

Received November 13, 2021, accepted December 2, 2021, date of publication December 13, 2021, date of current version December 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3135432

Toward Industrial IoT: Integrated Architecture of an OPC UA Synergy Platform

CHANGDAE LEE¹, NAMHYEOK KIM², AND SEUNGHO HONG¹, (Senior Member, IEEE)

¹Department of Electronic Engineering, Hanyang University, Ansan 15588, Republic of Korea

²Research and Development Center, Nestfield Company Ltd., Ansan 705, South Korea

Corresponding author: Seungho Hong (shhong@hanyang.ac.kr)

This work was supported by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea Government (MSIP) (Development and Proof of Open Manufacturing Operation System with ICT Convergence) under Grant 2020-0-009.

ABSTRACT Recently, the manufacturing industry has revolutionized with ‘smart manufacturing’ and ‘Industry 4.0’, and with the development of the Industrial Internet of Things (IIoT), increasing number of physical objects can access network services. However, critical issues such as real-time communication, interoperability, and interlinking of internet and industrial devices are yet to be addressed. These issues arise from the problem of interconnectivity between smart manufacturing systems and non-semantic industrial equipment. Therefore, we referred to ISO/IEC 30141, a standard IoT reference architect document, to solve the interconnection problem and bridge the gap between the Internet and industry. Furthermore, we proposed a four-domain integrated architecture that divides the manufacturing system into user, cloud service, sensing & controlling, and device domains. The proposed architecture is validated by constructing the applicable industrial pick-and-place testbed for the manufacturing system. Thus, this testbed can serve as guideline to examine the state of the art of the OPC UA PubSub based protocol in addition to several other types of protocol approaches.

INDEX TERMS OPC UA, interoperability, interlinking, industry 4.0, manufacturing system, IIoT.

I. INTRODUCTION

With the development of modern information technologies, internet traffic has rapidly increased; more than 50 billion IP connections were projected for the year 2025 [1]. Hence, an Internet of Things (IoT) architecture based on modern computing models, such as Fog or Edge computing, has been recently proposed to reduce process delays and address the limitations of IP resources on the internet [2]–[4]. However, critical issues such as real-time communication, interoperability, and interlinking of internet and industrial devices are yet to be addressed [5]–[7].

Recently, the manufacturing industry has revolutionized with ‘smart manufacturing’ and ‘Industry 4.0’, and with the development of the Industrial Internet of Things (IIoT), increasing number of physical objects can access network services. However, a major challenge that complicates the active adoption of IIoT in smart manufacturing is the problem of Operation Technology (OT) - Information Technology (IT) integration between the shop floor and smart manufacturing

systems. This is caused by the problem of interconnectivity between smart manufacturing systems and non-semantic industrial equipment. Therefore, attempts to solve the interconnectivity problem through semantically expressible devices (e.g. OPC UA gateways/ connectors) in smart manufacturing are being actively made to solve these problems [8], [10].

A. BACKGROUND

Research and development efforts in terms of interconnection are underway to solve the problems encountered in internet and industrial networks, and the integration of these two networks has been proposed [6], [7]. This proposed integration requires considering two vital aspects: reliability for the internet and immediate response for the industrial network. For example, to block malicious intentions on the internet, traditional firewalls employ a Deep Packet Inspection (DPI)-based algorithm [12] that examines the packets up to the transport layer; however, this algorithm requires a long processing time. Although this long processing time is not critical for the internet, it can cause problems in systems that require an immediate response at critical times, such as safety detection systems. Thus, security and message transmission reliability

The associate editor coordinating the review of this manuscript and approving it for publication was Tiago Cruz^{1b}.

are more important than immediate response for the internet. The traditional Supervisory Control And Data Acquisition (SCADA) firewalls in SCADA systems of industrial networks only provide limited security for fieldbus protocols on IEC 61158 [13], because industrial networks require rapid processing for time-critical missions. Thus, in stark contrast to the internet, industrial networks require instant response at critical times and face important challenges, such as limited service. To ensure the effective utilization of these two types of networks with different characteristics, the interlinking concept has been developed.

B. RELATED WORK

Endeley *et al.* [8] suggested a solution featuring a set of communication primitives, a platform-independent type system, and an intermediate language. They implemented Open Platform Communications Unified Architecture (OPC UA) connector through OPC UA information model for the Virtual Automation Bus (VAB) Gateway and for communicating with Industry 4.0 applications. They also evaluated the overhead created by consolidation in terms of round-trip time and message size imposed by metadata for abstraction. However, while these authors have sufficiently introduced the interface between the existing OPC UA Server/Client model and HTTP technology, they have not considered the OPC UA PubSub model for asynchronous data processing.

Kannoth *et al.* [10] studied the middleware solution that implements OPC gateway to ensure interconnectivity and interoperability and enable effective communication between OPC UA Server/Client model and DDS model. They also tested the performance of the middleware in different scenarios using Raspberry Pi and software middleware connected to a private LAN and evaluated its operation by measuring response times and reliability.

In their study as well, OPC UA's PubSub Model was not used, and processing delay time occurred while OPC UA server exchanged data through OPC gateway and TCP-based Binary Protocol (BP) and converted it into UDP communication DDS data. However, in our approach, the OPC UA PubSub model is applied to minimize the processing delay time, and when the broker sends and receives messages from the gateway, the UDP-based UADP protocol is used instead of the TCP-based binary protocol.

Koziolek *et al.* [16] suggested a plug-and-produce architecture for Industrial Internet of Things (IIoT) and implemented it on open-source code-based system. However, the authors considered only the industrial side for interoperability and ignored the IoT protocol and internet security or reliability.

Gruner *et al.* [17] proposed a web platform that integrates the Representational State Transfer (REST) architecture to utilize the internet for solving the interoperability problem between resource-constrained devices in industrial networks using the OPC UA [18]. Although these authors sufficiently introduced the interface between OPC UA and the REST technology for processing synchronous data, they neither considered it for industrial networks nor demonstrated the

implementation of the OPC UA PubSub interface with middleware for processing asynchronous data.

Luo *et al.* [9] suggested using the OPC UA technology and applied the Industrial Internet Reference Architecture (IIRA) with a three-tier structure proposed by the Industrial Internet Consortium (IIC) to a smart manufacturing system.

IIRA [11] proposed by IIC is a three-tier structure. The first enterprise tier implements domain-specific applications with interfaces to end-users, and the second platform tier receives, processes, and forwards data. The final edge tier collects data from edge nodes using a proximity network. However, the architecture proposed in this study refers to ISO/IEC 30141, a standard IoT reference architecture document, and proposes a four-domain integrated architecture that divides the manufacturing system into user, cloud service, sensing & controlling, and device domains.

Both architectures have similar structures for applying Industrial IoT applications to smart manufacturing systems and utilize industrial protocols and OPC UA technology that value real-time; however, the OPC UA PubSub model for asynchronous data processing was not considered in their study.

Regarding analysis of the application-level latency with OPC UA technologies, some studies have compared IoT protocols [19]–[23] other than the Unified Architecture Datagram Protocol (UADP)/Advanced Message Queuing Protocol (AMQP) [24] PubSub. These studies have provided not only an overview of the different features of OPC UA technology but also Constrained Application Protocol (CoAP) [15] and Message Queuing Telemetry Transport (MQTT) [25] and compared their performance against benchmarks.

C. MOTIVATION

Open Platform Communication (OPC) Foundation recently defined a new specification (part 14) for extending the PubSub method to ensure interoperability when applying IoT technologies to industrial networks. However, a concrete concept about how to link the two different technologies of the brokerless model (e.g., UADP PubSub) for industrial protocols and the broker model (e.g., AMQP PubSub) for IoT protocols is still missing. This missing concept can be crucial for addressing the different integrating data in the field of Industry 4.0, which is a cutting-edge technology that is currently being studied. In addition, there has been no case in which the latest OPC UA technologies incorporating the OPC UA PubSub between the internet and industrial networks has been used because the OPC UA specification has recently defined IoT communication. Therefore, this study aims to propose an integrated architecture, divided into user, cloud service, sensing & controlling, and device domains, and develop an OPC UA synergy platform concept to interlink server, broker, and brokerless models. To the best of our knowledge, compared with the existing studies, our contributions can be summarized as follows:

- First, unlike existing works, this paper takes a step forward to fill the gap between the proposed architecture

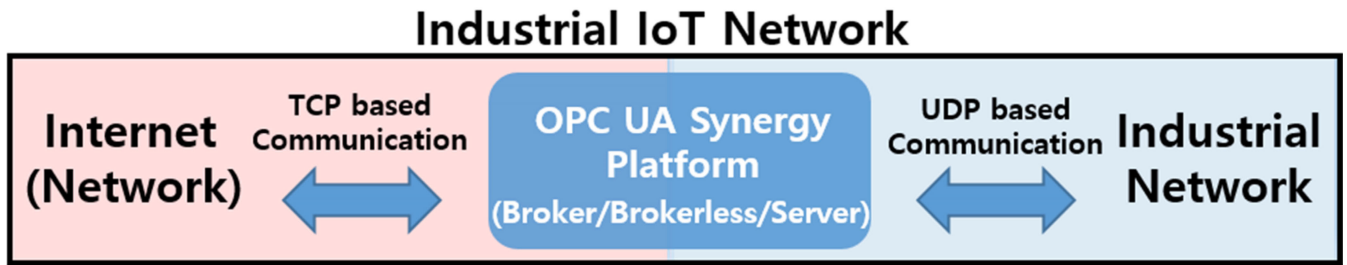


FIGURE 1. OPC UA synergy platform diagram.

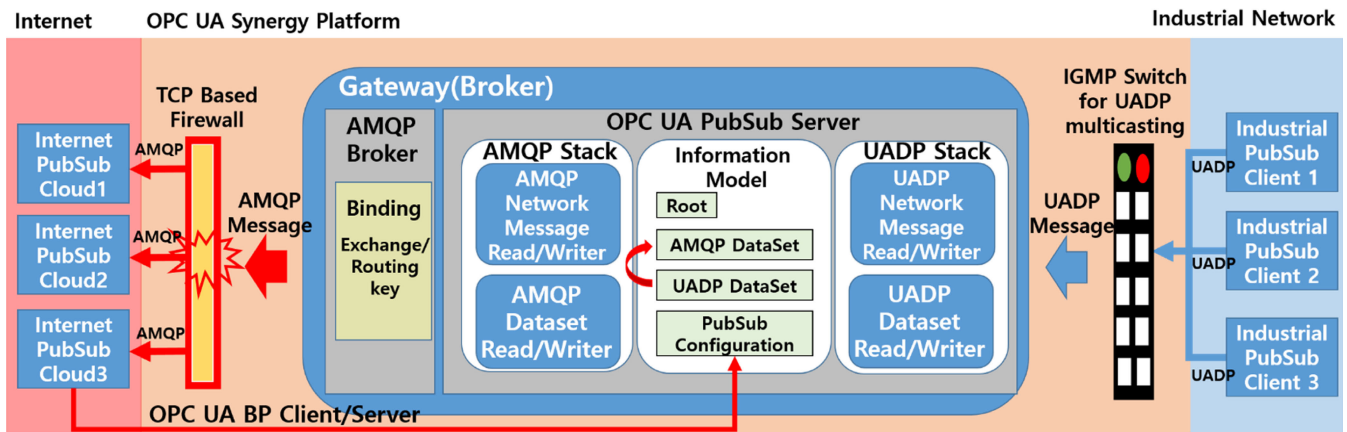


FIGURE 2. OPC UA synergy platform diagram.

and practical implementations by providing guidelines on how to demonstrate an OPC UA synergy platform concept through an experimental setup.

- Second, in this pioneering work, the integrated architecture for Industrial IoT is proposed to bridge the gap between the internet and industrial side for the interlinking concept.
- Third, the applicable industrial pick-and-place testbed for the manufacturing system is constructed by specifying the details of various protocols, over which the real-time delay was measured, and the OPC UA synergy platform is validated.
- Fourth, the established testbed can serve as a guideline to examine the state of the art of the OPC UA PubSub based protocol, and several other types of protocol approaches.

To summarize, the objective of this work is to explore the applicability of the synergy platform, the proposed integrated architecture, and the testbed, to verify our analysis of this advanced interlinking concept.

II. INTEGRATED ARCHITECTURE OF THE OPC UA SYNERGY PLATFORM

Fig. 1 illustrates the concept of the OPC UA synergy platform, which is divided into broker, brokerless, and server models based on IEC 62541 (OPC UA) [18], [26]–[30]. The broker and server models aim at transmitting reliable

messages based on Transmission Control Protocol (TCP) communication on the internet, whereas the brokerless model guarantees a rapid response based on User Datagram Protocol (UDP) communication in an industrial network. The OPC UA synergy platform comprises the OPC UA client/server and OPC UA PubSub pattern for transmitting messages between the two networks. This platform also plays an important role in interlinking the domains (for the internet and industrial network) of the manufacturing system architecture. The synergy platform refers to a synergy model that uses the Server/Client and the PubSub model at the same time, and it is a term officially used in the OPC UA spec document [18], [30].

Fig. 2 shows the detailed structure of the OPC UA synergy platform, which is divided into external internet PubSub cloud clients, gateway (broker), Internet Group Management Protocol (IGMP) switch, and internal industrial PubSub clients.

The external internet PubSub cloud (the cloud client) can asynchronously communicate with the OPC UA PubSub server through the AMQP broker from the out-of-TCP-based firewall. The AMQP broker sends messages following the exchange type & routing key to the internet PubSub cloud, as depicted in Fig. 2.

The OPC UA PubSub server converts the message to the AMQP dataset and transmits it to the AMQP broker.

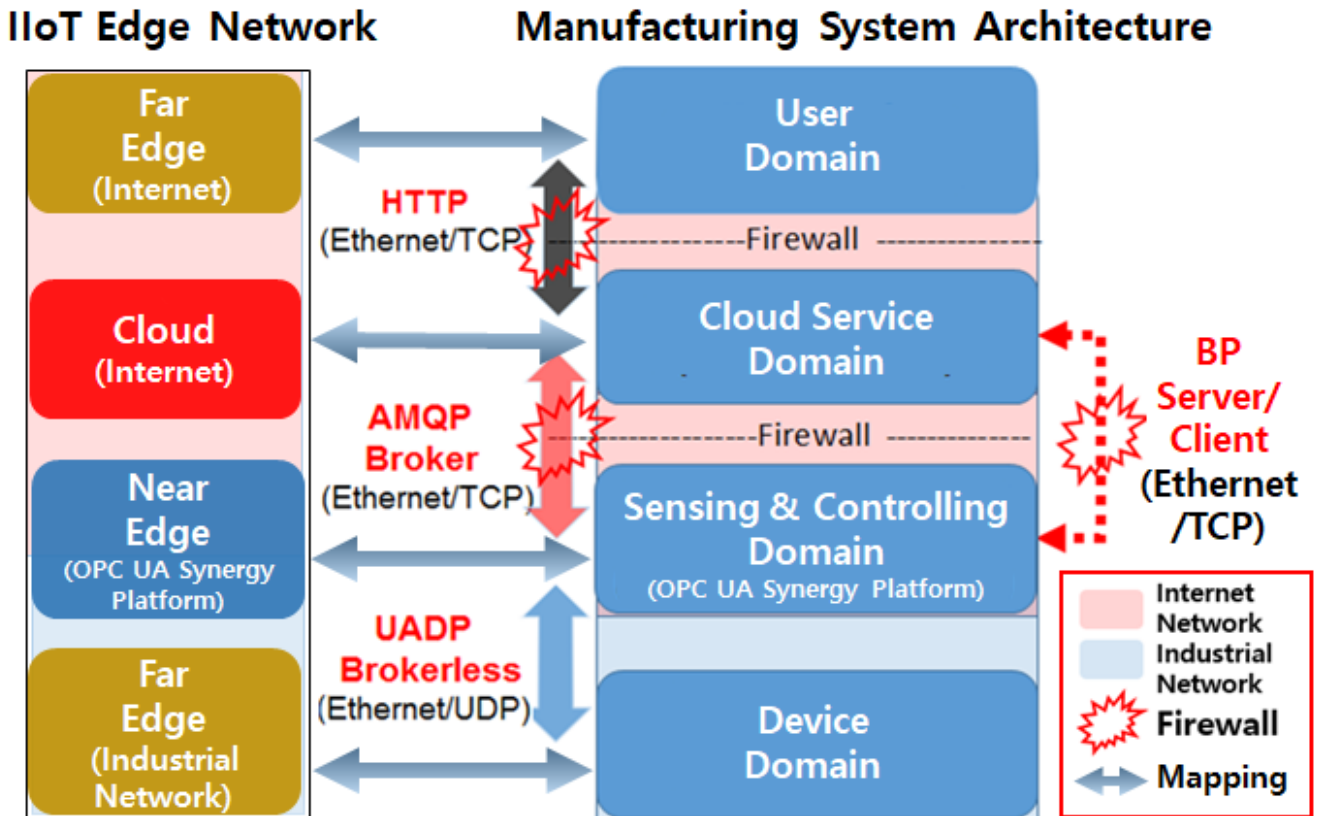


FIGURE 3. Integrated architecture for the internet and industrial network of the OPC UA synergy platform.

The PubSub stack defines a function called `Network_Message_Reader/Writer`, which verifies the data type (whether it is a UADP or AMQP message) and can read/write in the dataset format. The number of internal industrial PubSub clients can be easily increased by connecting to the multicasting band, which rapidly multicasts messages to the gateway through the IGMP switch.

For instance, Fig. 2 shows that the message from the industrial PubSub client is converted into the OPC UA dataset through the UADP PubSub stack, and the dataset is then stored in the OPC UA address space. Thus, TCP-based internet and UDP-based-industrial networks can be interlinked through this platform.

However, the OPC UA PubSub server needs to set the initial configuration from the user through the BP, as shown in Fig. 2. This initial configuration must mainly set the UDP multicasting destination address, the AMQP broker's destination address, and the interval cycle. In other words, the BP is used to communicate between the OPC UA server and the client pattern. The OPC UA PubSub server in the gateway contains an information model, including configuration, and the OPC UA client from outside the firewall can connect to this server to set the initial configuration.

The OPC UA Synergy Platform comprises an OPC UA PubSub server with the UADP & AMQP PubSub stack based

on open62541, developed by the authors, and the AMQP Broker version 0.9.2.

III. PROPOSED INTEGRATED ARCHITECTURE AND IMPLEMENTATION TESTBED

In this section, we describe the proposed integrated architecture of the OPC UA synergy platform. The proposed architecture was developed by us and referenced from the IoT reference architecture in ISO/IEC 30141 for applying the internet to the industrial network [31]. The proposed integrated architecture is depicted in Fig. 3 as the IIoT network of the OPC UA synergy platform and the manufacturing system architecture.

The manufacturing system architecture is divided into four domains, user, cloud service, sensing & controlling, and device, which can be mapped to the IIoT edge network.

In general, the concept of Far/Near Edge can be applied to IIoT edge networks. In the Far/Near Edge concept, the Far Edge can be mapped to an end-user and device domain in the manufacturing system which provides services or has been served from the cloud. Furthermore, in the IIoT edge network, the cloud can be mapped to the Cloud Service Domain in the manufacturing system. Near Edge is responsible for distributing services to be properly provisioned between users and the Cloud, which is where the OPC UA Synergy Platform plays

a role. This is where the sensing & controlling domains in manufacturing systems play this role.

The sensing & controlling domain is mapped to the OPC UA synergy platform to process large amounts of data between cloud servers and industrial devices. Here, the sensing and control domain connects the cloud services and device domains, and the OPC UA synergy platform acts as the OPC Gate or Connector between the AMQP side of the internet and industrial networks, solving the problems of the different interconnection of the two networks.

The roles and descriptions of each domain in the manufacturing system are as follows.

A. ROLES OF THE INTEGRATED ARCHITECTURE DOMAINS

1) USER DOMAIN

This domain defines the user as a customer receiving services from the factory. The user can easily access the monitoring and configuration information from anywhere with an internet connection, but not directly access the internal factory. The user indirectly requests service from the cloud service domain and delivers their configuration parameters to flexibly manage the factory.

2) CLOUD SERVICE DOMAIN

The cloud service domain directly connects to the internal factory and provides a monitoring service to users based on an internet protocol (e.g., HTTP/HTTPS) and internet security (e.g., TCP-firewall) to block malevolent users from accessing clouds. The domain consists of front-end servers for providing the user interface service and a back-end client for monitoring the factory information. The back-end clients require the IoT protocol (e.g., AMQP/MQTT) to stably obtain large amounts of data and multiple access for connecting directly to the many assets; thus, this domain should provide interconnection and fast processing to satisfy real-time requirement.

3) SENSING & CONTROLLING DOMAIN

The factory needs a network processing machine for the sensing & controlling domain to adjust the traffic between the OT and internet technology. For example, the IGMP switch is necessary for using multicasting for OT (e.g., UADP) and the Broker to using multicasting for IT (e.g., AMQP). This domain also plays the role of rapidly processing real-time data from multiple connections of the industrial devices and simultaneously replying to requests from the servers in the cloud domain. Thus, the OPC UA synergy platform can be included in this domain to bridge the cloud service and device domain as shown in Fig. 3.

4) DEVICE DOMAIN

The device domain includes industrial equipment used for implementing factory scenarios. For example, the pick-and-place scenario comprises entities such as robot arms, conveyor belts, and a controller device. When the

activating signal for the end-devices from the sensing & controlling domain arrives in the device domain, all the industrial equipment operate automatically in accordance with the pre-ordered scenario.

B. COMMUNICATION PROTOCOLS BETWEEN INDUSTRIAL IoT DOMAINS

1) HTTP/HTTPS OF USER AND CLOUD SERVICE DOMAINS

The user and cloud service domains require TCP-based communication for reliability purposes, when using the internet. In TCP-based communication, a relatively long delay has no critical impact on the user; hence, TCP-based communication is more reliable and consequently the preferred type of data communication. HTTP is widely used as an internet protocol to ensure data reliability. HTTP default ports are also usually open in firewalls, which is an advantage over other internet protocols. HTTPS is added for secure socket function because HTTP has no security; thus, we recommend using the HTTP added secure socket (HTTPS) for communication between the user and cloud service domains.

2) AMQP/BINARY PROTOCOL OF THE CLOUD SERVICE AND SENSING & CONTROLLING DOMAINS

For communication between the cloud service and sensing & controlling domains, not only the interconnection problem of accessing clouds but also using the firewall with an internet connection should be considered. The AMQP and OPC UA PubSub protocol enables the transmission of considerable and secure data through middleware; thus, we recommend the OPC UA synergy platform to provide an information model for the AMQP/UADP connection. This platform connects from the cloud domain to the sensing & controlling domain via a BP client/server as depicted in Fig. 3, configures the AMQP PubSub connection at the cloud domain side, and then establishes the AMQP connection between these two domains. The AMQP broker at the sensing & controlling domain side can ensure secure and reliable interconnection for cloud servers.

3) UNIFIED ARCHITECTURE DATAGRAM PROTOCOL OF THE SENSING & CONTROLLING AND DEVICE DOMAINS

As shown in Fig. 3, when the upper side communicates with the sensing & controlling domain, the TCP based communication is utilized; however, for the lower side communication, sensing data should be quickly aggregated from the industrial device, and messages should be concurrently sent to the equipment. The UADP based on the brokerless model ensures a rapid response and asynchronous communication as a multicast protocol for OT. The UADP also requires specific switches and shifts the work of distributing the messages to all subscribers into the network through the IGMP switches; thus, this protocol is recommended to ensure the processing of fast UDP communication on the lower side. Meanwhile, the device domain comprises various industrial protocols

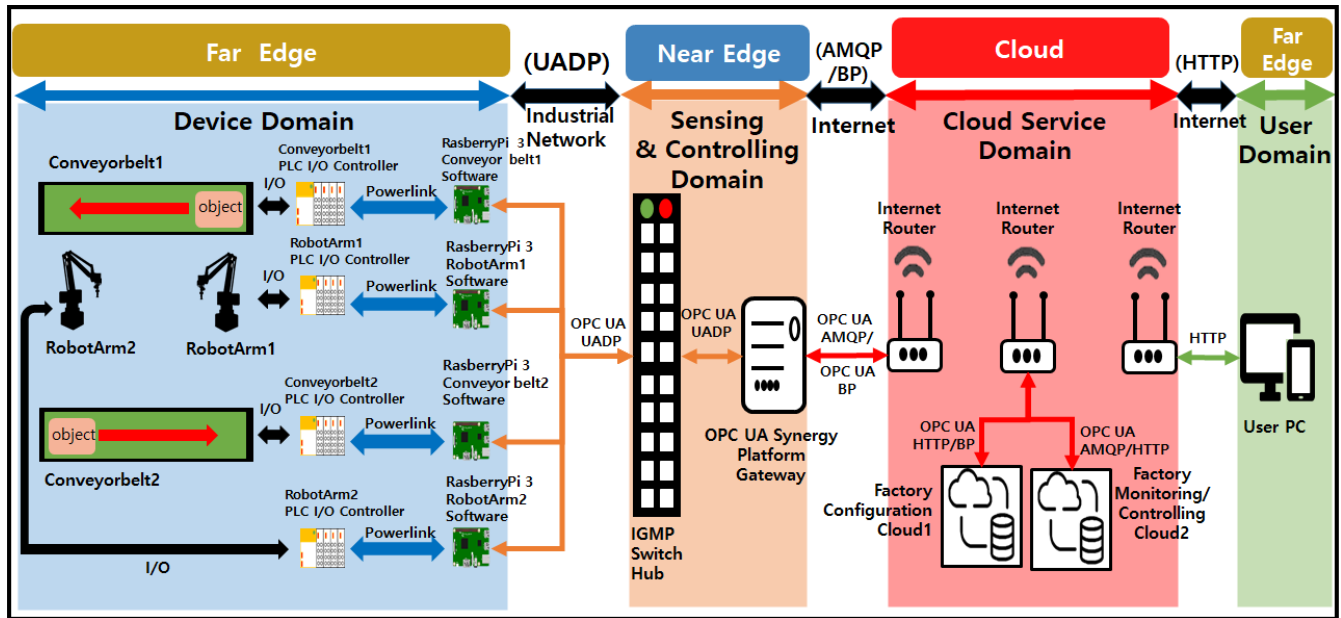


FIGURE 4. Diagram of the robot and conveyor belt system used as a usecase.

that can be supported by manufacturers. These protocols are real-time based industrial protocols that respond immediately at critical times, and the industrial devices can communicate with each other via OPC UA. If industrial devices do not provide OPC UA function, the OPC UA stack for UADP functionalities can be used to easily connect to the OPC UA synergy platform.

An important feature of the proposed architecture is that it guarantees interoperability and interconnectivity among heterogeneous devices by using OPC UA from the OPC foundation in conjunction with multiple vendors. Moreover, OPC UA provides a useful function for information models and communication services for strong interoperability solutions. Another key feature of this integrated architecture is its distributed structure and asynchronous operation from the user to the end-device, are divided into four domains. This advantage simplifies the complex life cycle and extends the flexibility of the system; thus, it enhances system management efficiency. Additionally, asynchronous operation of the four industrial domains solves the explosive traffic processing problem and provides more large-scale services, such as big data processing.

C. CASESTUDY: THE PICK AND PLACE SYSTEM

In this section, the pick-and-place system is demonstrated as a case study of the proposed architecture to validate the various performances. The system mapped the proposed architecture and implemented it to illustrate the pick-and-place scenario. Fig. 4 presents a diagram of the robot and conveyor belt system used as a case study. Fig. 5 shows the real implemented testbed, which can be divided into four domains and three networks.

1) USER DOMAIN WITH HTTP

As shown in Fig. 4, in the user domain, the user can be any entity that seeks monitoring services, such as coordination of a robot position, operation status of a device, or configuration service of adjusting the message period or speed through HTTP. There are typically several HTTP clients, and the user can choose a web browser that supports HTTP.

2) CLOUD SERVICE DOMAIN WITH AMQP AND BP

In the cloud service domain, to ensure that distributed clouds can communicate with each other, we designed two cloud servers supporting HTTP and three routers for internet connection. The first cloud server was used for the factory monitoring/controlling service; the second server was for the factory configuration service. The server for factory monitoring/controlling requests factory information from the OPC UA synergy platform through the AMQP of the OPC UA PubSub interface. When the configuration message from the user (e.g., period, speed, IP address information) is delivered to the server for the factory configuration, the server sends the information to the OPC UA synergy platform through the BP of the OPC UA client/server interface.

3) SENSING & CONTROLLING DOMAIN WITH OPC UA

As shown in Fig. 4, the sensing & controlling domain comprised the OPC UA synergy platform and IGMP switch for multicasting. The OPC UA synergy platform included the UADP, AMQP stack, and BP client/server stack, which is assumed to bridge the sensing & controlling domain with the cloud and device domains. Meanwhile, the OPC UA synergy platform subscribed to the UADP messages from four Raspberry Pi 3 computers through the industrial network, as shown

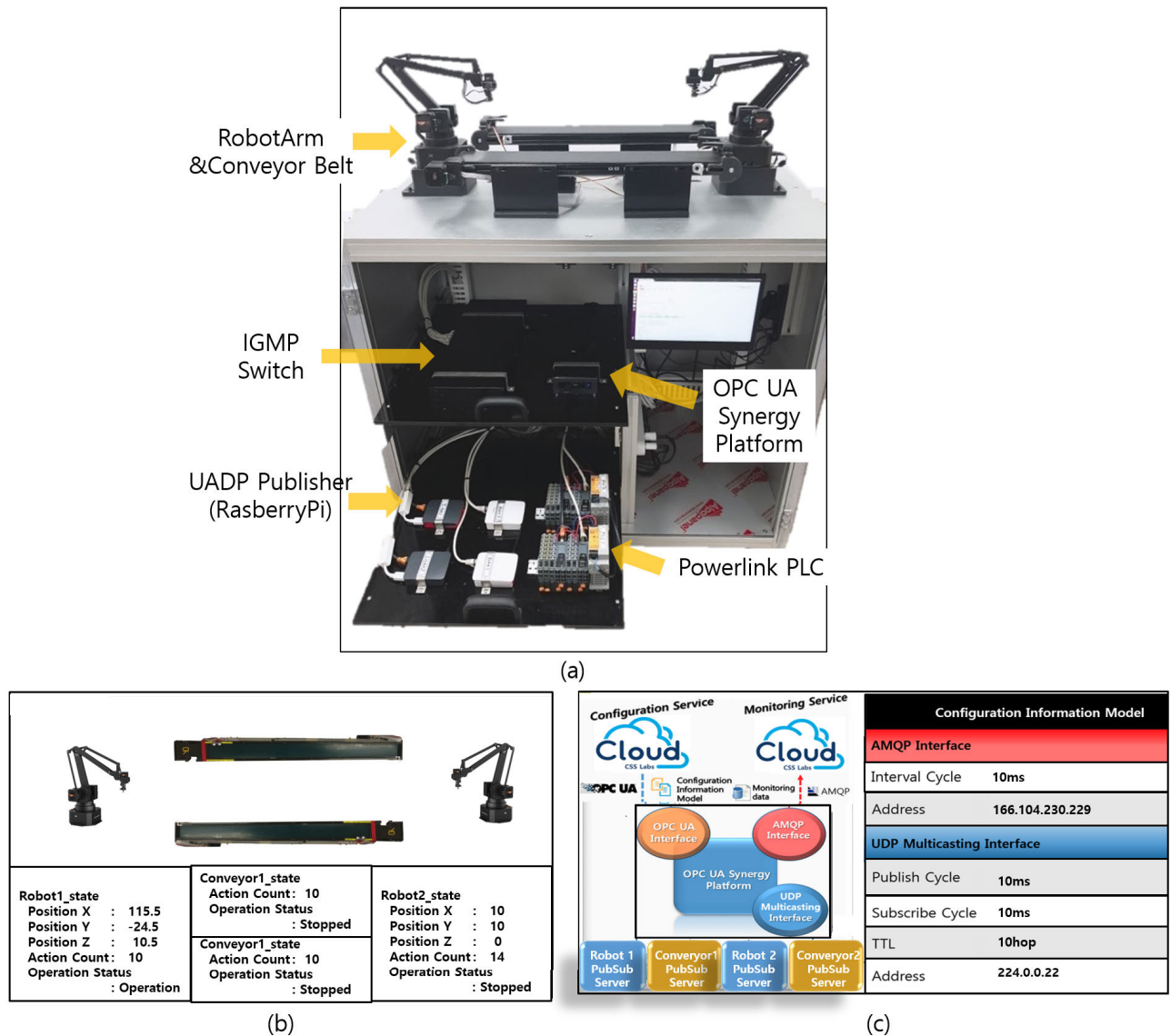


FIGURE 5. Implemented testbed. (a) Pick-and-place demo system; (b) controlling/monitoring cloud service; (c) configuration cloud service.

in Fig. 4. This platform also converts the UADP messages into AMQP datasets to transmit them to the external internet simultaneously.

4) DEVICE DOMAIN WITH UADP AND POWERLINK

The device domain comprised two conveyor belts, two robot arms, four PLC I/O controllers supporting Powerlink [33], and four Raspberry Pi 3 Model B computers for the robot/conveyor operating software, including control logic. Note that there are no available OPC products that support Powerlink and UADP communication at the same time, considering this, we have developed a simple UADP & Powerlink stack in Raspberry Pi 3 computers, such that these computers can be used for connecting to the OPC UA synergy platform and Powerlink network. As shown in Fig. 4, the Raspberry Pi 3 computers publish monitoring data to the OPC UA synergy platform through UADP. Additionally, these were

designed to send control signals for the robots/conveyor belts to each PLC I/O controller via the Powerlink network. After receiving the control signal from the PLC I/O controller, the two conveyor belts or robots operate according to the pick-and-place scenario.

The robot receives the completion signal after the conveyor belt finishes its job; when the conveyor belt does not stop completely, the robot arm will not operate to pick up the object, and vice versa. When the setup is ready, the robot picks up and places the object on the opposite side, and the conveyor belt starts the delivery, as shown in Fig. 4.

5) IMPLEMENTATION OF THE TESTBED

Fig. 5(a)–(c) illustrates the implemented pick-and-place system and a User Interface (UI) display for two cloud servers. As shown in Fig. 5(a), the system used four Raspberry Pi 3 Model B computers as the UADP publisher, together with

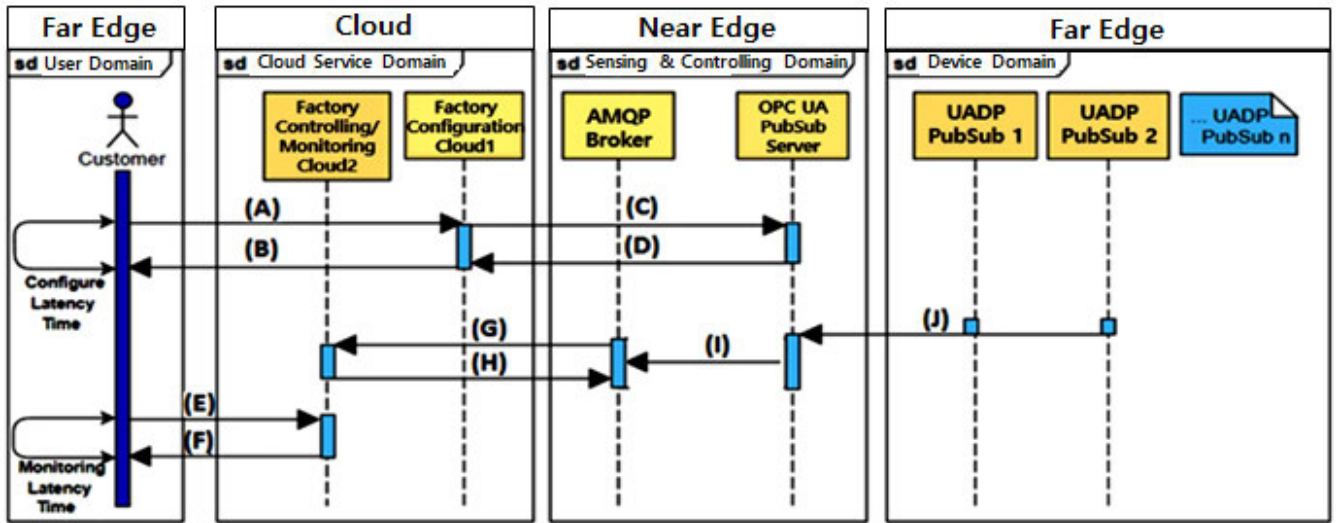


FIGURE 6. Operating sequence of the proposed architecture. The configuration latency of the: (A) HTTP request-get method, (B) HTTP response—status code 200, (C) OPC UA request—ReadRequest, and (D) OPC UA response—ReadResponse. The monitoring latency of the (E) HTTP request-get methods, (F) HTTP response—status code 200, (G) OPC UA (AMQP) subscribe—consume, (H) OPC UA (AMQP) ACK—consume-ok, (I) OPC UA (AMQP) Publish—deliver, and (J) OPC UA (UADP) publish—UADP message.

two PLC X20CP1584 and two X20BC0083 sets as the Powerlink I/O module. The system also included an ipTIME SW2400-mini 24 for the IGMP switch and Intel NUC Kits NUC8I3BEH as the OPC UA platform gateway. Two six-axis robot arms and two conveyor belts made by Ufactory were used as the end-devices. The real-time state values of the robot and conveyor belt operation and the coordinates of the target point obtained from the industrial network are shown in Fig. 5(b). Fig. 5(c) depicts the configuration servers and the setting of the address, speed, and Time To Live (TTL) of AMQP and UDP multicasting for the OPC UA synergy. Two Lenovo Think Centre M720 Tiny PCs were used as the servers, which are used to present the web content for users through the web application server.

In the Fig. 5, the robot and conveyor belt are connected to industrial PLC equipment. The PLC unit also communicates with each of the four Raspberry Pi modules including the robot software via Powerlink, an industrial protocol for real-time applications. Here, the powerlink is ethernet-based communication and has three characteristics:

- Configurable response times ensure time-critical data transmission with very short isochronous cycles.
- Synchronize time of all nodes in the industrial network with very high sub-microsecond precision.
- Less important data transfer on scheduled asynchronous channels.

Therefore, the real-time and synchronization can be maintained between PLC and Raspberry Pi through powerlink protocol. In addition, the UADP-based protocol using OPC UA is used between Raspberry Pi and OPC UA Synergy Platform gateway. Here, the chrony tool, a system synchronization tool based on Network Time Protocol (NTP), was used between the Raspberry Pi and the Synergy Platform gateway.

The chrony tool supports clock synchronization of each system through hardware timestamps, which enables system clock synchronization of all Raspberry Pis and Synergy Platform gateway.

IV. RESULTS

In this section, a sequence diagram is used to describe the operation of the proposed architecture. We also present the performance evaluation results for the four domains of the proposed architecture. The application-level latency is then compared with respect to each communication protocol by varying the payload sizes.

A. SEQUENCE OF OPERATING TASKS

The sequence of the operating tasks for the proposed architecture is shown in Fig. 6. The customer in the user domain can request the monitoring or response data to set the initial configuration of the OPC UA synergy platform. The cloud service domain includes two cloud servers: one for the initial configuration and the other for controlling/monitoring the industrial data. As shown in Fig. 6, the process (A–D) ranged from the user domain to the sensing & controlling domain, especially for the initial configuration of the first cloud, namely “Factory Configuration Cloud1.” This server can manage the configuration (e.g., message speed, period, and IP address of each industrial device) of the OPC UA synergy platform. “The factory configuration cloud1” is asynchronously processed through the HTTP between the user and cloud service domains, whereas the OPC UA PubSub server in the sensing & controlling domain is synchronously processed through the OPC UA client/server pattern using the Cloud1 server. The user and cloud service domains were

TABLE 1. Measurement range and latency time for the four domains.

Target Protocol	Measurement Range	Latency Time
HTTP	User domain to Cloud Service Domain	(B) – (A)
BP (OPC UA C/S)	Cloud Service domain to Sensing & Controlling Domain	(D) – (C)
AMQP (OPC UA P/S)	Cloud Service domain to Sensing & Controlling Domain	(H) – (G) + (I)
UADP (OPC UA P/S)	Sensing & Controlling Domain to Device Domain	(J)

*BP: Binary Protocol

*UADP: Unified Architecture Datagram Protocol

configured through the HTTP post/get method and Acknowledgement (ACK) signal (A and B).

The OPC UA PubSub server provided the ReadRequest and ReadResponse of the Remote Process Call (RPC) service to the cloud, as shown in Fig. 6 (C and D). The configure latency time depicted in Fig. 6 was obtained by summing the difference between (A) and (B), and that between (C) and (D). Fig. 6 illustrates (E–J) that the monitoring latency ranged from the user domain to the device domain, and that the monitoring/controlling data of the second cloud (Factory Monitoring/Controlling Cloud2) were processed. In this process, the customer receives the monitoring data between the cloud service and the sensing & controlling domains via the HTTP and OPC UA PubSub pattern. As shown in Fig. 6 (G)–(J), the OPC UA PubSub server receives the industrial dataset from the device domain and converts it into the AMQP dataset, which is sent to the AMQP broker. When this dataset reaches the AMQP queue, the AMQP broker transmits the dataset to the subscriber (Cloud2 server). When the device domain contains additional industrial equipment, UADP PubSub is added, as shown in Fig. 6 (J), and data are automatically published to the multicasting band. In this process, Cloud1 provides configuration service for the PubSub model, whereas Cloud2 monitors the transmission of a large number of messages and guarantees the interconnection of the end-devices. The total monitoring latency time, shown in Fig. 6, was obtained by summing the difference between (E) and (F), that between (G) and (H), and the multicasting publishing delays (I) and (J).

B. DATA LATENCY IN THE PROPOSED ARCHITECTURE

For the operation sequence diagram (Fig. 6), the target latency protocols and measuring range of each domain are shown in Table 1.

The table presents four measuring targets, including each main protocol of the domain: the HTTP protocol between the user and cloud service domains, BP based on the OPC UA

client/server model, AMQP based on the OPC UA PubSub model between the cloud service and sensing & controlling domains, and the OPC UA PubSub model of UADP between the sensing & controlling and device domains.

In the case of traffic generation, we implemented a user application program that periodically transmits a packet with a transmission/reception cycle of 10ms and a payload size of 256 to 16384 bytes. The user application program was implemented using open62541, which is an open-source resource for implementing OPC UA, and AMQP Stack of Rabbitmq project and HTTP Stack.

The Wireshark measurement tool was used to compare the latency at the receiver side. For every conducted experiment, each message transmission/reception period of the four-domain was 10 ms, and the payload size ranged from 256 to 16384 bytes. This latency measurement experiment was conducted ten times and gathered from a minimum of 5872 packets to a maximum of 11168 packets during a half-minute.

Fig. 7 illustrates the measurement results of application-level protocol latency, which is the key result obtained from Table 2. The HTTP browser from the user domain to the Factory Monitoring/Controlling Cloud2 send the message request through the internet and vice versa; Fig. 7(a) depicts the round-trip time with HTTP latency between User domain and the configuration cloud1 server; this maximum latency time had a delay of 37.97 ms with a mean time of 11.90 ms at 16384 bytes, and the retransmission of HTTP was measured a maximum of 48 times at 16384 kb by Wireshark. HTTP exhibits high reliability communication using retransmission when data are lost during transmission, or a problem occurs with the ACK value [34]. However, this retransmission has a significant impact on latency and occurs frequently as the payload value increases, which also results in a large average delay (Fig. 7a); thus, the latency result also demonstrates that among all TCP protocols, HTTP is most likely to be affected by payload and has the slowest response time.

Fig. 7(b) illustrates the latency time for the AMQP of the OPC UA PubSub model, indicating a minor effect on payload size compared with HTTP. Similarly, AMQP exhibited less fluctuating delay as the payload size increased; this protocol had a maximum latency of 3.85 ms and an average time of 3.37 ms at 16384 bytes. Fig. 7(c) depicts the RTT of BP based on the OPC UA client/server model, which exhibited the most monotonic and fastest response time compared with the other TCP protocols. This protocol had a maximum latency of 2.15 ms and an average time of 1.91 ms. In TCP networks, the maximum value of the TCP bandwidth is guaranteed when the Receive Windows Size (RWND) and Transmission Windows Size (TWND) are the same [35]; when these sizes differ, the delay time affects the maximum bandwidth and creates latency. Meanwhile, the TWND size in this experiment was fixed because it does not change the sender buffer size, whereas the RWND size depends on the windows scale option adjusted by the receiver. Likewise, the three protocols (HTTP, AMQP, and BP) of TCP applications shown in Fig. 7 exhibited application-level delays in all TCP

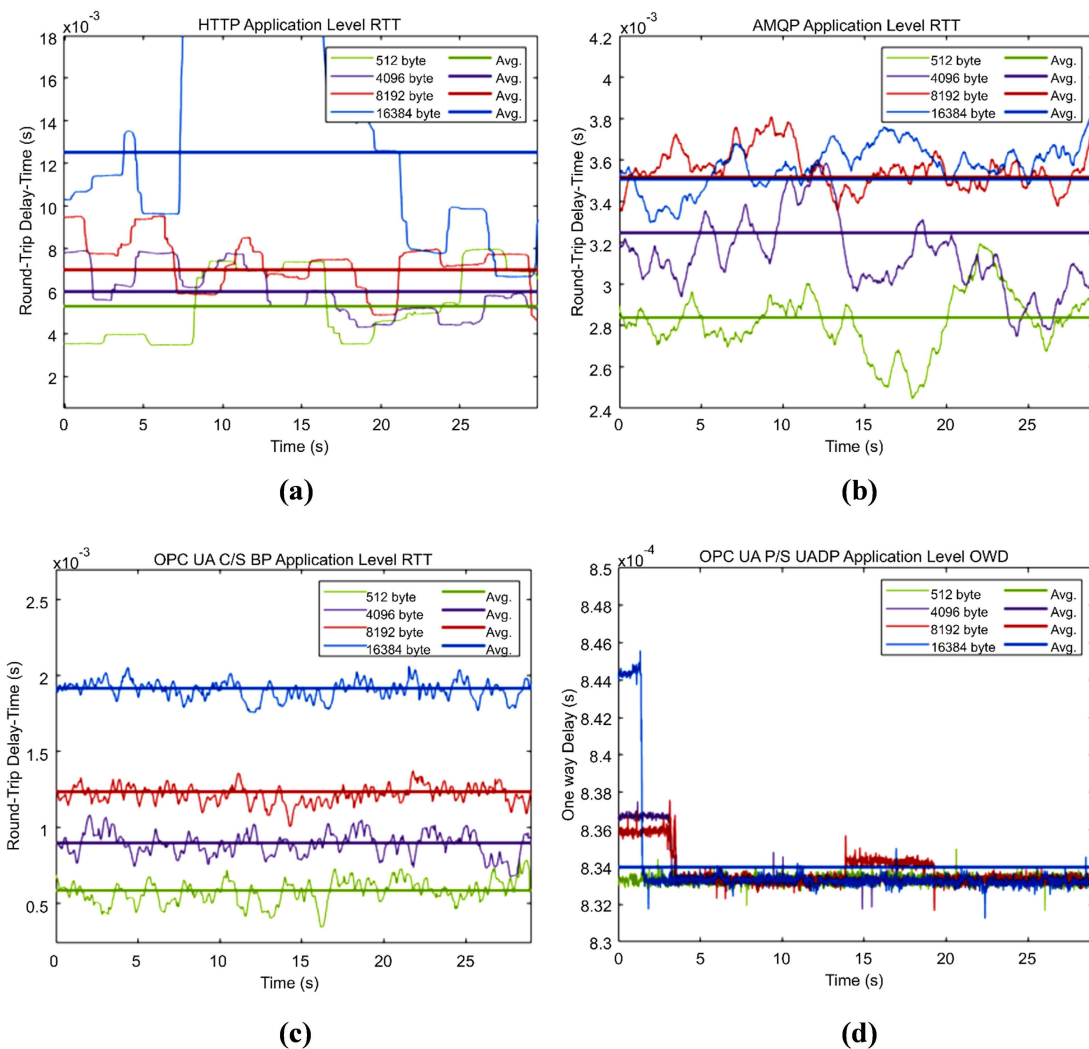


FIGURE 7. Measurement results for round-trip delay over time obtained using the proposed approach: (a) HTTP get/state RTT; (b) OPC UA P/S AMQP RTT; (c) OPC UA C/S BP RTT; (d) OPC UA P/S UADP OWD.

TABLE 2. Application-level latency with various payloads, as shown in Fig. 7.

byte	HTTP APPLICATION LEVEL RTT			OPC UA P/S AMQP APPLICATION LEVEL RTT			OPC UA C/S BP APPLICATION LEVEL RTT			OPC UA P/S UADP APPLICATION LEVEL OWD		
	Min. (ms)	Max. (ms)	Avg. (ms)	Min. (ms)	Max. (ms)	Avg. (ms)	Min. (ms)	Max. (ms)	Avg. (ms)	Min. (ms)	Max. (ms)	Avg. (ms)
256	0.0031	5.675	4.227	0.0028	4.010	3.021	0.0059	0.7946	0.571	0.0017	0.8377	0.7995
512	0.0034	7.985	4.885	0.0048	3.270	2.730	0.0058	0.7957	0.586	0.0016	0.8373	0.7993
1024	0.0034	6.155	4.193	0.0052	3.749	3.002	0.0061	0.8795	0.6433	0.0018	0.8373	0.7994
2048	0.0037	6.512	4.254	0.0046	4.120	3.288	0.0062	0.9086	0.71	0.0016	0.8431	0.7998
4096	0.0030	8.105	5.541	0.0052	3.889	3.065	0.0127	1.207	0.899	0.0014	0.8394	0.7998
8192	0.0042	9.523	6.500	0.0050	3.871	3.378	0.0163	1.422	1.237	0.0018	0.8376	0.7996
16384	0.0743	37.97	11.900	0.0085	3.846	3.371	0.0292	2.147	1.917	0.0016	0.8492	0.8003

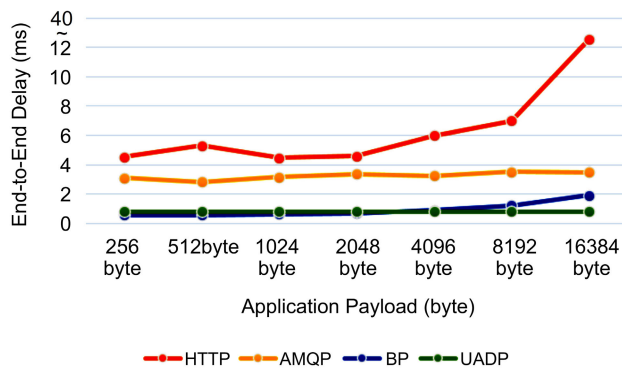


FIGURE 8. Comparisons of average latencies on the application level among four difference protocols.

measurements because of the different RWND and TWND sizes. The UDP application had a negligible effect except for the initial fluctuation times required for the address resolution in the IGMP switch, as shown in Fig. 7(d). Fig. 7(d) illustrates the One-Way Delay (OWD) [36] from UADP of the OPC UA PubSub model. Unlike a TCP network, UADP had minimal influence on latency regardless of the payload size and was only affected by the transmission cycle of PubSub. This protocol had a maximum latency of 0.85 ms and an average time of 0.80 ms at 16384 bytes.

Overall, UADP achieved the fastest response among all protocols; precisely, it was 45 times faster at 16384 bytes compared with the slowest HTTP, as depicted in Fig. 8. BP was 1.6 times faster at 16384 bytes compared with AMQP; On the other hand, BP was 2.5 times slower than UADP 16384 bytes. According to the experiment, UADP, based on OPC UA PubSub, was suitable for the device domain because it requires minimal delay, rapid response, and processes large amounts of data. Finally, HTTP had a relatively longer delay compared with other protocols; however, it is appropriate to be applied for user domain owing to its high reliability.

V. CONCLUSION

In this study, we proposed an integrated architecture divided into four domains that can simultaneously provide reliable internet and serve an industrial network requiring a rapid response. The OPC UA synergy platform plays an important role in integrating two different networks and was designed to ensure interoperability between the two networks and the transmission of substantial amounts of data. The proposed integrated architecture of the OPC UA synergy platform is expected to facilitate the adoption of Industry 4.0 technology, which aims to provide future autonomy for the adaptation of existing factories into smart manufactories [37].

In this study, we also studied the pick-and-place scenario, and analyzed and verified the proposed system through use cases. The application-level latency analyses demonstrate that the OPC UA protocol can ensure high performance compared with other protocols and provide a competitive service

for the information model. Based on the proposed approach and the results of our experiments, the most important contribution of this study is defining the role of the industrial system as four distributed domains (from user to end-device) using the PubSub-model-based OPC UA synergy platform.

These contributions can improve system management efficiency by simplifying the complex life cycle of industrial systems, as well as general systems, and by enhancing their flexibility. Moreover, the proposed comprehensive architecture that integrates the internet and industrial networks in previously unexplored ways can achieve interoperability and transmission of large numbers of messages.

We also provide an outlook related to OPC UA PubSub. First, recent studies [38]–[40] have paid attention to UADP based Time Sensitive Networking (TSN) oriented for achieving real-time industrial communication. Second, the OPC UA PubSub based on 5G will be a key research topic in the future [41]. In particular, AMQP PubSub can provide strong security and ultra-reliable features when combining IIoT with 5G with the cloud domain side [42]. Therefore, we envision that future technologies could be effectively connected or integrated via the proposed integrated architecture of the OPC UA synergy platform.

REFERENCES

- [1] M. AbdelHafeez, A. H. Ahmed, and M. AbdelRaheem, "Design and operation of a lightweight educational testbed for Internet-of-Things applications," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11446–11459, Dec. 2020, doi: [10.1109/JIOT.2020.3012022](https://doi.org/10.1109/JIOT.2020.3012022).
- [2] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, 2016, doi: [10.1109/JIOT.2016.2584538](https://doi.org/10.1109/JIOT.2016.2584538).
- [3] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102781, doi: [10.1016/j.jnca.2020.102781](https://doi.org/10.1016/j.jnca.2020.102781).
- [4] R. Xie, Q. Tang, C. Liang, F. R. Yu, and T. Huang, "Dynamic computation offloading in IoT fog systems with imperfect channel-state information: A POMDP approach," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 345–356, Jan. 2021, doi: [10.1109/JIOT.2020.3004223](https://doi.org/10.1109/JIOT.2020.3004223).
- [5] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial Internet of Things (IIoT): An analysis framework," *Comput. Ind.*, vol. 101, pp. 1–12, Oct. 2018, doi: [10.1016/j.compind.2018.04.015](https://doi.org/10.1016/j.compind.2018.04.015).
- [6] H. Seo, J. Park, M. Bennis, and W. Choi, "Communication and consensus co-design for distributed, low-latency, and reliable wireless systems," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 129–143, Jan. 2021, doi: [10.1109/JIOT.2020.2997596](https://doi.org/10.1109/JIOT.2020.2997596).
- [7] S. Cavalieri, M. G. Salafia, and M. S. Scroppo, "Integrating OPC UA with web technologies to enhance interoperability," *Comput. Standards Interfaces*, vol. 61, pp. 45–64, Jan. 2019, doi: [10.1016/j.csi.2018.04.004](https://doi.org/10.1016/j.csi.2018.04.004).
- [8] R. Endeley, T. Fleming, N. Jin, G. Fehring, and S. Cammish, "A smart gateway enabling OPC UA and DDS interoperability," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, Aug. 2019, pp. 88–93.
- [9] Z. Luo, S. Hong, R. Lu, Y. Li, X. Zhang, J. Kim, T. Park, M. Zheng, and W. Liang, "OPC UA-based smart manufacturing: System architecture, implementation, and execution," in *Proc. 5th Int. Conf. Enterprise Syst. (ES)*, Sep. 2017, pp. 281–286.
- [10] S. Kanno, F. Schnicke, and P. O. Antonino, "Enabling industry 4.0 communication protocol interoperability: An OPC UA case study," in *Proc. 7th Conf. Eng. Comput. Based Syst.*, May 2021, pp. 1–9.
- [11] Industrial Internet Consortium. (2017). *Industrial Internet Reference Architecture (Version 1.8)*. [Online]. Available: <https://www.iiconsortium.org/IIRA-1-7-ajs.pdf>

- [12] D. Li, H. Guo, J. Zhou, L. Zhou, and J. W. Wong, "SCADAWall: A CPI-enabled firewall model for SCADA security," *Comput. Secur.*, vol. 80, pp. 134–154, Jan. 2019, doi: [10.1016/j.cose.2018.10.002](https://doi.org/10.1016/j.cose.2018.10.002).
- [13] *Fieldbus Standard for Use in Industrial Control Systems—Part 3: Datalink Service Specification; Part 4: Data Link Protocol Specification*, Standard IEC 61158-3, 1998.
- [14] T. Winter, P. Thubert, A. Brandt, J. W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. K. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document IETF RFC. 6550, 2012 1–157.
- [15] The Constrained Application Protocol (CoAP), *Internet Engineering Task Force (IETF)*, document RFC 7252, 2014.
- [16] H. Koziol, A. Burger, M. Platenius-Mohr, J. Ruckert, and G. Stomberg, "OpenPnP: A plug-and-produce architecture for the industrial Internet of Things," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng., Softw. Eng. Pract. (ICSE-SEIP)*, May 2019, pp. 131–140.
- [17] S. Gruner, J. Pfrommer, and F. Palm, "RESTful industrial communication with OPC UA," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1832–1841, Oct. 2016, doi: [10.1109/TII.2016.2530404](https://doi.org/10.1109/TII.2016.2530404).
- [18] *Part 1: Overview and Concepts, OPC Unified Architecture Specification*, Standard IEC 62541-1:2020, Rev. 1.04, 2017.
- [19] Z. Liu and P. Bellot, "OPC UA PubSub implementation and configuration," in *Proc. 6th Int. Conf. Syst. Informat. (ICSAI)*, Shanghai, China, Nov. 2019, pp. 1063–1068, doi: [10.1109/ICSAI48974.2019.9010445](https://doi.org/10.1109/ICSAI48974.2019.9010445).
- [20] Z. Liu and P. Bellot, "A configuration tool for MQTT based OPC UA PubSub," in *Proc. RIVF Int. Conf. Comput. Commun. Technol. (RIVF)*, Ho Chi Minh City, Vietnam, Oct. 2020, pp. 1–6, doi: [10.1109/RIVF48685.2020.9140792](https://doi.org/10.1109/RIVF48685.2020.9140792).
- [21] G. Kokkonis, A. Chatzimpampas, and S. Kontogiannis, "Middleware IoT protocols performance evaluation for carrying out clustered data," in *Proc. South-Eastern Eur. Design Autom., Comput. Eng., Comput. Netw. Soc. Media Conf. (SEEDA_CECNSM)*, Sep. 2018, pp. 1–5, doi: [10.23919/SEEDA-CECNSM.2018.8544929](https://doi.org/10.23919/SEEDA-CECNSM.2018.8544929).
- [22] 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 *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2019, pp. 955–962, doi: [10.1109/ICIT.2019.8755050](https://doi.org/10.1109/ICIT.2019.8755050).
- [23] F. Bosi, A. Corradi, G. Di Modica, L. Foschini, R. Montanari, L. Patera, and M. Solimando, "Enabling smart manufacturing by empowering data integration with industrial IoT support," in *Proc. Int. Conf. Technol. Entrepreneurship (ICTE)*, Sep. 2020, pp. 1–8, doi: [10.1109/ICTE47868.2020.9215538](https://doi.org/10.1109/ICTE47868.2020.9215538).
- [24] *Advanced Message Queuing Protocol (AMQP)*, AMQP version 1.0, Standard ISO/IEC 19464:2014 (amqp), 2018.
- [25] *Message Queuing Telemetry Transport (MQTT)*, MQTT version 3.1, Standard ISO/IEC 20922:2016 (mqtt), 2014.
- [26] *Part 3: Address Space Model, OPC Unified Architecture Specification*, Standard IEC 62541-3:2020, Rev. 1.04, 2017.
- [27] *Part 4: Services, of OPC Unified Architecture Specification*, Standard IEC 62541-4:2020, Rev. 1.04, 2017.
- [28] *Part 5: Information Model of OPC Unified Architecture Specification*, Standard IEC 62541-5:2020, Rev. 1.04, 2017.
- [29] *Part 6: Mappings, OPC Unified Architecture Specification*, Standard IEC 62541-6:2020, Rev. 1.04, 2017.
- [30] *Part 14: PubSub, OPC Unified Architecture Specification*, Standard IEC 62541-14:2020, Rev. 1.04, 2017.
- [31] *Internet of Things (IoT)—Reference Architecture*, document ISO/IEC, ISO/IEC 30141, 2018.
- [32] C. Gomez, A. Arcia-Moret, and J. Crowcroft, "TCP in the Internet of Things: From ostracism to prominence," *IEEE Internet Comput.*, vol. 22, no. 1, pp. 29–41, Jan. 2018, doi: [10.1109/MIC.2018.112102200](https://doi.org/10.1109/MIC.2018.112102200).
- [33] S. Kirchmayer, "Ethernet POWERLINK communication profile specification," Ethernet POWERLINK Standardisation Group, Kurfuerstenstraße, Tech. Rep. EPSG DS 301 V1.2.0:2013, 2013, p. 10787, vol. 112.
- [34] N. Naik, P. Jenkins, P. Davies, and D. Newell, "Native web communication protocols and their effects on the performance of web services and systems," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT)*, Dec. 2016, pp. 219–255, doi: [10.1109/CIT.2016.100](https://doi.org/10.1109/CIT.2016.100).
- [35] B. Constantine, G. Forget, R. Geib, and R. Schrage, *Framework for TCP Throughput Testing*, Internet Engineering Task Force (IETF), document RFC 6349, 2011, p. 27.
- [36] A. Abdou, A. Matrawy, and P. C. van Oorschot, "Accurate one-way delay estimation with reduced client trustworthiness," *IEEE Commun. Lett.*, vol. 19, no. 5, pp. 735–738, May 2015, doi: [10.1109/LCOMM.2015.2411591](https://doi.org/10.1109/LCOMM.2015.2411591).
- [37] X. Ye and S. H. Hong, "Toward industry 4.0 components: Insights into and implementation of asset administration shells," *IEEE Ind. Electron. Mag.*, vol. 13, no. 1, pp. 13–25, Mar. 2019, doi: [10.1109/MIE.2019.2893397](https://doi.org/10.1109/MIE.2019.2893397).
- [38] M. Gundall, C. Huber, P. Rost, R. Halfmann, and H. D. Schotten, "Integration of 5G with TSN as prerequisite for a highly flexible future industrial automation: Time synchronization based on IEEE 802.1AS," in *Proc. 46th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2020, pp. 3823–3830, doi: [10.1109/IECON43393.2020.9254296](https://doi.org/10.1109/IECON43393.2020.9254296).
- [39] J. Pfrommer, A. Ebner, and S. B. Ravikumar Karunakaran, "Open source OPC UA PubSub over TSN for realtime industrial communication," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Turin, Italy, Sep. 2018 pp. 1087–1090, doi: [10.1109/ETFA.2018.8502479](https://doi.org/10.1109/ETFA.2018.8502479).
- [40] S. K. Panda, M. Majumder, L. Wisniewski, and J. Jasperneite, "Real-time industrial communication by using OPC UA field level communication," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 1143–1146, doi: [10.1109/ETFA46521.2020.9211998](https://doi.org/10.1109/ETFA46521.2020.9211998).
- [41] Y. Siriwardhana, P. Porambage, M. Ylianttila, and M. Liyanage, "Performance analysis of local 5G operator architectures for industrial internet," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11559–11575, Dec. 2020, doi: [10.1109/JIOT.2020.3024875](https://doi.org/10.1109/JIOT.2020.3024875).
- [42] H. Abukwaik, A. Gogolev, C. Groß, and M. Aleksey, "OPC UA realization for simplified commissioning of adaptive sensing applications for the 5G IIoT," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100221, doi: [10.1016/j.iot.2020.100221](https://doi.org/10.1016/j.iot.2020.100221).



CHANGDAE LEE received the B.S. degree in electronic engineering from Kangwon National University, Gangwon-do, South Korea, in 2017. He is currently pursuing the Ph.D. degree in electronic engineering with Hanyang University, Ansan, South Korea. His research interests include reliability, performability, communication protocols for industrial control and automation and industrial informatics, and wireless communication protocols.



NAMHYEOK KIM received the B.S. degree in electronic engineering from Kangwon National University, Gangwon-do, South Korea, in 2018, and the M.S. degree in electronic engineering from Hanyang University, Ansan, South Korea. He is currently a Researcher with the Nestfield Research and Development Center, Ansan. His research interests include cyber-physical systems and industry 4.0.



SEUNGHO HONG (Senior Member, IEEE) received the B.S. degree in mechanical engineering from Yonsei University, Seoul, South Korea, in 1982, the M.S. degree in mechanical engineering from Texas Tech University, Lubbock, TX, USA, in 1985, and the Ph.D. degree in mechanical engineering from The Pennsylvania State University, University Park, PA, USA, in 1989. He was a Visiting Scholar with the U.S. National Institute of Standards and Technology, the Vienna University of Technology, Austria, and Zhejiang University, China. He is currently a Professor with the Department of Electronic Engineering and the Director of the Connected Smart Systems Laboratory, Hanyang University, Ansan, South Korea. He is also a Foreign Expert with the Tianjin University of Technology through the Tianjin Thousand Talents Program. His research interests include smart manufacturing, smart grid, industrial Internet of Things, cyber-physical systems, and artificial intelligence. He is a member of the IEEE Industrial Electronics Society and the International Electrotechnical Commission System Committee Smart Manufacturing.

...