

# Problems of the Time Deterministic in Applications for Process Control from the Cloud

<http://dx.doi.org/10.3991/ijoe.v10i4.3832>

O.Makarov<sup>1,2</sup>, R. Langmann<sup>1</sup>, S. Nesterenko<sup>2</sup> and B.Frank<sup>1</sup>

<sup>1</sup> Dusseldorf University of Applied Sciences, Düsseldorf, Germany

<sup>2</sup> Odessa National Polytechnic University, Odessa, Ukraine

**Abstract**—This paper is devoted to the problem of data transfer rates between automation devices and cloud services. The possibilities of the device connection to the automation systems are described, their time characteristic are determined. The connections are compared and evaluated accordingly to their compliance with the current requirements of Industry 4.0. The decision about the possibilities of their implementation in modern Cyber Physical Production Systems is made.

**Index Terms**—time deterministic, OPC, Industry 4.0, CPPS, Cloud

## I. INTRODUCTION

The modern world puts forward a number of requirements to devices, including a method of communication between them. Such requirements are fixed in a number of recommendations. Example of these recommendations is Industry 4.0 developed by the German government, so called 4th industrial revolution that was caused by implementation of the integrated autonomous system that communicate with each other via the Internet. In industry this technologies are integrated in so-called Cyber-Physical Production Systems – CPPS with intelligent machines, systems of data and resources storage that can share independently different types of information and control signals [1]. Their implementation can significantly streamline production processes, usage of resources, as well as life-cycle management of the enterprise. In the Industry 4.0 guidelines, among other things is mentioned a requirement to optimize the interaction of the elements of automation systems. Interaction of the system components on different levels should be more transparent and flexible and this can be achieved only by using the same technology on the level of process control as well as in the production management area. That's why nowadays it is extremely important to develop new and research the existing data transfer methods that meet the modern requirements of Industry 4.0. It will allow to use the existing solutions in this area more effectively, e.g. to use data, received from the automation systems in the cloud.

The aim of this research is to determine time characteristics of the process control applications from the cloud and to describe the existing methods of data transfer from the point of Industry 4.0 requirements and make a decision about the possibilities of their implementation in modern CPPS.

## II. MEASUREMENT METHODS

The goal of the measurements is to find out the time that the signal needs to get from the PLC to the website. Also it is important to know, how much time the signal needs on different parts of the way and to explore which parts are most time efficient. A general scheme for data transfer, using OPC DA technology is represented on the figure 1. A Phoenix Contact ILC 150ETH PLC is connected to Spectra Industrial PC (Model: NISE 101

K1.2, Intel Atom N270 1,6GHz, 533MHz, Windows XP) with Ethernet cable. To exclude other network influences it was decided to start the Html-page on the PC, but using an Apache webserver. On the industrial PC the AX OPC- Server 3.0 is installed. To make the data from server available in the Internet some Web to OPC server should be used. This server has an integrated OPC DA client, which is connected to the OPC Server and forwards data to the Web. To make data available on the Web-page a Java Script should be used.

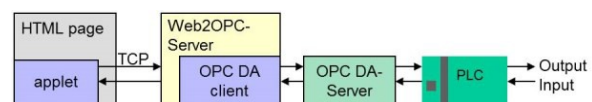


Figure 1. Signal transfer infrastructure

A webpage has no physical output and some connections to them are complicated as well as synchronization of PC and PLC that is why it was made a decision to measure the time for both ways. It is possible to do it on each side of transfer way. The time could be measured between inputs and outputs of the PLC. In this case a software-bridge in webpage was made and the delays were measured with oscilloscope (for this purpose an oscilloscope “Picoscope 5204” was used). Another possibility is to bridge output and input of the PLC and to measure the delay in the webpage using a system time.

The signal transfer way could be divided in separate delay times. Time for proceeding in PLC is marked as  $T_{PLC}$ . Accordingly to PLC Datasheet delay time for 1 K mix- instructions and 90  $\mu s$  for 1 K bit instructions. In our case the program use only a few variables with few connections.

Time for data transfer between PLC and OPC DA server is marked as  $T_{PLCf}$ . For this transfer a TCP protocol was used. In our case as mentioned above the PLC and IPC are directly connected with approx. 0,5m length STP cat.5 cable. If some switches or other network equipment are used, their delay time should be added.

Time for proceeding in OPC DA server is marked as  $T_{OPCs}$ .

Time for data transfer between OPC DA server and client is marked as  $T_{OPCtr}$ . In our case both parts are on the same PC, but the data exchange will occur through TCP protocol. Here it is the same situation as with transfer between PLC and OPC DA server – in case if some switches or other network equipment are used, their delay time should be added.

Time for proceeding in OPC DA client is marked as  $T_{OPCcl}$ .

Time for proceeding in Web to OPC server is marked as  $T_{webS}$ . In our case it is difficult to separate the time for proceeding in OPC DA client and in Web to OPC server, because they are parts of same program code.

Time for data transfer between Web server and client is marked as  $T_{webTr}$ . In our case client is still executed on the same PC and data transfer occurs internally, but we can imagine, that in case of data transfer on long distances it could be the most time efficient part.

Time for proceeding in Web client is marked as  $T_{webC}$ . That means that the full route will take the time:

$$T = T_{PLC} + T_{PLCtr} + T_{OPCs} + T_{OPCtr} + T_{OPCcl} + T_{webS} + T_{webTr} + T_{webC}$$

Remembering that we measure the time for signal sended in both ways:

$$T = (T_{PLC} + T_{PLCtr} + T_{OPCs} + T_{OPCtr} + T_{OPCcl} + T_{webS} + T_{webTr}) \cdot 2 + T_{webC}$$

At the input of the PLC, a function generator is connected, which square-wave voltage is recorded on the A channel of the digital oscilloscope. The logical "value" of input variable is provided by the OPC server. Channel B of the oscilloscope records the delayed output signal. The delay between the input and output wave is the transfer time ( $T$ ) (Fig. 2)

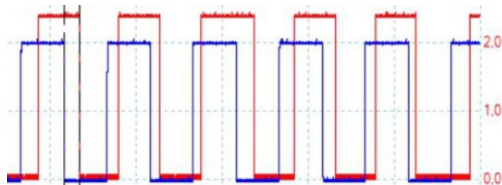


Figure 2. Delay time

### III. MEASUREMENT RESULTS AND ANALYSIS

During the measurements the following dependence was noted. If the frequency is increased or the square wave is modified, that the length of the signal is 32 ms or less some input signals have no outputs (marked on Figure 3 with arrow). The reason for it should be a refresh time of the OPC server [2]. If the length of the signal is less than refresh time of the server, sometimes the signal fall in the gap between two refreshes. That means that only a signal that is longer than 32 ms could be surely recognized. That means also that the delay time can be up to 32 ms longer than just a transfer and processing time because of this. A minimum time could be described as a time when a refresh point is immediately after input signal was changed.

#### A. OPC DA client

OLE for Process Control (OPC), which stands for Object Linking and Embedding (OLE) for Process Control, is the original name for standards specification developed in 1996 by an industrial automation industry task force. To read, write and store the data from/to devices of different manufacturers we can use an OPC technology.

OPC was designed to provide a common bridge for Windows based software applications and process control hardware. Standards define consistent methods of accessing field data from plant floor devices. This method remains the same regardless of the type and source of data.

An OPC Server for one hardware device provides the same methods for an OPC Client to access its data as any other OPC Server for that and any other hardware device. The aim was to reduce the amount of duplicated efforts required from hardware manufacturers and their software partners, and from the SCADA and other HMI producers[2].

To find out the most time efficient part of the signal way it was decided to measure the shorter way from the PLC input through OPC DA server to the OPC DA Java client and back to the PLC output (Fig.3).

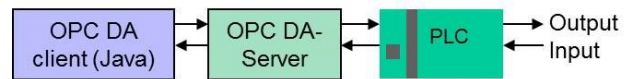


Figure 3. Connection to the OPC DA client

$$T = (T_{PLC} + T_{PLCtr} + T_{OPCs} + T_{OPCtr}) \cdot 2 + T_{OPCcl}$$

A computer connected to the OPC server JAVA client reads the "Value" and writes that value to the output variable of the PLC. The response time of the "short-circuit" client in this case is approximately 0.1 ms.

The following times were measured for input frequencies  $f = 0.5$  Hz,  $f = 2$  Hz and  $f = 5$  Hz, each with an OPC variable for reading and writing:

- Minimum 15,4 ms
- Average 31,85 ms
- Maximum 56,8 ms

During the measurements was noted the following dependence. If the frequency is increased or the square wave is modified, that the length of the signal is 32 ms or less some input signals have no outputs (marked on Figure 4 with arrow). The reason for it should be a refresh time of the OPC server. If the length of the signal is less than refresh time of the server, sometimes the signal fall in the gap between two refreshes.

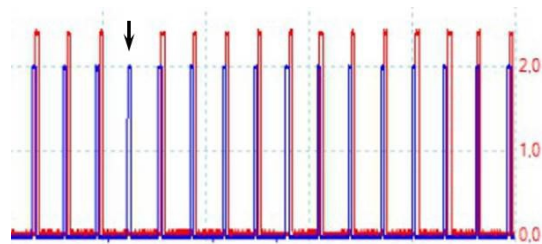


Figure 4. Missing output signal

That means that only a signal that is longer than 32 ms could be surely recognized. That means also that the delay time can be up to 32 ms longer than just a transfer and processing time because of this. A minimum time could be described as a time when a refresh point is immediately after input signal was changed.

**B. WAK**

Web Access Kit – a software application that is an extension for OPC DA client and allows reading and writing data on the server through the Internet browser [4]. The transfer of data between OPC- client and Web applications implemented on top of TCP / IP socket.

The WAK was developed in Java and exists of a row of flexibly useable applets and a server-sided part of the application, the so-called WAK server. The applets serve for the reading and writing of digital and analog process variables of an automated arrangement. Four of it – visualization applets – dispose of a graphical user'ssurface, which shows topical process values dynamically or allows the user to give new values to the process variables. Another applet – W2OPCProxy – serves principally as an interface for script programs and program objects of a HTML page for process data of an automation device. The remote access on the arrangement for the applets is managed by the WAK server. The WAK server runs on a computer connected to a control system and accesses process data via the OPC interface. In the easiest case the calculator can operate as a web server.

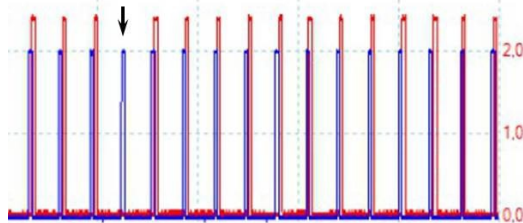


Figure 5. Structure of the data transfer with WAK

The data transfer between the WAK server and the Applets occurs with the help of a Socket connection on a certain TCP port (Figure 5) [5].

A figure 6 gives a representation how the time deterministic by usage of WAK can look like. In this case we can make a conclusion that 95,8 % of signals have a delay time between 18 and 52 ms. The following times were measured for input frequencies  $f = 0.5$  Hz,  $f = 2$  Hz and  $f = 5$  Hz, each with an OPC variable for reading and writing:

- Minimum 12,4 ms
- Average 34,87 ms
- Maximum 75,4 ms

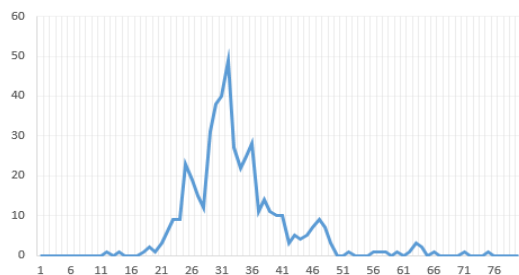


Figure 6. Delay time deterministic for WAK

For this case we can count the data transfer time between WAK server and webclient by difference between average full transfer time and transfer time to OPC Client.

$$2 \cdot T_{\text{WEBtr}} + T_{\text{webC}} = 34,87 - 31,85 = 3,02 \text{ ms}$$

**C. WebSocket**

WebSocket is a protocol providing full-duplex communications channels over a single TCP connection. The protocol consists of an opening handshake followed by basic message framing, layered over TCP. The goal of this technology is to provide a mechanism for browser-based applications that need two-way communication with servers that does not rely on opening multiple HTTP connections.

In this case a HTML page communicates with Java WebSocket-server using JSON-string telegrams. Such method of data transfer is used in project WOAS (WebOriented Automation System) [6]. As it is shown on the figure 7 the structure of the data transfer is similar to that of WAK. The main difference is only the WebSocket protocol instead of TCP-socket. In addition, WOAS kernel is more complicated than that of WAK.

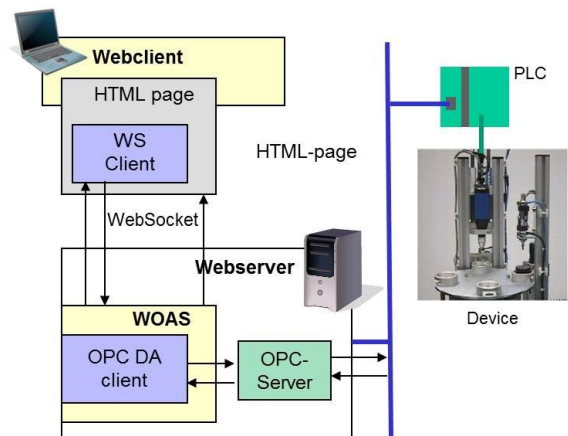


Figure 7. Structure of the data transfer with WAK

In this case, we can make a conclusion that 97% of signals have a delay time between 29 and 63 ms. The following times were measured for input frequencies  $f = 0.5$  Hz,  $f = 2$  Hz and  $f = 5$  Hz, each with an OPC variable for reading and writing:

- Minimum 21 ms
- Average 46,1 ms
- Maximum 81 ms

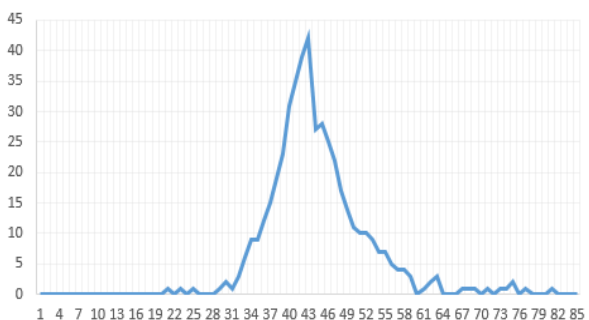


Figure 8. Delay time deterministic for WebSockets

In this case the data transfer time between WOAS server and webclient will be (Figure 8):

$$2 \cdot T_{\text{WEBtr}} + T_{\text{webC}} = 46,1 - 31,85 = 14,25 \text{ ms}$$

D. PHP

In this measurement, a PHP script is called cyclically for synchronous read or write of an OPC variable. It represents a classical polling with AJAX. That follows to the very high CPU load (approx. 70%). In this case, we can make a conclusion that 93% of signals have a delay time between 126 and 236 ms. This data transfer method gives The following times were measured for input frequencies  $f = 0.5 \text{ Hz}$ ,  $f = 2 \text{ Hz}$  and  $f = 5 \text{ Hz}$ , each with an OPC variable for reading and writing:

- Minimum 82 ms
- Average 160,17 ms
- Maximum 648 ms

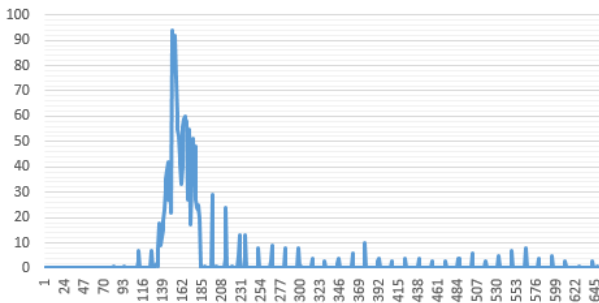


Figure 9. Delay time deterministic for PHP

In this case the data transfer time between server and webclient will be (Figure 9):

$$2 \cdot T_{\text{WEBtr}} + T_{\text{webC}} = 160,17 - 31,85 = 128,32 \text{ ms}$$

A comparison of time delays in ms by usage of 3 different methods is represented in table 1.

TABLE I.  
TIME DELAYS IN COMPARISON

	min	av.	max.
WAK	12,4	34,87	75,4
WS	21	46,1	81
PHP	82	160,17	648

Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance

dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.

IV. CONCLUSIONS

The common research and measurements of the existing data transfer methods has shown that also in a simple installation with no influence from outside the data transfer time is not a constant. At the moment it is important to define the scope of the data transfer technologies usage e.g. controls of real/non-real time processes, monitoring etc. The delay time of most of the methods do not allow using them in real time (e.g. motion control) but they could be used for monitoring or non-real time control. It is also possible to make real-time control directly on PLC and the processes that need lot of computing resources to transfer them using mentioned above data transfer methods to some server or cloud. A topic of time deterministic need further research, first of all the requirements of automation systems to the time deterministic should be formulated.

REFERENCES

- [1] Kagermann, H., Wahlster, W., Helbig, J., *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0-Abschlussbericht des Arbeitskreises Industrie 4.0*, 2012
- [2] Lange, J., Iwanitz F., Burke T., „*OPC Von Data Access bis Unified Architecture*“, VDE Verlag, Berlin, 2010
- [3] Iwanitz, F., DCOM versus Web-Services, *Computer& AUTOMATION*, 4/2004
- [4] Langmann, R., “Lean Web Automation in der Praxis“, Franzis Verlag GmbH, Poing, 2004
- [5] Langmann, R., *WEB Access Kit for Process Data (WAK) Java components for the Lean Web Automation - User’s handbook*, 2004
- [6] Langmann, R., Makarov, O., Meyer, L., *The WOAS project: Web-oriented Automation System*, REV2012: 9th International Conference on Remote Engineering and Virtual Instrumentations, 2012

AUTHORS

**O. Makarov** is with the Düsseldorf University of Applied Sciences, Düsseldorf, 40474 Germany and Odessa National Polytechnic University, Odessa, Ukraine (e-mail: makarov@ccad.eu) .

**R. Langmann and B. Frank** are with the Düsseldorf University of Applied Sciences, Düsseldorf, 40474 Germany (e-mail: langmann@ccad.eu).

**S.Nesterenko** is with Odessa National Polytechnic University, Odessa, Ukraine (sa\_nesterenko@ukr.net).

Submitted 30 April 2014. Published as re-submitted by the authors 08 June 2014.