# WHITE PAPER

# Implementing Deterministic OPC UA Communication

#### **EXECUTIVE SUMMARY**

In recent years, OPC UA has proven to be the leading manufacturer-independent standard for data exchange between different devices, one which supports various operating systems and integrates different automation levels.

optimize!

However, when looking into the requirements for implementing the Industrial Internet of Things and "Industrie 4.0" in detail, we find that the OPC UA standard based on a client/server architecture is not the ideal model. As a consequence, the OPC Foundation is in the process of standardizing OPC UA Publisher/ Subscriber as an additional communication model.

From a theoretical point of view, the OPC UA Publisher/Subscriber communication model is ideal for addressing the needs of an Industrial Internet of Things application. So far, however, a practical evaluation of OPC UA Publisher/Subscriber capabilities has not yet been carried out.

This white paper provides an overview of a prototype implementation of the OPC UA Publisher/Subscriber stack and shows the detailed results of the timing measurements that were performed based on this demonstrator.



# **Table of Contents**

1.	Introduction to OPC UA1	
1.	1 From OPC Classic to OPC UA1	
1.	2 Overview of OPC UA	
2.	Implementing the Industrial Internet of Things5	
2.	1 Implementing the Industrial Internet of Things5	
	2.1.1 Use Cases Requiring Publisher/Subscriber Communication Capabilities5	
	2.1.2 Use Cases Requiring Deterministic Publisher/Subscriber Communication Capabilities	
2.	2 OPC UA Publisher/Subscriber Communication Model9	
2.	3 Time-Sensitive Networking	
3.	Proof of Concept of the Industrial Internet of Things Environment	
3.	1 Test Infrastructure	
3.	2 Measurements Using the Demonstrator14	
3.	3 Measurement Results 14	
4.	Summary16	
5.	About Softing	
6.	Authors	
7.	References	
8.	List of Figures	
9.	List of Tables	



#### 1 Introduction to OPC UA

#### 1.1 From OPC Classic to OPC UA

The OPC standard for data exchange was initially based on the use of PC platforms running Windows operating systems. Microsoft's COM/DCOM (Distributed) Component Object Model was used as the base technology for the OPC standard. This close tie has placed a number of restrictions on the use of OPC technology. For example, DCOM-based OPC technology is not suitable for communication over the Internet, for the use of firewalls or for operation on non-Windows platforms. In particular, configuring the exchange of data between PCs requires extensive expertise in DCOM security settings. As COM/DCOM has since been discontinued by Microsoft, this technology is no longer being maintained and further developed. In addition, new desires and demands relating to OPC data exchange have evolved in the industry regarding security support, protection against data loss, redundancy capabilities and support for complex data connections, for example.

In response to this situation, the OPC Foundation developed a fully revised and expanded version of the OPC standard that eliminates the weak points of classic OPC technology. This version is independent of the employed operating system, programming language or proprietary technologies. It is manufacturer-independent and supports scalability, high availability and Internet capability. This new OPC technology generation has been released under the name of OPC UA (Unified Architecture), while the classic OPC technology is now known as OPC Classic. Figure 1 and Figure 2 illustrate the major differences between OPC Classic and OPC UA.







Figure 1: Architecture of OPC Classic Using Windows PCs and Microsoft's DCOM Technology for Data Exchange



Figure 2: Architecture of Simplified Data Exchange Using OPC UA



#### 1.2 Overview of OPC UA

OPC UA is an open and free standard that is specified in the IEC62541 standard. As opposed to OPC Classic, OPC UA makes use of an object-oriented information model supporting structures, objects, state machines, inheritance, etc. In addition, it uses a newly designed service-oriented architecture (SOA) that supports the easy customization of OPC UA. OPC UA makes it possible to exchange raw data and pre-processed information between embedded systems at the sensor and field levels and ERP, MES and process visualization systems, while ensuring that all data is available to any authorized application and person anywhere and anytime. To exchange data, OPC UA uses an optimized binary protocol based on TCP. Only a single port needs to be opened in a firewall.

OPC UA incorporates proven Internet security concepts, such as the Secure Sockets Layer (SSL), Transport Layer Security (TLS) and Advanced Encryption Standard (AES), which protect against unauthorized access, process value modifications, sabotage and faults caused by negligent use. The security capabilities are a mandatory part of the standard and include user and application authentication, digital message signatures and the encryption of transmitted data. Users can freely combine the different security features to suit their specific use cases so that scalable solutions can be created.

OPC UA uses a robust architecture with reliable communication mechanisms, configurable time monitoring and automatic fault detection. The fault correction mechanisms automatically reestablish the communication link between the OPC UA client and OPC UA server without data loss. OPC UA provides redundancy functionality that can be integrated with client and server applications to achieve high system availability and maximum reliability.

OPC UA represents automation processes using an integrated address space and a common information model based on objects and variables, method calls for process data, alarms and conditions, and the representation of historical data. In addition to covering the full OPC Classic functionality, OPC UA provides features for describing complex procedures and systems in standardized object-oriented components. Information consumers that only support the basic rules can process the data even without knowing the interrelations of the complex structures of a server. The universal applicability of the OPC UA technology makes it possible to implement completely new vertical integration concepts. By cascading OPC UA components, information can be securely and reliably transported from the production level to the ERP and MES systems. This is accomplished by establishing a direct connection between embedded OPC UA servers at the field level and integrated OPC UA clients in business-level systems. The individual OPC UA components can be geographically distributed and separated from each other by firewalls. This makes OPC UA ideally suited for use in distributed systems. As a result, OPC UA occupies an important position in the vision of future industrial production as part of a fully intelligent environment (see Figure 3). In addition, OPC UA enables other standardization bodies to use OPC UA services as transports for their own information models.





Figure 3: Vision of Future Industrial Production as Part of a Fully Intelligent Environment

In summary, OPC UA uses a very powerful and standardized technology for industrial data exchange. It is based on a client/server communication model which requires an established connection. OPC UA offers major advantages here:

- OPC UA provides the means for a secure cross-platform data exchange.
- OPC UA offers interoperability between platforms from multiple manufacturers.
- OPC UA is more than a pure communication protocol and also addresses semantics and modeling capabilities to represent complex physical systems in digitalized mode.
- OPC UA can be used for implementing interoperability between different standardized communication protocols.



#### 2 Implementing the Industrial Internet of Things

Because it offers the ideal solution to the individual needs and requirements of the industry, the OPC UA standard has great potential for the future. After all, OPC UA is envisaged as the data exchange technology for the future implementation of the Industrial Internet of Things (IIoT) and "Industrie 4.0"<sup>1</sup>.

#### 2.1 Implementing the Industrial Internet of Things

When looking at the various scenarios for IIoT implementations in more detail, it soon becomes clear that the related communication requirements do not fit ideally with the communication capabilities as defined by the OPC UA standard so far. In order to understand this issue, we will first discuss various cases of IIoT implementations as defined by the OPC Foundation. These use cases can be categorized as large-scale communications from one to many, from many to one or from many to many. Configured controller-to-controller communication is also included here.

#### 2.1.1 Use Cases Requiring Publisher/Subscriber Communication Capabilities

When discussing possible IIoT implementations, we find different scenarios which do not fit well with a client/server architecture as defined by the OPC UA standard. Instead, a publisher/subscriber model is more appropriate. A summary of these use cases is provided in Table 1.<sup>2</sup>

<sup>&</sup>lt;sup>1</sup> "Industrie 4.0" is a German government-driven initiative for implementing an Industrial Internet of Things.

<sup>&</sup>lt;sup>2</sup> The use cases presented in this section are taken from the OPC Foundation document *UseCases-OpcUa-PubSub*, version 0.09.



Use Case	Description	
Public Subscriptions	<ul> <li>A large number of clients require information about configuration changes for a list of variables. The data exchange is performed after the initial system setup or upon any configuration change.</li> <li>The client/server model is not very efficient for handling this situation due to the following reasons: <ul> <li>A large number of client/server connections have to be established.</li> <li>Each client needs to provide memory for storing the connection information as well as the individual variable values.</li> <li>A high processor load is generated in the server for encoding the individual messages per established connection. Additional load is required if the clients have defined different sampling rates for the variables.</li> </ul> </li> </ul>	
Secure Multicast	Data values need to be sent by the server to a large number of clients. The data exchange is performed either cyclically or upon any value change.	
Many-to-One Publishing	One or more client(s) located in a cloud needs data from a large number (thousands) of devices behind a firewall. The data exchange is performed either cyclically or triggered by the change of a value, quality or alarm. It is not possible to handle this situation, as the number of open connections is too huge to be handled in parallel.	
Machine-to-Machine Communication	Machine units within a plant need to exchange process data either down- stream with machine modules or upstream with SCADA systems, ensuring an end-to-end delivery time in the range of 2 to 100 ms. Process data may contain control and status information, e.g., PackML/PackTags as defined in ISA Technical Report TR88.00.02. The data exchange is performed cyclically.	
Dynamic Network Relations	Mobile devices, such as optional machine parts, mobile robots and measure- ment equipment, can be flexibly added to or removed from a machine (for example, this situation applies to robots supporting the full programming of the robot together with its mobile devices, or to remote procedure calls). The machine and the mobile devices have separate controllers which need to communicate in a deterministic way. Depending on the individual mobile device, different tasks can be performed by the mobile device (e.g., a mobile robot can support different handling tasks: gluing at one station, welding at another station, handling at a third station, etc.). Once a mobile device has been connected to a machine, an appropriate communication relationship needs to be established which fits with the list of variables and functions available in the mobile device. Cyclic process data as well as remote function calls can be exchanged.	

Table 1: Use Cases Requiring Data Exchange Beyond the Scope of OPC UA Client/Server Capabilities



7

Use Case	Description	
OPC UA Multi-HMI	Complex control systems require data to be served simultaneously to many clients. For instance, several human machine interfaces (HMIs) may display identical information throughout the facility to provide operators access to information wherever they are in the facility. To support specialized HMIs as well, several pre-defined data sets have to be supported to be exchanged. A typical quantity structure for data exchange includes up to 30,000 data points being served to 50-100 clients at a desired update rate of 200 ms. The data exchange is performed cyclically, but asynchronous events also need to be communicated to multiple HMIs simultaneously so that they are viewable throughout the entire plant. As soon as a user acknowledges an alarm or performs other interactions at one HMI, this HMI controls the state of the corresponding event.	
Controller-to-Controller Communication	Controller-to-controller communication can either be performed based on PLC program function blocks ("programmed controller-to-controller commu- nication") or be established using a central configuration tool ("configured controller-to-controller communication"). While the establishment and modification of programmed controller-controller communication requires changes to the PLC program, the establishment and modification of config- ured controller-controller communication requires a standardized configura- tion interface but no changes to the PLC program.	
Data Streaming	A server provides a stream of measurement values as fast as possible, but reliable transport is not required. Either this measurement stream is trans- ferred at high speed for a certain period of time, or the measurement stream is continuously transferred. The data exchange is therefore either performed on demand or permanently.	
Video/Audio Streaming Management (Audio Streaming)	A server provides access to a system that has video and/or audio sources available. This type of server supports control actions such as zooming a camera or modifying settings for video and audio streams through standard OPC UA services and protocols, while the video or the audio stream is trans- ferred through another channel using standard video and audio streaming protocols.	

Table 1:Use Cases Requiring Data Exchange Beyond the Scope of OPC UA Client/Server Capabilities<br/>(Continued)



#### 2.1.2 Use Cases Requiring Deterministic Publisher/Subscriber Communication Capabilities

Beyond the use cases described in Section 2.1.1, additional use cases exist which, in addition to the support of a publisher/subscriber communication model, require deterministic communication behavior as well. Table 2 provides an overview of these use cases.<sup>3</sup>

Use Case	Description		
Cyclic Controller-to-Controller Communication	For laser-cutting machinery, cyclic communication between controllers (PLCs, NCs, laser controllers) with a cycle time of 1 ms is required during operation. As this communication is used for control tasks, the latency and jitter must be minimal. Besides critical communication, there is also non-critical cyclic and event-based communication, e.g., for visual data or for exchanging data with MES/ERP systems.		
Event-based Controller-to-Controller Communication	A package/parcel identification and handling system needs to exchange data with cameras and RFID systems as well as with PLCs for sorting tasks during regular operations. The data exchange is event-based (up to 18 events per second) and defined by the presence of a parcel. As this communication is used for control tasks, a minimal latency (100 ms) is required. Non-critical cyclic and event-based communication is also performed.		
Cyclic Communication Between Smart Sensor and Controller	For laser range measuring applications, laser range sensors cyclically exchange data with a central controller or actuator. In this case, a cycle time of 10 ms is required, resulting in a data volume of 15 MB per second. Time synchronization is required for coordinating the measurements from the various sensors.		
Cyclic Communication Between Robot and Tool Controller	For robotic tool coordination, a cyclic data exchange based on a cycle time of 1-5 ms is required. As this communication is used for control tasks, a minimal latency is required. In addition, other non-critical communication is performed between the robot and the tool controller as well as for data exchange, e.g., with MES systems and databases.		
Cyclic Communication Between Machine Controller and Transporter/Carrier	When moving materials through the production facility, cyclic communication between the controller and the mobile devices is required. The communica- tion has to meet a cycle time of 1-5 ms. A minimal latency ensures no side- effects when controlling the mobile devices. In addition, non-critical commu- nication is performed between the tool controller and the carrier.		
Safety-Related Communication	The exchange of safety-related data is performed using a safety protocol. This protocol makes use of a black channel communication layer, meaning that there are components in the communication path that have no safety-related design. Therefore, the response message from the partner device has to be received within a defined time frame. If the worst-case transmission delay time has expired, the machine will transition into a safe state. The bi-directional data exchange is performed cyclically based on a deterministic timing.		

 Table 2:
 Use Cases Requiring Deterministic OPC UA Publisher/Subscriber Data Exchange

<sup>3</sup> The use cases included in this section are taken from the OPC Foundation document *UseCases-OpcUa-PubSub-TSN*, version 0.04.



#### 2.2 OPC UA Publisher/Subscriber Communication Model

As the individual messages are sent from one sender to one recipient in this case, the OPC data exchange via client/server is not very efficient for implementing the use cases as described in Section 2.1. Instead, multicast and broadcast messages would fit better with the situations described. As a consequence, the OPC Foundation has started to work on a new OPC UA Publisher/Subscriber communication model which addresses these requirements. An overview of the components used for OPC UA Client/Server Subscriber communication and OPC UA Publisher/Subscriber communication can be found in Figure 4.

	OPC Items	OPC Items
OPC Communication Model	OPC UA Client/Server Stack	OPC UA Publisher/Subscriber Stack
OPC Protocol	OPC Binary, HTTP SOAP, HTTPS,	UDP Secure Multicast, AMQP,
Transport Layer	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Network Layer	IP Protocol (IP)	IP Protocol (IP)
	OPC UA Client/Server	OPC UA Publisher/Subscriber

#### Figure 4: Comparison of the OPC UA Client/Server and the OPC UA Publisher/Subscriber Communication Models

The OPC UA Publisher/Subscriber communication model defines the exchange of messages between a publisher and a list of subscribers within a distributed architecture. For this exchange to take place, an appropriate infrastructure is required which allows publishers to send messages without knowing what, if any, subscribers may exist. Similarly, subscribers communicate with this infrastructure to express their interest in certain types of messages, only receiving messages that are of interest without knowing what, if any, publishers there are. As a result, the OPC UA Publisher/Subscriber communication model reduces the load on the information provider's side by collecting and sending the required data just once for several recipient applications at an individually defined rate. The general structure of OPC UA Publisher/Subscriber communication is shown in Figure 5.





Figure 5: OPC UA Publisher and Subscriber Applications (Source: OPC Foundation)

Depending on the specific use case, different implementations of the OPC UA Publisher/Subscriber communication model are possible. The ideal option for implementing the IIoT use cases presented in Section 2.1, which require a fast local network, is based on the User Datagram Protocol (UDP) Secure Multicast protocol. It enables the implementation of thin and efficient protocol stacks for message handling, and it supports a cyclic data exchange as well. Communication is characterized by a small load plus a fast and reliable data exchange, while the specified OPC UA information model does not need to be modified. In addition to the UDP Secure Message protocol, other protocols are defined for the OPC UA Publisher/ Subscriber communication model as well, e.g., the Advanced Message Queuing Protocol (AMQP), which is best used for the delivery of messages to subscribers within a global network, e.g., using a cloud.

Within the scope of this white paper, the OPC UA Publisher/Subscriber communication model is discussed based on the UDP Secure Multicast protocol.



# 2.3 Time-Sensitive Networking

When checking the requirements of the use cases presented in Section 2.1, and particularly in Section 2.1.2, against the capabilities provided by the OPC UA Publisher/Subscriber communication model, it is evident that most requirements are met, but the required deterministic data exchange cannot be ensured. Therefore, an additional approach has to be followed.

The details of the requirements to be addressed include:

- The individual nodes within the network have to work based on synchronized timing. As a result, a system-wide common clock, for example, is available for scheduling the transmission of data or performing a correlated input and output of data.
- Path redundancy is required to ensure the data exchange even if specific components within the overall network fail. This increases the reliability of the communication and establishes a fault-tolerant system.
- For enabling deterministic control loops, the communication between individual nodes has to be performed within a pre-defined latency time.
- A reserved bandwidth has to be supported for the exchange of data between critical applications, ensuring reliable operation even in the presence of a high traffic load and network congestion.

Deterministic communication behavior in an Industrial Ethernet environment which addresses the requirements described above can be achieved using the Time-Sensitive Networking (TSN) standard. This is based on the Audio-Visual Bridging (AVB) standard released in 2011 and has been identified as a good starting point for implementing industrial applications. The resulting TSN standard provides some enhancements in comparison to the AVB standard, which are defined by a set of IEEE 802 Ethernet sub-standards. These enable time-scheduling capabilities and, as a result, fully deterministic real-time communication. This is achieved by using globally synchronized timing together with a schedule which is shared between the various network components. This schedule provides reserved time slots within an overall TSN cycle time which can be used to transfer prioritized messages, resulting in a bound maximum latency for the scheduled traffic throughout a switched network.



#### **3** Proof of Concept of the Industrial Internet of Things Environment

In theory, both the OPC UA Publisher/Subscriber model and the Time-Sensitive Networking technologies described in Chapter 2 are capable of enabling a deterministic data exchange as required for IIoT implementations. Before using these technologies in real applications, however, it is necessary to check the adaptability of this approach. This chapter therefore addresses the prototype installation of an environment incorporating these technologies.<sup>4</sup>

#### 3.1 Test Infrastructure

The set-up of the test infrastructure included the implementation of the OPC UA Publisher/Subscriber stack together with an OPC UA client and server interface. This interface provides a general way to establish a communication application for data exchange based on existing OPC UA clients and servers. This implementation has been carried out for a device including a field programmable gate array (FPGA). The architecture of the device's FPGA is shown in Figure 6.



Figure 6: Device Architecture Supporting the OPC UA Publisher/Subscriber Communication Model

Using the OPC UA Publisher/Subscriber devices as the main components, a demonstrator has been set up for investigating the timing behavior of OPC UA Publisher/Subscriber-based communication using a TSN network.

<sup>&</sup>lt;sup>4</sup> This prototype installation has been developed and set up by Softing.



#### • 2 Devices

running a sample application based on a static OPC UA Publisher/Subscriber configuration The data exchange is performed based on the OPC UA Publisher/Subscriber communication model using the OPC UA Publisher/Subscriber stack.

The devices implement an Industrial Ethernet switch<sup>5</sup> and provide a standard Ethernet interface.

• 2 Switches

supporting the TSN standard, thus enabling Ethernet traffic based on a TSN schedule

• Frame Generator

used for generating additional load on the Ethernet network

• Analyzing Tool

used for recording and analyzing the network traffic

#### Note:

As the test infrastructure only includes two TSN-enabled devices, no general time synchronization is in place yet.

The architecture of this demonstrator is presented in Figure 7.





<sup>&</sup>lt;sup>5</sup> The implemented switch uses Softing's Switch IP Core which was especially developed for supporting Industrial Ethernet, e.g., PROFINET IRT requirements, in an FPGA environment.



#### 3.2 Measurements Using the Demonstrator

The demonstrator introduced in Section 3.1 has been used to measure the jitter (interval between individual sent frames). This measurement is performed with and without TSN network communication support.

Within the jitter measurement, the goal has been to determine the jitter experienced while sending OPC UA Publisher/Subscriber frames.

The jitter measurement is performed by the following steps:

- 1. Insert the analyzing tool between the receiving device and the switch
- 2. Start the OPC UA Publisher/Subscriber data exchange using a configured sending interval of 10 ms
- 3. Record the jitter with and without TSN network communication support using a configured TSN cycle time of 0.5 ms
- 4. Generate background traffic based on 70,000 packets per second
- 5. Record the jitter with and without TSN network communication support using a configured TSN cycle time of 0.5 ms

#### 3.3 Measurement Results

Figure 8 and Figure 9 provide an overview of the jitter measured for the OPC UA Publisher/Subscriber communication.

When using standard Ethernet networks, the jitter results show a certain range even when no additional traffic is added. With additional background traffic, the bandwidth grows even wider (see Figure 8).

When using Time-Sensitive Networking, the measured trigger is largely bound to the configured time interval of 10 ms. Only a small part of the communication is performed in the next TSN cycle, resulting in an appropriately reduced time interval for the subsequent data exchange. This effect is caused by the fact that there is no general time synchronization between the various network devices. In general, this behavior does not change when adding background traffic to the network (see Figure 9).





Figure 8: Jitter Measurement Results With and Without Background Traffic Without Using TSN-enabled Networks



Figure 9: Jitter Measurement Results With and Without Background Traffic When Using TSN-enabled Networks



#### 4 Summary

In combination with Time-Sensitive Networking, the OPC UA Publisher/Subscriber communication model significantly improves deterministic timing behavior in comparison to standard OPC UA Client/Server communication. In particular, it was found that additional network load does not result in any deterioration of the communication activities and, parallel to this, the available bandwidth is not wasted by restricting the transferred messages. This approach therefore ideally addresses the specific communication requirements of IIoT applications as presented in Section 2.1.

In order to implement the discussed solution, consisting of the OPC UA Publisher/Subscriber communication model and Time-Sensitive Networking, appropriate software and hardware is required. It is critical, therefore, to not base any planned IIoT implementation solely on the available software. Instead, suitable hardware support is also essential.<sup>6</sup>

<sup>&</sup>lt;sup>6</sup> This requirement fits perfectly with Softing's offer, which combines the required hardware and software components within an FPGA.



#### 5 About Softing

The Softing Industrial Automation segment is part of the Softing Group, which was founded in 1979.

Softing Industrial Automation is a global specialist in industrial communication technologies such as fieldbuses and Industrial Ethernet. With over 35 years of experience, Softing's Industrial Automation business units deliver connectivity, diagnostic products and services to customers in the factory and process automation industry.

With a track record of more than 20 years of experience in OPC technology, Softing is the OPC partner of choice for a large number of companies. Softing offers a complete set of OPC UA and OPC Classic development tools and end-user products. These provide a comprehensive set of functionalities for implementing state-of-the-art data exchange solutions addressing all individual connectivity issues. The portfolio is supplemented with complementary training and development services as well as the world's leading OPC book.

Softing has also developed a general device platform based on FPGA technology. This platform can be tailored to individual requirements by loading the appropriate hardware and software onto the FPGA. For instance, it is used for implementing fieldbus and Industrial Ethernet devices of various forms.

Softing's products are tailored to the requirements of system integrators, device vendors, machine and equipment manufacturers or end users, and they are known for their ease of use and functional advantages.

Softing Industrial Automation GmbH Richard-Reitzner-Allee 6 – 85540 Haar – Germany Phone: +49 89 45 656 -340 Fax: +49 89 45 656 -488 E-mail: info.automation@softing.com http://industrial.softing.com



#### 6 Authors

Cristian Pogacean Product Manager OPC Softing Industrial Automation GmbH Haar (near Munich), Germany

Sebastian Broschei Development OPC Softing Industrial Automation GmbH Haar (near Munich), Germany

Georg Süss Operational Marketing Softing Industrial Automation GmbH Haar (near Munich), Germany

#### 7 References

#### • Expert Interview: OPC UA Publisher/Subscriber Model und IEEE TSN Standard

November 2015 Available at: http://blog.softing.com/wp-content/uploads/2015/11/Interview\_Softing\_TTTech\_EN.pdf (as of April 7, 2016)

# White Paper



### 8 List of Figures

Figure 1:	Architecture of OPC Classic Using Windows PCs and Microsoft's DCOM Technology for Data Exchange
Figure 2:	Architecture of Simplified Data Exchange Using OPC UA2
Figure 3:	Vision of Future Industrial Production as Part of a Fully Intelligent Environment
Figure 4:	Comparison of the OPC UA Client/Server and the OPC UA Publisher/Subscriber Communication Models9
Figure 5:	OPC UA Publisher and Subscriber Applications10
Figure 6:	Device Architecture Supporting the OPC UA Publisher/Subscriber Communication Model 12
Figure 7:	Basic Architecture of OPC UA Publisher/Subscriber Demonstrator13
Figure 8:	Jitter Measurement Results With and Without Background Traffic Without Using TSN-enabled Networks
Figure 9:	Jitter Measurement Results With and Without Background Traffic When Using TSN-enabled Networks

# 9 List of Tables

Table 1:	Use Cases Requiring Data Exchange Beyond the Scope of OPC UA Client/Server Capabilities.6
Table 2:	Use Cases Requiring Deterministic OPC UA Publisher/Subscriber Data Exchange