Enabling Industrial Motion Control through IIoT Multi-Agent Communication

Razvan-Adrian Luchian
Research and Development
Asti Automation
Bucharest, Romania
razvan.luchian@astiautomation.com

Sabin Rosioru
Automation and Industrial Informatics
University "Politehnica" of Bucharest
Bucharest, Romania
sabin.rosioru@stud.acs.upb.ro

Iulia Stamatescu
Automation and Industrial Informatics
University "Politehnica" of Bucharest
Bucharest, Romania
iulia.stamatescu@upb.ro

Ioana Fagarasan
Automation and Industrial Informatics
University "Politehnica" of Bucharest
Bucharest, Romania
iulia.stamatescu@upb.ro

Grigore Stamatescu
Automation and Industrial Informatics
University "Politehnica" of Bucharest
Bucharest, Romania
grigore.stamatescu@upb.ro

***Abstract*—Increasing complexity and performance requirements of distributed industrial applications offer fertile ground for the development and evaluation of novel industrial communication and control paradigms. As a representative example, multi-agent system (MAS) design relies on distributed software agents that collaborate to achieve joint objectives. The application of MAS in industry is currently emerging as a significant trend to bridge the information technology (IT) and operational technology (OT), previously formally separated, domains. We present a multi-agent distributed system implementation for IIoT communication. The system is evaluated through a motion control scenario with IIoT agent data collection and relaying control commands over standards-based industrial networks with a PLC-based control system. The system architecture, enabling technologies and interoperability aspects between open-source and proprietary components are discussed. Experimental deployment of a conveyor-driven motion control system with S7-1200 PLC, Profinet and IO-Link industrial communication and the Coaty distributed agent framework validates the approach.**

***Index Terms*—industrial internet of things, multi-agent systems, distributed control, communication protocols, cyberphysical systems**

I. Introduction

Industry applications are currently under pressure to deliver timely, cost efficient and environmentally sound products. Modern technologies for factory automation cover intelligent sensors and controllers that not only monitor and actuate upon the production process but also have extended data processing, control, cognition and communication capabilities. The benefits of opening up industrial networks to external third parties have to be carefully balanced against potential security risks. New control and optimization algorithms implemented as external components to the plant network have to be validated and trusted in a contained sandbox before being allowed to operate on plant data and control the production facility. At the same time bridging heterogeneity across proprietary manufacturer equipment, legacy devices and data silos remains a significant research and development challenge.

The Internet of Things (IoT) consists of dense networks of embedded data acquisition, computing and communication devices which are deployed to instrument the physical world at high spatial and temporal resolutions. When applied in the industrial domain, with its stringent dependability, reliability and security requirements, we address them as Industrial Internet of Things (IIoT) networks. The main promise of such system is to compress the conventional automation pyramid to two levels: highly intelligent field level devices for sensing, control and actuation, and a higher level infrastructure, e.g. cloud based, that offers increased resources for advanced learning and optimization primitives, while assuring compliance with standardized service level agreements (SLAs) [1].

Per definition [2], an agent is a physical or virtual entity acting in a structured or unstructured environment. It has the ability to perceive various characteristics of the environment through sensors and act upon the environment through actuators, thus altering its properties in a desirable manner. For the industrial domain and specifically factory automation, the agents act in structured environments, determined by the predefined layout of production processes. By extension to multi-agent systems, further agent interactions and synchronization tasks are considered to achieve joint objectives. One important direction for research is how to adapt such systems to improve the production process while considering conflicting objectives of productivity, resource efficiency and market-imposed flexibility demands, such as batch size optimization [3] and batch-size-one.

In the presented context, considering the growing role of IIoT solutions for IT and OT convergence, the main contributions of the paper are related to:

- Conceptualization of a distributed industrial system ar-

Financial support from the Competitiveness Operational Program 2014-2020, Action 1.1.3: Creating synergies with RDI actions of the EU's HORIZON 2020 framework program and other international RDI programs, MySMIS Code 108792, project acronym "UPB4H", financed by contract: 250/11.05.2020 is gratefully acknowledged.

chitecture with software and hardware based multi-agent entities communicating over heterogeneous protocols and interfaces;
- Implementation and evaluation of an IIoT-enabled motion control application in which a Siemens S7-1200 PLC controls a conveyor belt system through a Siemens G120 static frequency converter, based on data from a network of dedicated software agents which read the relevant sensor values and provide control signal services in real time.

The rest of the paper is structured as follows. Section II briefly reviews related work which focuses on complex industrial communication paradigms and approaches for relevant applications. We describe in detail the methodology of the current work, the designed system architecture and enabling technologies in Section III. Section IV presents the experimental implementation that embodies the discusses conceptual design as a laboratory set-up. Section V concludes the paper with possibilities for future work and replication of the architecture at scale.

II. Related Work

The design and testing of reliable real-time communication networks for industry represents a timely research topic. [4] describes the usage of OPC-UA publish-subscribe mechanisms in an industrial scenario through simulations based on the OMNeT++ networking environment. In order to accommodate real-time requirements a Time Sensitive Networking (TSN) layer is implemented that is routed through dedicated networking components. Using dedicated simulation environments, replicable communication libraries can be tested and then deployed on commodity embedded hardware for experimental validation.

Practical TSN implementation with motion control and robot synchronization is described in [5]. Over a three-layer communication architecture composed of: factory cloud layer, edge layer and field layer the authors evaluate communication system latency as key metric to evaluate complex heterogeneous industrial networks in time critical applications. Reported results include worst-case latencies for various testing scenarios where the robotic arms communication over TSN have to be coordinated with a system of conveyor belts controller over Powerlink interface. A graphical user interface is provided to control the synthetic traffic generation used in the experiments.

Machine learning use cases and system design from an IIoT communication perspective are discussed in [6]. First a classification is carried out in terms of reliability, latency, data rate, connection density, criticality for typical IIoT devices in smart factories. This underlines the need for an adaptive approach that accommodates various orders of magnitude in variation for these metrics. Subsequently a series of state-of-the-art neural network architectures are mapped onto use cases to predict and classify generated network traffic for optimization and prioritization purposes.

The current contribution builds upon previous own work in [7] where we focused on testing the Coaty[1] distributed agent framework in a data acquisition and system monitoring scenario. Based on the achieved results we are able to scale up the system and include control requirements for the motion control use case. We worked also on integrating WSN sensor systems with a cloud back-end for industrial applications [8]. The need for well specified interfaces and APIs is underlined based also on the specfic cloud provider connection patterns, whether Microsoft Azure, Amazon AWS, Google Cloud, and others. Fog computing architecture was defined and evaluated in [9] together with several algorithms for event-based data reduction to limit communication bottlenecks in networks of constrained embedded devices. The payload part of the data packet is compressed as the sensor nodes are aware of the date type and timeliness requirements of the monitoring.

III. Methodology

This paper proposes a system architecture for decentralized supervision and control of an industrial system. Looking at the new paradigms of IIoT [10], the system was implemented using high level communication protocols and the latest open-source frameworks.

This architecture is based on ad-hoc and collaborative communication between software agents. Within the network, agents are able to collect information about the state of the industrial system and are also the entities that control how it operates. One type of distributed algorithm that can be run over newtorks of industrial software agents is the consensus algorithm for distributed agreement [11]. The discrete time communication consensus is expressed as:

$$x_i[k+1] = \sum_{j=1}^{n} d_{ij}[k] x_j[k] \tag{1}$$

where the agreement is reached among agents when $|x_i[k] - x_j[k]| \approx 0$. d_{ij} in this case is the adjancecy matrix which symbolizes physical or logical connections between agents.

The IIoT system architecture for pervasive monitoring and control is presented in Figure 1. It includes three layers:

- Decision making and optimization layer with distributed communication among software agents;
- Local monitoring and control layer that includes the data acquisition and control devices running application-oriented code;
- Field layer that represents the manufacturing process as conveyor motion control application in our particular case.

Each agent present in the network uses two fundamental IIoT communication protocols: MQTT and OPC UA. The MQTT protocol is used to collect data from the automation system using the publish-subscribe functionality. At the same time, this protocol is used for the control of the industrial system, but also for the interconnection between agents in the

[1] https://coaty.io

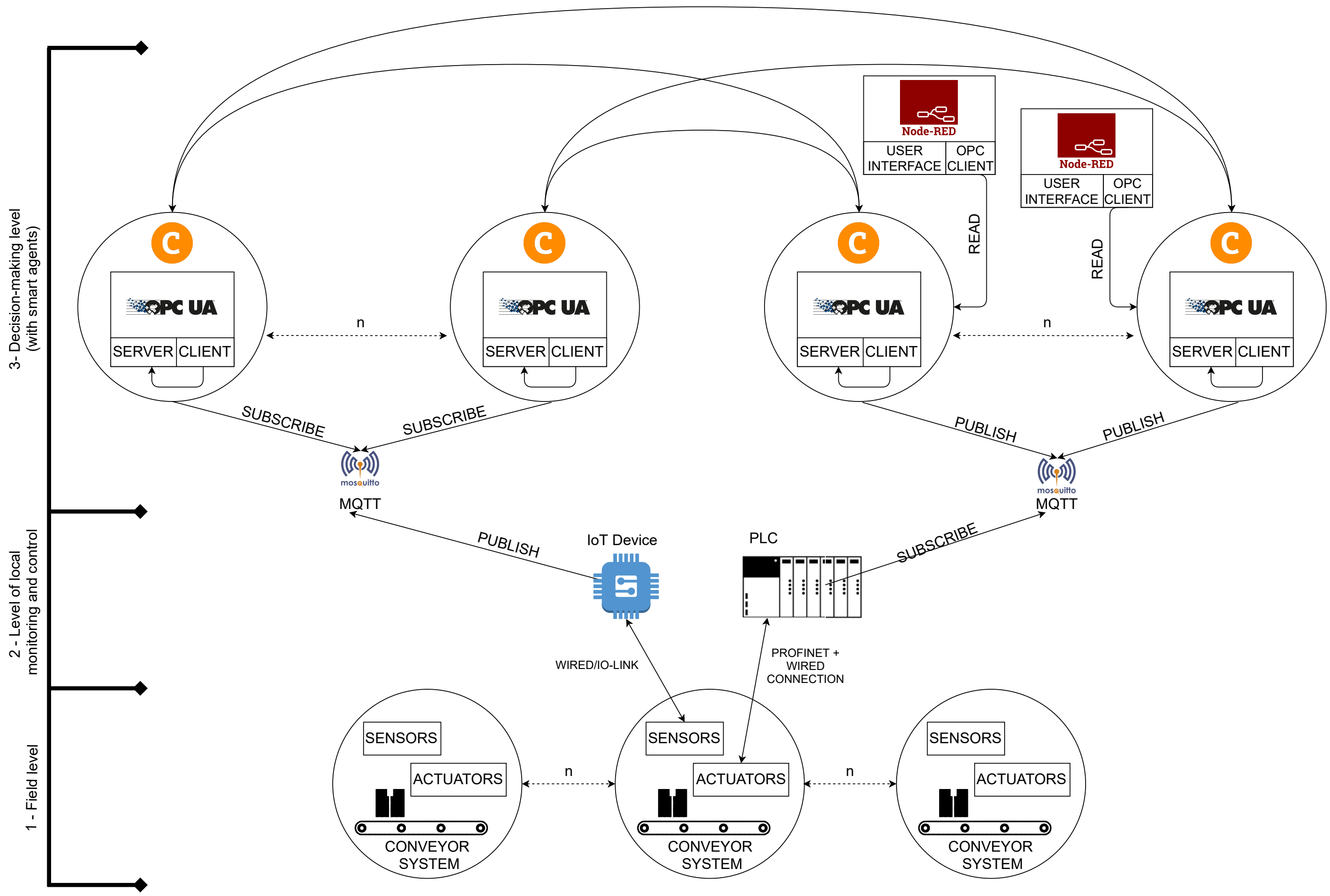

Fig. 1: IIoT System Architecture for Pervasive Monitoring and Control - Advanced Architecture with Scalability

network. The OPC UA protocol based on the client-server relationship is also used in several situations: for storing the data retrieved via MQTT for the subsequent connection of a monitoring system, at the level of the agents responsible for data collection) and for the connection with the control interface [12], at the level of the agents responsible for control.

The most significant benefit of MQTT is that it can provide real-time and dependable messaging services for remote devices with minimal code and restricted bandwidth. It has a wide range of applications in the Internet of Things, small devices, and mobile applications because it is a low-overhead, low-bandwidth instant messaging protocol [13].

MQTT is designed to use very low overhead, which ensures lossless two-way asynchronous communication and two-way communication in constrained networks. Importantly, MQTT is not a message-oriented middleware where messages are stored and delivered. It stands out as an efficient protocol for connecting devices in real-time and reliably. It provides 3 modes of data delivery: QoS0 - at most once, QoS1 - at least once and QoS2 - exactly once. At the core of MQTT there are two types of elements: the MQTT clients that publish and/or subscribe to certain topics of interest; the MQTT broker that manages the messages circulating through the network [14].

The OPC Foundation developed OPC-UA, a machine-to-machine communication protocol that is based on the IEC62541 standard and uses the TCP binary protocol. OPC-UA defines the structure of semantic information models, as well as how information is exchanged between communication partners and how these models can be extended with user-specific models, and could be used as a communication protocol between sensors and IT applications in Industry 4.0 [15].

IO-Controllers i.e., devices that control communication, such as PLCs and IO-Devices are two types of PROFINET nodes i.e. the field devices like remote IO, sensors and actuators. Real-time communication channels are used to transfer process data between the IO-Controller and the IO-Devices. Other information, such as configuration and statistics, is sent over a non-real-time channel, such as UDP/IP. PROFINET devices cyclically switch process data using Layer 2 messages that cannot cross IP routers to ensure low latency efficiency [16]. Depending on the scope there are 3 types of Profinet: NRT, RT and IRT. These types are primarily differentiated by latency as shown in Table 1.

TABLE I: Profinet latency

Application area	Profinet type	Latency
Process automation	NRT	100ms
Factory automation	RT	10ms
Motion control	IRT	< 1ms

IV. Results

The experimental setup we implemented to demonstrate our approach consists of an assembly station based conveyor with suitable automation hardware installed. It is controlled via a driver by the control agent that decides its movements. In this sense, the PLC is used as an intermediary between the control agent and the driver due to the possibility of using the Profinet protocol it offers. The PLC also contains an energy management module which is used to record the current consumption of the motor at different speeds which we use to highlight the functionality of the control agent. In this assembly there is also a gateway that offers the possibility of communication via MQTT through an IoT port to which the inductive sensors on the conveyor are connected. Above the conveyor is the element that also defines the role of this assembly station, namely a pallet storage. In the assembly line, this station is used to supply the pallets on which the part to be assembled will be positioned. The pallet is equipped at each corner with metal pins to be detected by the sensors on the conveyor ends. The software agents that manage this assembly station run on two Raspberry Pi model 3B+ boards. The whole hardware architecture is shown in Figure 2.

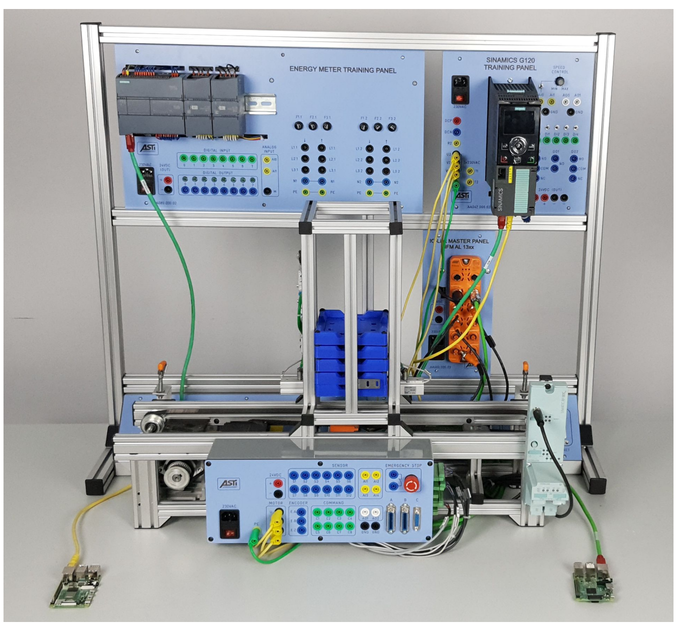

Fig. 2: Assembly station

On the software side, the multi-agent system implementation is based on 2 types of agents:

- data acquisition agent;
- control agent.

A. Data acquisition agent

This type of agent is a low resource cost software entity that uses Typescript as its programming language. The communication protocol used to collect process data is MQTT. Within the application, the software runs on a low-cost Raspberry Pi model 3B+ single board computer. This node communicates via the MQTT protocol with an IFM AL1302 IO-Link master communication module that is able to receive information from the sensors in the system via IO-Link or electrical connection. Thus, the agent collects information from inductive sensors on the conveyor heads, inductive sensors on the pneumatic pistons symbolizing their position. At the same time, an OPC UA server is opened at the Raspberry Pi board level. Within this OPC UA server, the agent stores sensor data collected via MQTT using an OPC UA connector to provide the possibility for other entities outside the agent network to access this information. All this information of the conveyor system can be accessed by agents in the network that are interested in the topic to which it is published by the AL1302 module. The principle architecture of the process data acquisition system can be found in Figure 3.

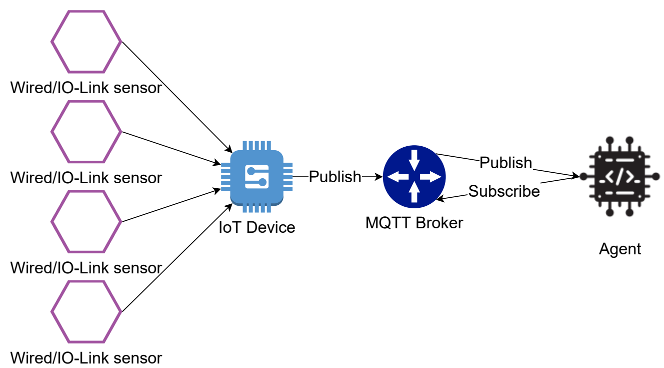

Fig. 3: Data Acquisition Agent

The information collected from the sensors is then parsed into a system status message, which will be made available in the network to the agents controlling the system. Each character in this status message(Figure 4 shows the characteristics of the control message such as message length and message topic length.) has a meaning corresponding to the sensors in the automation system which can be seen in Table 2.

TABLE II: Status message

Character number	Meaning
0	The status of the inductive sensor at the left end of the conveyor
1	The status of the inductive sensor at the right end of the conveyor
2	the status of the inductive sensor of the upper piston in the warehouse
3	the status of the inductive sensor of the lower piston in the warehouse

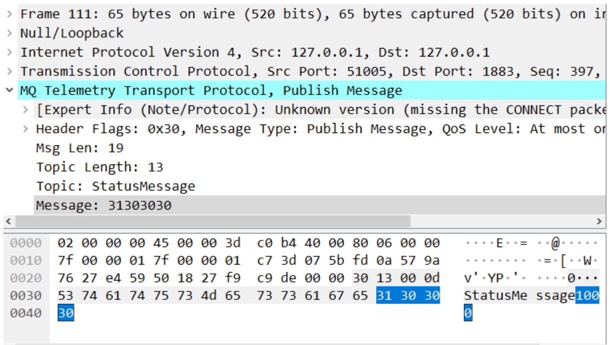

Fig. 4: Status Message characteristics

B. Control agent

This type of agent is very similar in construction to the agent that collects data. The difference between the two types of agents is in their role in the network. The control agent, as the name implies, is in charge of managing the control of the system. It runs on the same type of embedded device, Raspberry Pi model 3B+, and interconnects via the MQTT protocol with a Siemens S7-1200 programmable logic controller (PLC), and via the OPC UA protocol with the control interface developed in Node-RED. The OPC UA server stores the commands transmitted by the user via the control interface, such as commands to start the process, change the conveyor speed or change the direction of movement. Depending on these commands, the agent manages the operation of the system. The architecture of the control system is shown in Figure 7.

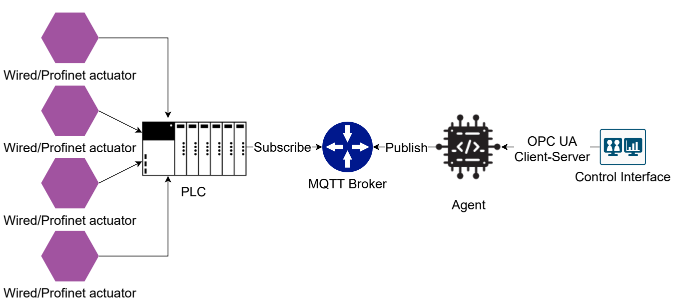

Fig. 5: Control Agent

At the PLC level, the actuators are connected in two ways: electrically and via the Profinet communication protocol. The pneumatic pistons are connected electrically to the PLC's digital outputs, and the Siemens G120 converter that drives the conveyor motor communicates via Profinet.

For the operation of the conveyor system, a control message is created where each character has a certain meaning which is shown in Table 3. Depending on the commands received from the user and depending on the status message obtained from the purchasing agents, the value of the characters in the control message is changed. Commands to the actuators are in binary form 0 and 1, except for the motor speed setting, where the associated character takes values between 1 and 4 and signifies percentage speed values (25%, 50%, 75% and 100%). Through the MQTT protocol, the agent publishes the control message(Figure 6 shows the characteristics of the control message such as message length and message topic length) which reaches the PLC where it is interpreted exactly.

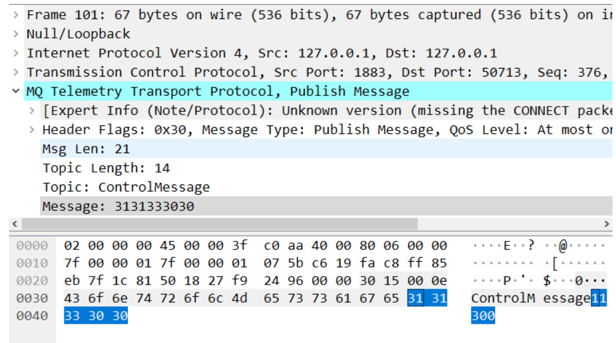

Fig. 6: Control Message characteristics

TABLE III: Control message

Character number	Meaning
0	Start/Stop motor
1	Conveyor direction
2	Motor speed
3	Upper piston command
4	Lower piston command

The control logic is realized at the control agent level and is implemented using the Typescript programming language. It is decided based on commands sent by the user from the control interface and the status of the assembly station. It mainly relies on the "if...else" statement to keep the program as simple and efficient as possible:

```
{
if (commandWord === "1") {
this.communicationManager.publishRaw(RawEvent.
    withTopicAndPayload(
"ControlCommands",
"10100",
{
retain: true,
}));
} else {
this.communicationManager.publishRaw(RawEvent.
    withTopicAndPayload(
"ControlCommands",
"00100",
{
retain: true,
}));
}
```

We further illustrate the functionality of the control agent which is able to retrieve information via MQTT about the current consumption of the motor at run time. An operating cycle was simulated in which the conveyor was started in one direction with a speed of 50%, after which it was changed to 25%, and in the last stage of the movement the speed was increased to 75%, at the end of which the blade carried on

TABLE IV: Control message

Character number	Meaning
0	Start/Stop motor
1	Conveyor direction
2	Motor speed
3	Upper piston command
4	Lower piston command

the conveyor reached the inductive sensor at the end of the conveyor. You can see the current consumption for each speed, but also that the motor needs more current when starting. This data can be further used in the control logic and in the predictive maintenance process. Results are shown in Fig. 7.

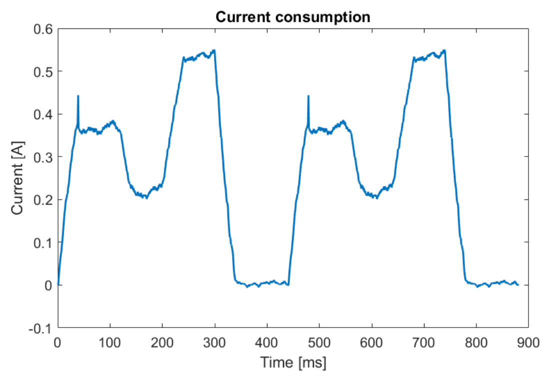

Fig. 7: Current consumption

To transmit commands to the driver, the PLC uses the PROFINET PTCP communication protocol. In addition to data transmission, it also synchronizes clocks and time between the RJ47 ports of the interconnected devices. In our configuration, the PLC sends one data package per second (Figure 8), which is sufficient for efficient engine control.

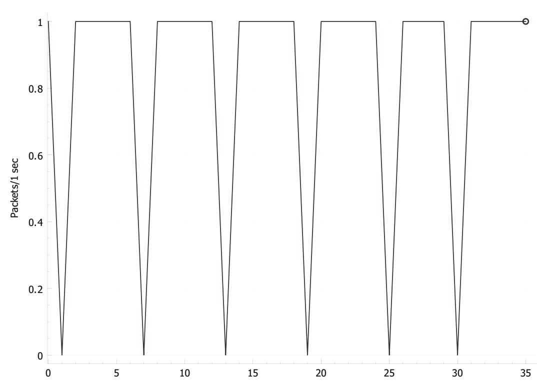

Fig. 8: Profinet packets

V. Conclusion

We presented a system architecture and associated application-oriented implementation of a decentralized industrial motion control system. The solution includes both software-based agents and proprietary commercial automation equipment that are integrated to achieve the desired functionality. Standards-based communication protocols such as OPC-UA and MQTT are employed as well as the open-source Coaty distributed framework. The functionality of the system is highlighted.

Future work will be focused on validating the approach at scale, mainly through the integration of a robotic cell in the practical set-up with dedicated software agents, and detailed analysis of the latency/scalability/security trade-off that such an approach imposes, in conjunction with variable production-relevant KPIs.

References

[1] S. Girs, S. Sentilles, S. A. Asadollah, M. Ashjaei, and S. Mubeen, "A systematic literature study on definition and modeling of service-level agreements for cloud services in iot," *IEEE Access*, vol. 8, pp. 134 498–134 513, 2020.

[2] M. Hajduk, M. Sukop, and M. Haun, "Multi-agent systems—terminology and definitions," in *Cognitive Multi-agent Systems*. Springer, 2019, pp. 1–9.

[3] M. U. Akhtar, M. H. Raza, and M. Shafiq, "Role of batch size in scheduling optimization of flexible manufacturing system using genetic algorithm," *Journal of Industrial Engineering International*, vol. 15, no. 1, pp. 135–146, 2019.

[4] S. K. Panda, M. Majumder, L. Wisniewski, and J. Jasperneite, "Real-time industrial communication by using opc ua field level communication," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 1143–1146.

[5] Y. Li, J. Jiang, C. Lee, and S. H. Hong, "Practical implementation of an opc ua tsn communication architecture for a manufacturing system," *IEEE Access*, vol. 8, pp. 200 100–200 111, 2020.

[6] H. Zhou, C. She, Y. Deng, M. Dohler, and A. Nallanathan, "Machine Learning for Massive Industrial Internet of Things," *arXiv e-prints*, p. arXiv:2103.08308, Mar. 2021.

[7] V. Luchian, G. Stamatescu, I. Stamatescu, I. Fagarasan, and D. Popescu, "Iiot decentralized system monitoring for smart industry applications," in *29th Mediterranean Conference on Control and Automation (MED)*, 2021.

[8] O. Chenaru, G. Stamatescu, I. Stamatescu, and D. Popescu, "Towards cloud integration for industrial wireless sensor network systems," in *2015 9th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, 2015, pp. 917–922.

[9] V. Mihai, C. Dragana, G. Stamatescu, D. Popescu, and L. Ichim, "Wireless sensor network architecture based on fog computing," in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2018, pp. 743–747.

[10] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, 2018.

[11] G. Stamatescu, I. Stamatescu, and D. Popescu, "Consensus-based data aggregation for wireless sensor networks," *Journal of Control Engineering and Applied Informatics*, vol. 19, no. 2, pp. 43–50, 2017.

[12] M. Gundall, C. Glas, and H. D. Schotten, "Introduction of an architecture for flexible future process control systems as enabler for industry 4.0," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 1047–1050.

[13] H. Shi, L. Niu, and J. Sun, "Construction of industrial internet of things based on mqtt and opc ua protocols," in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2020, pp. 1263–1267.

[14] P. Ferrari, E. Sisinni, D. Brandão, and M. Rocha, "Evaluation of communication latency in industrial iot applications," in *2017 IEEE International Workshop on Measurement and Networking*, 2017.

[15] M. Ghazivakili, C. Demartini, and C. Zunino, "Industrial data-collector by enabling opc-ua standard for industry 4.0)," in *14th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2018.

[16] P. Bellagente, P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, "Enabling profinet devices to work in iot: characterization and requirements," in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, 2016.