Hardware Requirements	●	●	●	●	The available hardware in this use case has enough resources to handle the necessary number of connections.
Configuration	●	◑	◔	◔	To configure the communication and the data of interest the Server/Client architecture is easier to use. It can just browse and read the data from the Server without additional configuration.
QoS	●	◕	◔	◔	Server/Client architecture can already use the QoS provided by TCP. In comparison OPC UA PubSub without a broker uses UDP/Ethernet and therefore doesn't have the QoS that TCP provides. Brokers provide usually different level of QoS.
HMI					**Explanation of the evaluation results**
Latency / Cycle Time	●	●	●	●	All latency and cycle time requirements are fulfilled.
Time Sync Accuracy	●	●	●	●	All time sync accuracy requirements are fulfilled.
Decoupled Communication	◑	●	◑	◑	Server/Client can decouple in synchronization and in time with history service. Broker in time, space and synchronization. The other combinations in space and synchronization.
Number of Subscribers	●	●	●	●	All of the combinations fulfill the requirement.
Hardware Requirements	●	●	●	●	The available hardware in this use case has enough resources to handle the necessary number of connections.
Configuration	●	◑	◔	◔	To configure the communication and the data of interest the Server/Client architecture is easier to use. It can just browse and read the data from the Server without additional configuration.
QoS	●	◕	◔	◔	Server/Client architecture can already use the QoS provided by TCP. In comparison OPC UA PubSub without a broker uses UDP/Ethernet and therefore doesn't have the QoS that TCP provides. Brokers provide usually different level of QoS.
C2C					**Explanation of the evaluation results**
Latency / Cycle Time	○	○	◕	●	Server/Client architecture and the broker concept might not be performant enough. Using Layer 2 (L2) communication fulfills the requirement. TSN provides additional mechanisms for real-time.
Time Sync Accuracy	◔	◔	◕	●	As long as there is no other traffic to disturb the communication, the requirements are fulfilled. Otherwise TSN is needed.
Decoupled Communication	●	●	●	●	Server/Client can decouple in synchronization and in time with history service. Broker in time, space and synchronization. The other combinations in space and synchronization.
Number of Subscribers	○	◑	●	●	100 Connections between Server and Clients are not supported by most industrial controls. Therefore Server/Client doesn't fulfill the requirement. In case of a broker it depends on its configured maximum number of connections.
Hardware Requirements	○	○	●	●	The resources are very limited on the required hardware, therefore a lightweight message mapping is needed which is in this case UADP.
Configuration	○	○	●	◔	When it comes to configuring C2C applications, most of it is done offline in a project. Therefore Server/Client architecture, which needs to be online, isn't suitable for it. TSN is very complex to configure.
QoS	○	○	●	●	The retransmission of packages isn't needed in this use case. Therefore the main advantage of TCP can't be used.
Field Level					**Explanation of the evaluation results**
Latency / Cycle Time	○	○	◔	●	Server/Client architecture and the broker concept might not be performant enough. Using Layer 2 communication fulfills the requirement. TSN provides additional mechanisms for realtime.
Time Sync Accuracy	○	○	○	●	TSN is needed to achieve the required time sync accuracy.
Decoupled Communication	●	●	●	●	Server/Client can decouple in synchronization and in time with history service. Broker in time, space and synchronization. The other combinations in space and synchronization.
Number of Subscribers	○	◑	●	●	500 Connections between Server and Clients are not supported by most industrial controls. Therefore Server/Client doesn't fulfill the requirement. In case of a broker it depends on its configured maximum number of connections.
Hardware Requirements	○	○	●	●	The resources are very limited on the required hardware, therefore a light weight message mapping is needed which is in this case UADP.
Configuration	○	○	●	◔	When it comes to configuring field level applications, most of it is done offline in a project. Therefore Server/Client architecture, which needs to be online, isn't suitable for it.
QoS	○	○	○	●	Only TSN can provide scheduled traffic.

Fig. 3 Explanation of the evaluation results.

Fig. 4 gives an overview of the evaluation results. The general trend of the bars in the figure is that the Server/Client architecture is more applicable on the enterprise and HMI level than the new Publish Subscribe architecture.

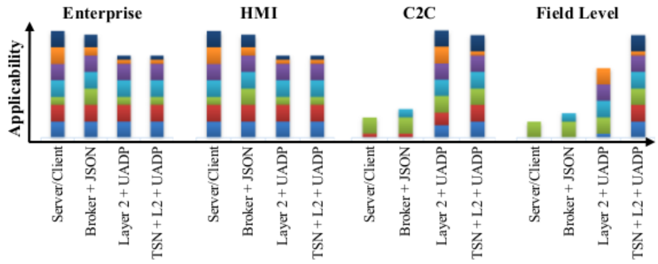

Fig. 4 Overview of the evaluation results.

VI. Conclusion

First we have investigated possible communication use cases from the perspective of an industrial control. Second we have given an overview of today's OPC UA communication portfolio with the focus on the new specification OPC UA PubSub. Third we have defined four different combinations of OPC UA communications to evaluate and compare the applicability of those combinations on defined factory automation use cases based on our experience. Further research is planned on measuring the performance of the described combinations. Especially our assumptions on the applicability of the combinations on the use cases regarding cycle time, time synchronization accuracy and the number of subscribers are of interest and shall be verified through measurements planned in the future.

References

[1] VDI/VDE, "Reference Architecture Model Industrie 4.0 (RAMI4.0): Status Report," 2015.

[2] OPC Foundation, "OPC UA Part 14 - PubSub Specification," 2018.

[3] S. Cavalieri and G. Cutuli, Eds., *Performance evaluation of OPC UA*. 2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010), 2010.

[4] H. C. Jo and H. W. Jin, Eds., *Adaptive Periodic Communication over MQTT for Large-Scale Cyber-Physical Systems*. 2015 IEEE 3rd International Conference on Cyber-Physical Systems, Networks, and Applications, 2015.

[5] A. Balador, N. Ericsson, and Z. Bakhshi, Eds., *Communication middleware technologies for industrial distributed control systems: A literature review*: IEEE, 2017.

[6] A. Botthof and E. Hartmann, Eds., *Zukunft der Arbeit in Industrie 4.0*. Berlin: Springer Vieweg, 2015.

[7] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114–131, 2003.

[8] A. Müller, "Funktechnologien für Industrie 4.0: ITG AG Funktechnologie 4.0," *VDE Positionspapier*, 2017.

[9] OASIS Standard, *mqtt-v3.1.1: MQTT Version 3.1.1*. [Online] Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. Accessed on: Apr. 14 2018.

[10] P. Priller, A. Aldrian, and T. Ebner, Eds., *Case study: From legacy to connectivity migrating industrial devices into the world of smart services*. Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), 2014.