An OPC UA-based Crane Model Using Time-Sensitive Networking Technology

Frank Golatowski*, Hannes Raddatz*, Fabian Hölzke*, Enkhtuvshin Janchivnyambuu*
Peter Danielis† and Dirk Timmermann*
* Institute of Applied Microelectronics and CE
University of Rostock, Rostock, Germany
Email: frank.golatowski@uni-rostock.de
† Department of Computer Science
University of Rostock, Rostock, Germany
Email: peter.danielis@uni-rostock.de

***Abstract*—Current control processes in the material handling domain are generally not automated but require manual human intervention. However, the ongoing development towards the Industry 4.0 involves new models and concepts and hence more and more automated control functions will become necessary. That is, devices such as cranes in the material handling domain will become smarter and will also be networked to a larger extent. However, it will be required that control messages passed to the cranes through a network, possibly shared with other traffic, are guaranteed to arrive in time. In this paper, we present an OPC UA-based model of an overhead travelling crane as part of a material handling scenario. The crane can be controlled by a crane remote controller, which we model as well. We use time-sensitive networking technology to ensure deterministic communication between the crane remote controller and the crane. To demonstrate the crane movement, we illustrate it in a simulation before we prove the deterministic communication in a real testbed.**

***Index Terms*—material handling domain, cranes, OPC UA, time-sensitive networking, Industry 4.0**

I. Introduction

Today's control of industrial processes such as those in the material handling domain (MHD) works mainly manually, i.e., the control is carried out by humans [1]. Future concepts require on the one hand more automated control functions, for example, to control the movement of cranes, while on the other hand ensuring their timely and deterministic execution [2]. Necessary measures concern improving the aspects of distributed networked control, adapting IoT technologies to industrial requirements, improving control and assistance applications through context and location awareness as well as a design based on common models [3]. This supports partners and industry in the MHD in preparing for the challenges of Industry 4.0. OPC UA is a widely-used machine-to-machine communication protocol developed by the OPC Foundation and has been standardized in the IEC 62541 series [4]. It provides interoperability in process automation and a service-oriented architecture for industrial applications. Hence, it is feasible to model the system architecture of a more intelligent and automated crane in the MHD.

The objective of this paper is therefore to develop an OPC UA-based model for overhead travelling cranes to be used in the MHD while ensuring deterministic communication. Based on an analysis and evaluation of existing OPC UA tools, stacks, and platforms, we develop an OPC UA framework, which can be used in the MHD. In particular, we model a location-aware overhead travelling crane using the OPC UA framework and design the movement of a crane using a crane remote controller, which forms the basis for realizing assistance function such as come-to-me or go-to for cranes. We first simulate the crane movement and then show in a real testbed that deterministic communication between a crane and its crane remote controller can be achieved by using time-sensitive networking (TSN) technology [5].

The main contributions of this paper are:

- Development of a location-aware crane model based on the information model of OPC UA
- Design of crane movement based on the developed model
- Demonstration of the designed functionality in a simulation and proof of deterministic communication in a real testbed

The rest of this paper is organized as follows. Section II discusses related work. Section III presents the developed crane model. Section IV describes the implementation and results from simulation and testbed evaluation before Section V concludes the paper.

II. Related Work

Currently, there is a wide variety of projects ongoing to fulfill the new Industry 4.0 requirements. In what follow we focus on two competitors of OPC UA in the industrial automation field, which are the Data Distribution Service (DDS) standard [6], [7] and the open-source Web Object Oriented Protocol for Software and Automation (WOOPSA) protocol [8], [9].

Both DDS and OPC UA standards are designed to solve the same problem of information management in distributed systems in different ways. Also, both standards provide support for information modeling. DDS accomplishes that with relational data modeling whereas OPC UA uses object-oriented modeling. DDS is an open middleware standard that is based on the publisher/subscriber model which enables scalable, realtime, dependable, high-performance and interoperable data exchanges between publishers and subscribers. DDS is broker-less and provides a standard API as well as an interoperable wire protocol. It is mostly used in dynamic environments because of its data-centric model. For example, in a hospital scenario, the device can be moveable. At one moment, it generates data related to patient A from room A and after some time the same device may send out data of patient B from another room. However, it is not of interest for users which device sends data from where since they only need data for a specific patient. In contrast, OPC UA is more common in the static configuration environment that depends on the network topology.

WOOPSA is an emerging Industrial Internet of Things protocol. It is simple, lightweight, free, open-source, web and object-oriented, publish-subscribe, realtime capable and Industry 4.0 ready. WOOPSA is based on the object-oriented paradigm which models any application in object-oriented manner which makes it an approach similar to OPC UA. It is web-based and uses HTTP and JSON. WOOPSA supports remote access to structured data between applications in different languages like C#, JavaScript and C. It represents an alternative to OPC UA, but its implementation is a lot heavier than OPC UA because of using JSON over HTTP. Thus, it is less suited for resource-constrained devices.

Both DDS and WOOPSA protocols have advantages and disadvantages compared to OPC UA. However, OPC UA is well standardized and open-source. Most importantly, it is supported by key manufactures in the industrial automation field, and it has emerged from the automation industry. Thus, OPC UA's ecosystem is its biggest strength. Also, there are many open-source implementations of OPC UA in many different programming languages including C, C++, Java, .NET and Python. In addition, OPC UA provides flexible mechanisms for exchanging data between enterprise level systems and field devices which interact with real-world data. Due to this flexible mechanisms that can integrate data generated by low level devices with enterprise systems and its ecosystem, we have selected OPC UA for our developed model. For a comparison of the performance of OPC UA to other Industry 4.0 protocols, the interested reader is referred to [10].

III. Crane Model

OPC UA was initially developed based on the client-server pattern and integrates the publish/subscribe communication model as well [11]. The system architecture is designed as follows: the overhead travelling crane is an OPC UA server and the crane remote controller typically used to control the crane is an OPC UA client. The network connection between crane and controller is supposed to be a wired link since we want to evaluate our crane model in a TSN-capable network, but could however be a wireless link, prospectively. The generic system architecture that describes the main structure of the server and client applications is defined in accordance with Figure 1. Also, this section describes the details of the required functions and mechanisms as well as dependencies for a prototypical implementation of a crane remote control. First, certain general architectural decisions are described, followed by a specific description regarding OPC UA.

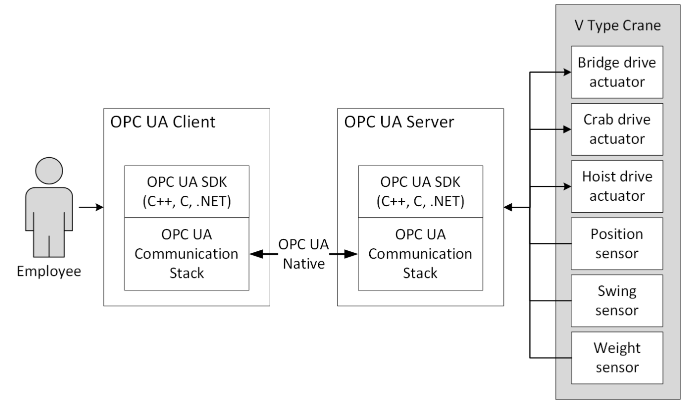

Fig. 1. Generic system architecture.

Fig. 2. Overhead travelling crane and controller.

A. System Architecture

As shown in Figure 2, an employee controls an overhead travelling crane using a remote controller. Basically, our system architecture and use-case diagrams are defined based on this main operation. In accordance with UML, see use-case diagram in Figure 3, there are two main actors: an employee who controls a crane and the crane which is controlled by an employee. The OPC UA client is a controller, which is used by the employee. It has eight main functionalities including moving forward/backward, moving left/right and lifting up/down. Also, there is an emergency stop function, which is used in safety related emergency cases and the come-to-me function, which is based on a indoor-positioning system (IPS). The OPC UA client is a digital twin of the Demag D3 radio control that is commonly used to remotely control Demag V-type cranes [12]. The OPC UA server is an embedded system that activates or deactivates actuators of the crane based on command from the client and data from the cranes sensors. The crane sensors and actuators are directly connected to the embedded system. The stop function of the OPC UA server is activated in the case of receiving stop command from the client or alarm from the swing and position sensors of the crane itself. As illustrated in Figure 3, the stop function is an internal part of control of all drives whereas it is an extended function for the swing and position sensors which activate the stop function in the pre-defined

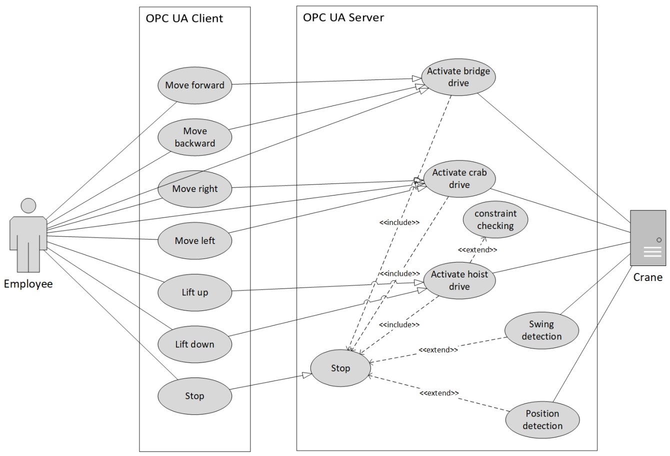

Fig. 3. Mapping of OPC UA client to server functions.

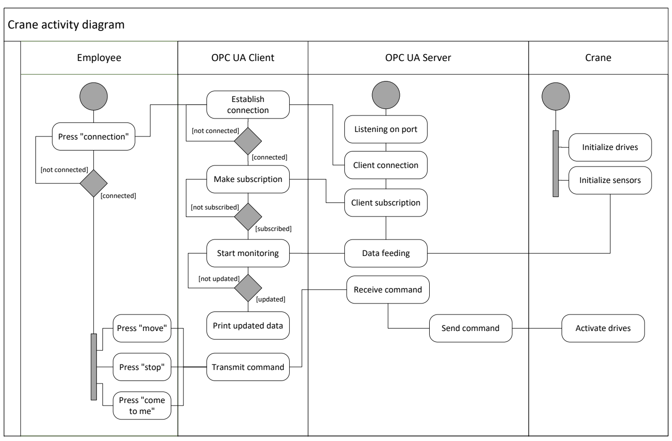

Fig. 4. Activity diagram for crane automation.

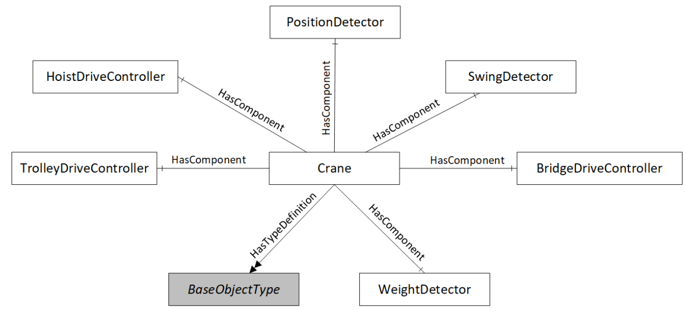

Fig. 5. Crane object model.

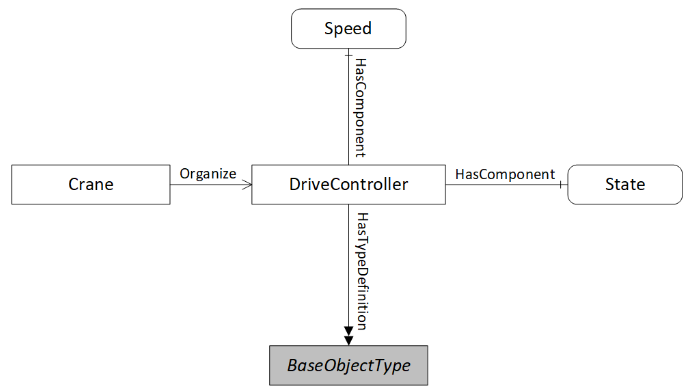

Fig. 6. Object model for drive controllers.

cases. According to the activity diagram in Figure 4, an employee initializes all activities by pressing the connection button. Once the connection between OPC UA server and client is established, the OPC UA client automatically makes a subscription to the OPC UA server and starts the data monitoring for crane positioning and alarming. When data is changed on server side, the client receives the updated data and displays them in the GUI of the client application. After the connection establishment, the employee can control the crane using movement buttons. When he presses one of the movement buttons, the OPC UA client submits the respective data to the server. Once the server receives them, it activates the drive of the crane. The crane feeds its sensors data to the server and server will transmits them to the subscriber. The employee needs to hold the movement button until a crane reaches a desired position.

B. OPC UA Model for Crane Automation

There are many possible ways to model the crane using OPC UA. For example, data with the same purpose can be grouped as an object or we can define the object based on the main functionality. In order to simplify the modelling and programming process, it was decided to model a crane in an object-oriented manner. This way, all components of the crane are modelled as independent objects. As shown in Figure 5, all objects are created using ”BaseObjectType” and all sub-objects are referenced to the main object ”Crane” using ”HasComponent” as reference because these sub-objects are internal components of the crane. Modelled like this, it is convenient to expand the model when a new component is added to the crane. The drives of bridge, crab, and hoist are the main components of the crane. They have the same structure and functions. Due to their similarity, the model of these objects is the same. As illustrated in Figure 6, a controller object has two main variables: *speed* that indicates movement of the crane in vertical or horizontal direction and *state* that indicates that the drive is on or off. The position detection sensor model is illustrated in Figure 7. The position detection sensor is part of an indoor localization system assumed to be available, see, e.g., [13]. The sensor object has four main variables: *PositionX*, *PositionY* and *PositionZ* that indicate positions of the hoist in accordance to vertical and horizontal directions and *state* that indicates whether the sensor is on or off. It is assumed that the sensor sample rate is happening sufficiently fast so that the crane can adapt its position in time.

The swing detection sensor is modelled as illustrated in Figure 8. The sensor object has two main variables: *swing* which is represented by a boolean value and *state* which indicates whether the drive is on or off. If the magnitude of the hoist swing passes a threshold, its value changes to true

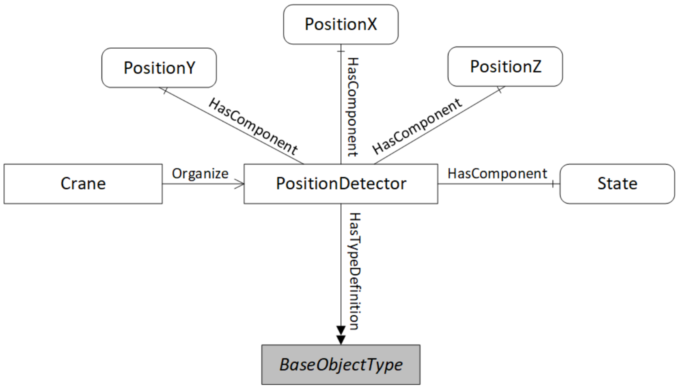

Fig. 7. Object model for *PositionDetector* object.

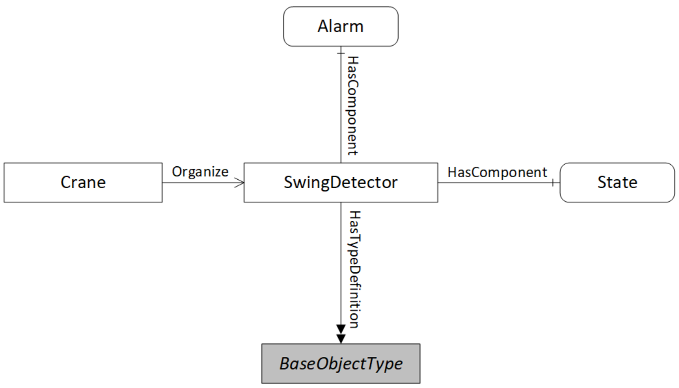

Fig. 8. Object model for *SwingDetector* object.

and otherwise it is always false. It is assumed again that the sensor provides data sufficiently fast.

The weight detection sensor is modelled as shown in Figure 9. The sensor object has two main variables: *weight* that indicates the load of the hoist and *state* that indicates if the sensor is on or off. The sensor is supposed to provide data in real-time.

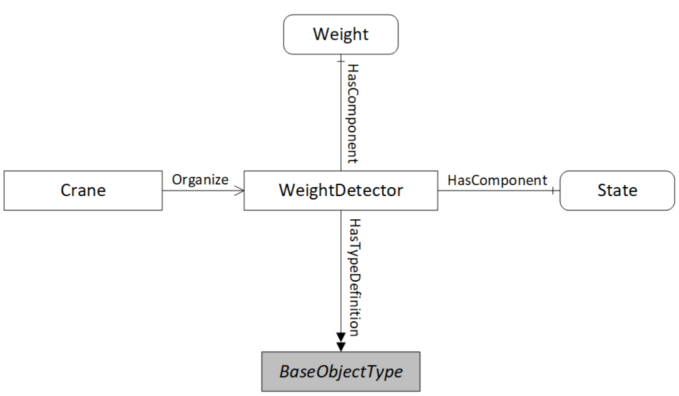

Fig. 9. Object model for *WeightDetector* object.

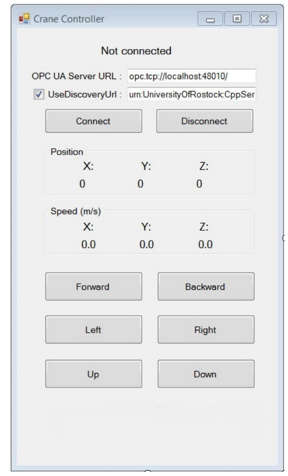

Fig. 10. OPC UA client GUI representing the crane remote controller.

IV. Implementation and Results

This section first describes the implementation of the developed OPC UA crane model. Subsequently, it describes the simulation implementation to demonstrate the crane movement. Finally, the testbed implementation and setup as well as measurements obtained from it are presented.

A. OPC UA Implementation

OPC UA server (crane) and client application (crane remote controller) were programmed in two different variants to be able to compare their suitability for implementing the model: (1) UA SDK [14] and (2) the open62541 library [15]. In terms of programming, the open62541 library was easier to use compared to the UA SDK because the UA SDK requires more coding overhead for definition of a class for each object type, its object implementation, and other classes for accessible nodes. An exhaustive comparison of existing OPC UA implementation can be found in [16].

As illustrated in Figure 10, the crane controller has the following possibilities:

- Move forward and backward (activate the bridge drive)
- Move right and left (activate the crab drive)
- Move up and down (activate the hoist drive)

The position of the hoist and speed of drives are displayed on the GUI in realtime.

B. Simulation Implementation and Function Demonstration

To demonstrate the movement of a crane, a 2D crane simulation has been implemented in C# based on the crane specifications of Demag [12]. The swing detection sensor is optional for crane installation and a position detection sensor is not contained in the standard crane. However, we assume that a indoor positioning system will be used for the crane automation to prevent collision and to implement additional semi-autonomous function such as come-to-me or go-to prospectively.

From Figure 12, the GUI is apparent, which shows the movement of the crane from side view and top view. The

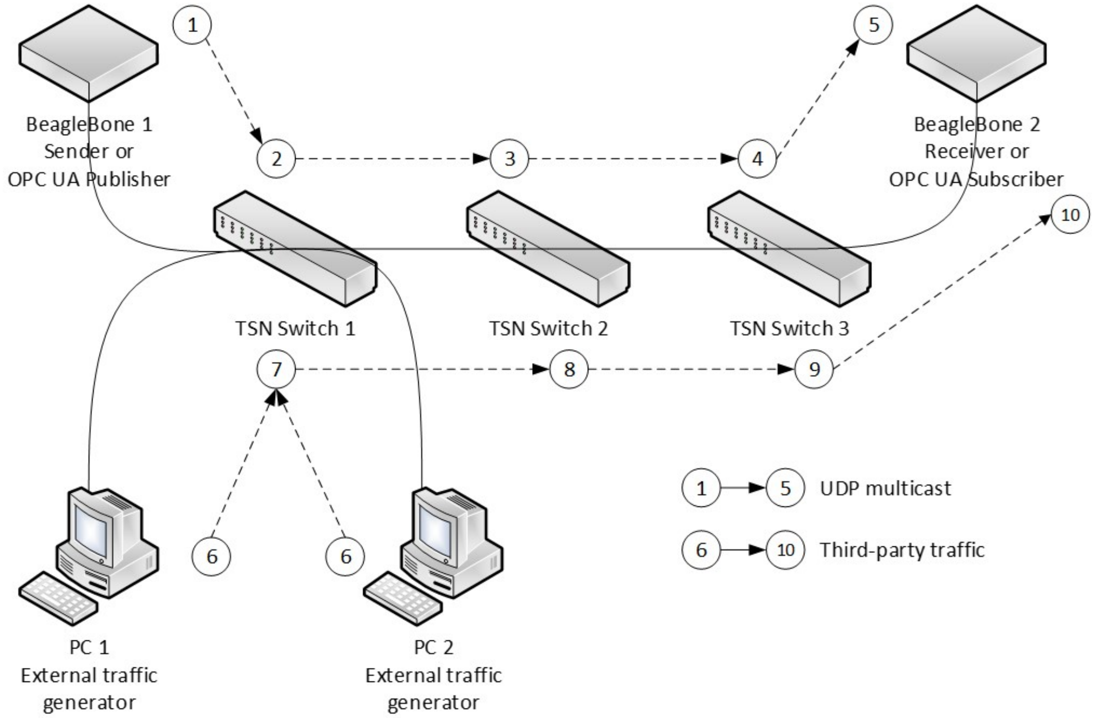

Fig. 11. Testbed for the experiment with best-effort and time-triggered multicast traffic generated by pre-installed OPC UA publish/subscribe applications.

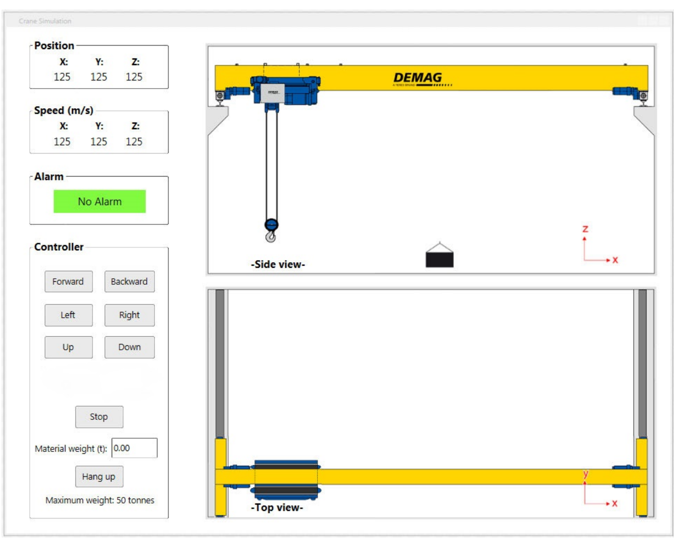

Fig. 12. GUI of 2D crane simulation demonstrating the movement of a crane.

crane moves with constant velocity in the simulation and its position is given.

C. Testbed Setup and Implementation

To use OPC UA for a system control task, the deterministic transmission of packets is required. This can be achieved by combining OPC UA and TSN, referred to as OPC UA over TSN. We have used the TTTech Starter Kit which contains three pre-configured 100 Mbps TSN switches and two BeagleBone Black boards with a time synchronization feature based on the IEEE precision time protocol (PTP). OPC UA publish/subscribe applications are pre-installed on the BeagleBones. Beaglebones are used as end points that host the OPC UA server (crane) and client (crane remote controller), respectively, and deterministic Ethernet Akro TSN switches of TTTech are used as networking switches.

The setup of our testbed is apparent from Figure 11. The publisher, BeagleBone 1, sends a UDP multicast packet with VLAN tag 3 every 50 ms to the receiving subscriber, BeagleBone 2. This traffic is supposed to be the realtime control traffic. Please note that the periodic sending interval of 50 ms is due to the preconfigured behavior of the testbed and, according to the documentation, could be set to other values by the vendor when applied in practice. The three TSN switches have preconfigured rules for UDP multicast packets with VLAN tag 3 so that they directly route and forward these packets deterministically. The objective of this experiment is to make sure that the real-time control traffic is always preferred in front of best-effort traffic and therefore the interval between two consecutive messages should be constant. Hence, two additional PCs each generating up to 100 Mbps inject third-party traffic into the network to find out if this traffic obstructs the real-time control traffic, which it should not do.

D. Testbed Results

Eight measurements were taken for different third-party traffic loads. Table I shows that the result of each experiment is the same, as expected. The time between two consecutive received messages at BeagleBone 2 was 50 ms even in the presence of 200 Mbps third-party traffic. Each PC generates

TABLE I

MEASUREMENTS FOR DIFFERENT THIRD-PARTY TRAFFIC LOADS.

Third-party traffic [Mbps]	**Interval between two consecutive received multicast messages [ms]**
0	50
50	50
60	50
70	50
80	50
90	50
100	50
200 (100+100)	50

100 Mbps to be sure to fully utilize the links to the switches causing the switches to drop third-party traffic and to prioritize the real-time traffic received from BeageBone 1 in front of it.

V. CONCLUSION

In this paper, we proposed an OPC UA model for an overhead travelling crane, which is location aware and supports different assistance functions such as come-to-me and go-to. The crane can be controlled with a crane remote controller. To demonstrate the movement of a crane, a simulation has been developed. Finally, with the help of the implemented OPC UA crane model, is has been proven in a testbed that the communication between crane and crane controller is deterministic when using time-sensitive networking technology. Prospectively, several assistance functions will be implemented and different kind of sensors and localization systems will be integrated in practice.

ACKNOWLEDGMENT

This work has been achieved in the European ITEA project ”OPTimised Industrial IoT and Distributed Control Platform for Manufacturing and Material Handling (OPTIMUM)” and has been funded by the German Federal Ministry of Education and Research (BMBF) under reference number 01IS17027. We want to thank all partners in the OPTIMUM project for the stimulating discussions and their contributions to the project. All project partners can be found onhttps://itea3.org/project/optimum.html.

We further would like to thank the German Research Foundation (DFG) (project ”Real-Time Publish/Subscribe for the Industrial Internet of Things”, GZ: TI 254) for their financial support.

REFERENCES

[1] E. Skilling and C. Munro, “Human factors in materials handling,” in *Human Factors in the Chemical and Process Industries*. Elsevier, 2016, pp. 257–270.

[2] H. Kumar and S. Kamboj, “Modelling of automatic material handling system using plc,” *International Journal*, vol. 8, no. 3, 2019.

[3] P. Hyla, A. Kosoń-Schab, J. Szpytko, and J. Smoczek, “Integrated supervision for supporting control and proactive maintenance of material handling system,” *Journal of KONES*, vol. 26, no. 1, pp. 65–72, 2019.

[4] M. Schleipen, S.-S. Gilani, T. Bischoff, and J. Pfrommer, “Opc ua & industrie 4.0-enabling technology with high diversity and variability,” *Procedia Cirp*, vol. 57, pp. 315–320, 2016.

[5] D. Bruckner, M.-P. Stănică, R. Blair, S. Schriegel, S. Kehrer, M. Seewald, and T. Sauter, “An introduction to opc ua tsn for industrial communication systems,” *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1121–1131, 2019.

[6] Object Management Group, Inc., “Data distribution service for real-time systems (DDS), version 1.2,” Needham, MA, USA, Jan. 2007. [Online]. Available: http://www.omg.org/cgi-bin/doc?formal/07-01-01

[7] J. Yang, K. Sandström, T. Nolte, and M. Behnam, “Data distribution service for industrial automation,” in *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*. IEEE, 2012, pp. 1–8.

[8] “Woopsa, Industry 4.0 Internet of Thing Protocol,” 2020. [Online]. Available: http://www.woopsa.org/

[9] J. E. Rubio, C. Alcaraz, R. Roman, and J. Lopez, “Analysis of intrusion detection systems in industrial ecosystems.” in *SECRYPT*, 2017, pp. 116–128.

[10] S. Profanter, A. Tekat, K. Dorofeev, M. Rickert, and A. Knoll, “Opc ua versus ros, dds, and mqtt: Performance evaluation of industry 4.0 protocols,” in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, 2019.

[11] A. Eckhardt, S. Müller, and L. Leurs, “An evaluation of the applicability of opc ua publish subscribe on factory automation use cases,” in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2018, pp. 1071–1074.

[12] DEMAG, “Revolution in crane engineering - v-type crane,” 2020. [Online]. Available: https://www.demagcranes.com/en/products/cranes/universal-cranes/v-type-crane

[13] F. Hölzke, H. Raddatz, F. Golatowski, D. Timmermann, and J. Lategahn, “Evaluating the synergy of relative and absolute indoor localization in industrial spaces,” in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1. IEEE, 2019, pp. 4215–4220.

[14] Unified Automation, “OPC UA SDK .NET/C++ Documentation,” 2020. [Online]. Available: https://www.unified-automation.com/downloads/documentation.html

[15] “open62541 implementation for OPC UA,” 2020. [Online]. Available: https://open62541.org/doc/current/toc.html

[16] H. Haskamp, M. Meyer, R. Möllmann, F. Orth, and A. W. Colombo, “Benchmarking of existing opc ua implementations for industrie 4.0-compliant digitalization solutions,” in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. IEEE, 2017, pp. 589–594.