Roman KWIECIEŃ[1], Leszek SZYCHTA[2]

Kazimierz Pulaski Technical University of Radom, Faculty of Transport and Electrical Engineering (1),
Kazimierz Pulaski Technical University of Radom, Faculty of Transport and Electrical Engineering (2)

Consequences of applying tunneling to OPC

Streszczenie. *Niniejszy artykuł przedstawia analizę pracy aplikacji komputerowych zbudowanych w technologii OPC, używanej jako standard komunikacyjny w systemach automatyki przemysłowej. Pomimo licznych zalet tej technologii ujawniają się również jej wady. Jednym ze sposobów eliminowania tych wad jest zastosowanie tunelowania. Jednakże tunelowanie technologii OPC przyczynia się do innych negatywnych skutków w pracy systemu komputerowego. W związku z tym w artykule autorzy pragną pokazać konsekwencje używania komputerowych aplikacji tunelujących w procesie przesyłania informacji przy pomocy sieci komputerowych. (**Konsekwencje używania komputerowych aplikacji tunelujących w procesie przesyłania informacji przy pomocy sieci komputerowych**)*

Abstract*. This paper presents an analysis of work computer applications built in OPC technology, used as a communication standard in industrial automation systems. Despite numerous advantages, this technology also has its defects. Tunneling is a way of eliminating these drawbacks. However, tunneling of OPC contributes to other adverse effects on the operation of a computer system. Therefore, the authors want to discuss consequences of use of tunneling computer applications for the process of information transfer via computer networks.*

Słowa kluczowe: Tunelowanie, OPC, OLE, COM, DCOM.
Keywords: Tunneling, OPC, OLE, COM, DCOM.

Introduction

OPC (OLE for Process Control) is a communication standard used in industrial automation systems. Its specifications describe methods of data transfer between information objects, which are part of the same or different computer units. Due to the disperse nature of the control system, computer networks are employed to transfer information as well. Standardization of data transfers is therefore a leading aspect of the concatenation process of wide-area computer systems [1,8,9,10].

OPC using COM/DCOM (Component Object Model/Distributed Component Object Model), introduced to Microsoft Windows operating system, has been termed OPC Classic. The technology exhibits certain defects in data transmission between computer units in a computer network. These defects are eliminated by means of tunneling or additional computer applications that mediate the information exchanges process. However, tunneling is not free from some drawbacks, either. In the circumstances, this article discusses operation of an OPC-based computer system using tunneling.

Origins of the OPC technology

Attempts at applying various measurement devices to monitor specific physical quantities, whose values provided information on correct and effective execution of technological processes, have been undertaken for a long time. Depending on technical and economic considerations, digital and analogue processors have been used, with their signals supplied to control panels including measurement devices (e.g. voltmeters, ammeters, thermometers, etc.) and signaling lamps.

Only the development of IT systems made a significant contribution to organization of a variety of applications of microprocessing, monitoring and measurement, and object equipment to monitor process status by means of synoptic screens. This led to elimination of measurement apparatus, which had taken plenty of space on operator panel decks.

Growth of computer technology resulted in development of IT networks and methods of organizing their operations. Application of one of the first industrial computer networks, Modbus, removed control systems to considerable distances and thereby eliminated complexities of wiring, which had carried analogue and two-state signals.

A range of IT systems were created for purposes of controlling engineering processes and data acquisition. Each business implementing its solutions was driven by other considerations in design of its entire system and management of appropriate information transfers. The fundamental difficulty consisted in various automatic devices not being compatible with a single common computer system. This was noted in 1995 by automation system leaders and manufacturers of microprocessing equipment, i.e. Fisher-Rosemount, Intellution, Intuitive Technology, Opto22, Rockwell, Siemens AG. At the time, OLE (Object Linking and Embedding) was the most advanced IT technology, which had already been used in various computer applications as part of Microsoft Windows. It was for that reason that this technology was selected to construct IT systems in industrial automation. OPC Task Force was formed to develop a communication standard between industrial equipment. This resulted in publication of the first OPC specification in August 1996, and OPC Foundation was established in September of the same year. Further coordinated actions were intended to maintain and publish new OPC specifications [2].

OLE technology

OLE (Object Linking and Embedding) gradually evolved away from DDE (Dynamic Data Exchange). DDE enables information and command exchange by notifying clients of appropriate changes in a server (Fig.1). Updating of a text document, where contents of a spreadsheet attached to this document change, is a typical example of DDE application [3,4,5].

In 1992, DDE mechanism was replaced with more efficient and more functional technology of embedding and linking documents (objects) in other OLE documents. OLE technology helps to exchange data between server and client applications including information concerning the server or references to certain information stored in the register of Windows. Microsoft extended OLE into OLE2 in 1993 and began adding new options, such as automation of OLE and OLE controls. Building of Windows 95 shell using OLE and interfaces was the next step. The name of OLE controls, known as OCX, was changed to ActiveX, and specifications were modified to distribute simple controls via the Internet.

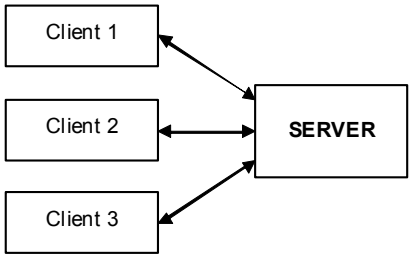

Fig. 1. Information exchange as part of client-server scheme

Due to steadily increasing importance of OLE for the Windows platform, Microsoft changed its name to COM and then COM+ for Windows 2000. These shifts in the nomenclature only party reflect technological changes and are largely driven by marketing considerations.

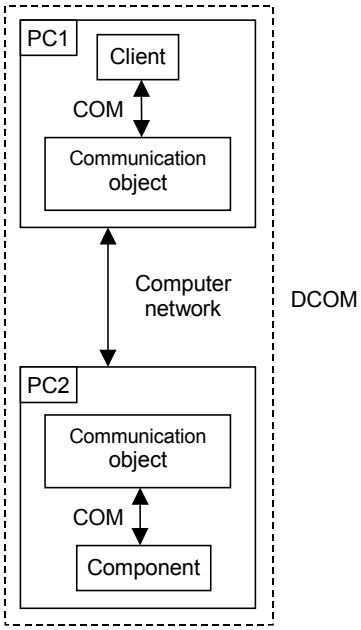

Fig.2. Information exchange between IT objects; PC1, PC2 – computer units.

Since information needs to be exchanged between computer units, the area of COM was extended to comprise communication objects operating as part of a computer network (Fig. 2), named DCOM (Distributed Component Object Model). It combines COM, where information is transferred as part of Windows, and data exchange via a computer network. Communication objects work in an operating system and mediate in data transmission between a client application and a component providing certain services. DCOM replaces local communication between processes with a network protocol using DCE RPC (Distributed Computing Environment / Remote Procedure Call). It does not matter to a client whether a server is located in the same or another computer unit.

Specification of OPC

OPC specifications were based on COM technology. Appropriate interfaces including call parameters were specified. They helped to define a standard of information exchange in computer applications operating as part of client-server arrangements. They define separate tasks for OPC servers with regard to their functionality and encompass:

- OPC Data Access (OPC DA) – provides access to current process data in real time,
- OPC Historical Data Access (OPC HDA) – provides access to archive data,
- OPC Alarms & Events (OPC A&E) – highlights events in a system and reported alarms,
- OPC Security – defines a method of access to data,
- OPC Batch – required for input management,
- OPC and XML – integrates OPC and XML (eXtensible Markup Language) in order to build Internet applications.

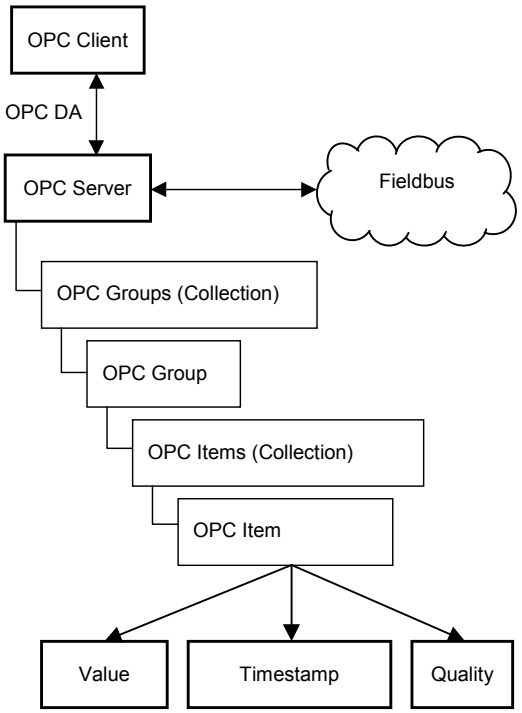

Fig. 3. Architecture of access to process variables via OPC DA

OPC DA specification provides access to a single process variable (OPC item) and reading or recording options, each of which has a value, a timestamp, type and quality, as depicted in Figure 3. The timestamp may be generated by a network point or OPC server if the point has no such capability. This specification can only enable viewing or changing of current process variables. As processes realized by the OPC server were complex, a logical division of process variables into groups was introduced. Variables in these groups are characterized by different scanning times and reading modes.

Data access by means of OPC DA can be implemented in three ways:

- using COM/DCOM,
- using XML (eXtensible Markup Language) and SOAP (Simple Object Access Protocol),
- by means of .NET Remoting (servicing of different formats and communication protocols, easy communication via the Internet).

Two modes of data reading are possible depending on OPC DA version:

- synchronous – reading always at identical intervals,
- asynchronous – reading takes place when certain data change – thresholds can be defined and reading must take place when they are exceeded.

Tunneling of Classic OPC

Introduction of the communication standard OPC contributed to: standardization of communication and industrial data exchanges, improved universality and scalability of solutions, and considerable reduction of

costs of integrating extensive industrial systems. In conditions of network operation, these authors have noted the following communication defects based on DCOM:

- difficulties with connection configuration,
- problems initiating connections,
- problems maintaining connection between a client and server.

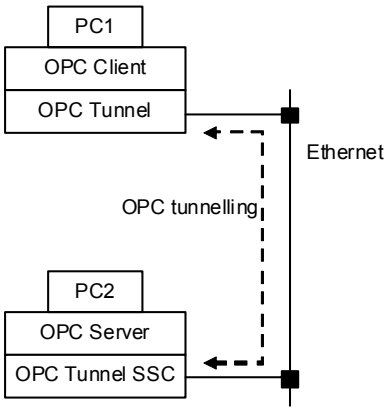

Fig. 4. Diagram of tunneling communication between an OPC client and OPC server by means of OPC Tunnel; OPC Tunnel SCS – OPC servers application; OPC Tunnel SSC – OPC clients application

Fig. 4 shows two computer units, PC1 and PC2, operating as parts of an Ethernet. They communicate between an OPC client and server applications in a classic scheme based on DCOM. These applications do not affect operation of IT objects, which are parts of DCOM and are dependent on their limitations as a result. To prevent communication problems of DCOM, tunneling applications are created to provide for information transfer that should be realized by DCOM. Applications of OPC SCS servers and SSC clients work in a client-server arrangement and exchange data via a defined port. They are installed on all computer units involved in data exchange by the computer network (OPC client and server). Their principal task is to convert OPC messages, collected by means of COM from a locally installed OPC application, into a relevant, productive standard of network communication, i.e. TCP/IP, HTTP, HTTPS, XML etc. Data are then dispatched via a computer network and the tunneling application OPC on the receiving computer executes the reconversion. In the circumstances, the tunneling application OPC carries out two tasks:

- it transfers data to another OPC tunneling object,
- it converts all data from OPC tunneling into the basic, standard format of OPC message.

Impact of Classic OPC tunneling on load-carrying capacity of Ethernet

Matrikon software served to determine load-carrying capacity of Ethernet employing Classic OPC (Fig. 4). A simulation OPC server, 'Matrikon Simulation Server', was worked on the computer unit PC1 while a process data viewer, 'Matrikon OPC Explorer', was started on PC2. A reading of a four-bite integer, Int4, generated at random by the OPC server, was assumed to range between 1000 and 10 000. The experiment was carried out in the mode of synchronous and asynchronous process data reading, with present and absent tunneling. Comparative characteristics are shown in Figure 5. The diagram indicates that the load-carrying capacity of Ethernet is linear, which is proof that the load-carrying capacity of the computer network is proportional to the quantity of process data transmitted between a client and an OPC server. The load-carrying capacity of Ethernet is highest In the case of synchronous transmission of process data without tunneling (characteristic a). This is due to the fact that, in synchronous data reading, information is transmitted as a whole at equal intervals while only the data which have changed within a specific range are as part of asynchronous reading. Tunneling reduced Ethernet load by approx. 40%, though loading of the computer unit processor increased roughly 4 times, unfortunately (Fig. 6). This is caused by operations of tunneling applications that mediate in data transmission process. The increased load-carrying capacity of the processor also raises power consumption of computer units from the supply network.

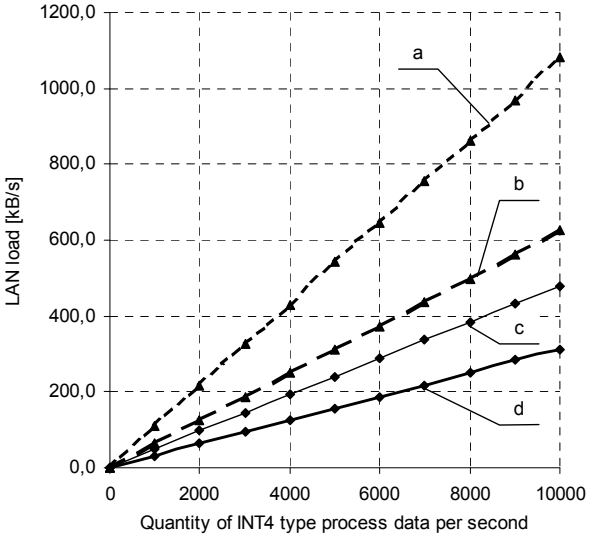

Fig. 5. Characteristics of LAN loading executed by means of OPC Martikon operating: a) synchronously without tunneling; b) synchronously with tunneling; c) asynchronously without tunneling; d) asynchronously with tunneling

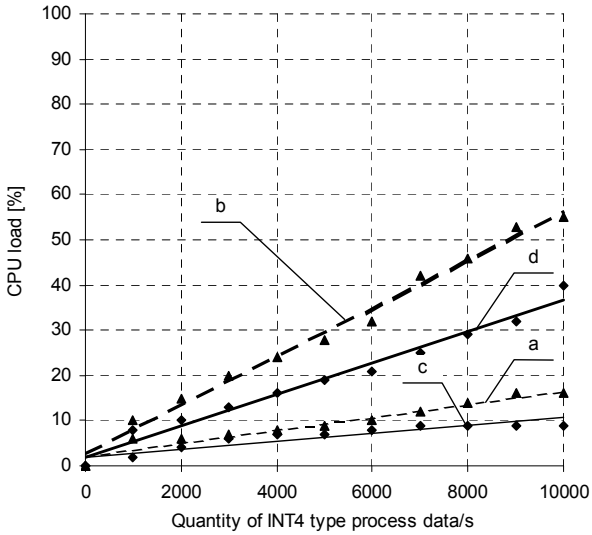

Fig. 6. Characteristics of computer unit processor loading executed by means of OPC Martikon operating: a) synchronously without tunneling; b) synchronously with tunneling; c) asynchronously without tunneling; d) asynchronously with tunneling

Application of tunneling to data transmission eliminates drawbacks of DCOM while giving rise to some other defects, such as [7,11]:

- invalidation of names and passwords of operating system users since any computer unit realizing

connections from any user account has access to tunneling servers,
- additional loading of computer unit processors (even by an order of magnitude) since tunneling applications are involved in message translation and data transmission as part of Ethernet.

Conclusion

OPC communication standard made a significant contribution to integration of extensive computer systems. Manufacturers of automatic equipment additionally make computer server applications tasked with control of their products and reading of their process data. Other computer applications, working as OPC clients, can connect with a required OPC server to gain control over a selected monitoring and measurement device. In this way, image of an engineering process is organized as a two or three-dimensional presentation which takes advantage of all media of state-of-the-art information processing technologies, spatial identification of a visualized object, verbal communication with an image object of an engineering process.

Drawbacks of OPC described here and methods of their elimination are often key to attaining desired goals in automatic systems. Shortcomings of OPC Classic, as well as emergence of operating systems other than Microsoft Windows, gave rise to a new communication standard termed OPC UA (Unified Architecture). Specification of OPC UA was introduced by OPC Foundation in January 2007. Functionally, it defines methods of realising three older specifications: OPC DA, OPC HDA and OPC A&E. It bases on commonly accepted communication protocols, such as TCP/IP (Transmission Control Protocol / Internet Protocol), HTTP (Hypertext Transfer Protocol), SOAP (Simple Object Access Protocol). Thus, the new standard is completely independent from DCOM, cumbersome as it is when it comes to configuration of a computer system [6].

Classic OPC continues to be used in industrial automatic systems although new data transmission formats are developed. This is due to the fact that the existing solutions of appropriately configured computer control systems are well tested. Certain tool software has been based on Classic OPC technology to generate designs for monitoring, visualisation, control, and process data acquisition in a production plant. In addition, OPC UA organises operations of OPC server in a different manner and is still being tested. This consideration is often decisive in assessments of applicability of communication standards. There are situations where most recent operating systems and the associated tool software are intentionally avoided. Therefore, designers of computer control systems must take into account consequences of applying tunneling to Classic OPC technology.

REFERENCES

[1] Chrupek R., Akwizycja Danych w systemach przemysłowych, *Napędy i sterowanie*, nr 4, kwiecień 2008r.

[2] The OPC Foundation: www.opcfoundation.org

[3] Iwanitz F., Lange J.: OLE for Process Control. Fundamentals, Implementation and Applications, *Huthig Verlag heiderberg*, RFN, 2001.

[4] SOAP: http://www.w3.org/TR/soap/

[5] DCOM Technical Overview, *Microsoft Developer Network*. http://www.microsoft.com/com/

[6] Zbrzezny M., Infrastruktura komunikacyjno-usługowa OPC Unified Architecture (OPC UA). 23 luty 2009, *Programowanie i Technologie*: http://maciej-progtech.blogspot.com

[7] Kwiecień R., Szychta E., Szychta L., Data acquisition in OPC-based industrial IT systems, *The 4TH international conference on electrical and control technologies,* ECT 2009, ISSN 1822-5934.

[8] Postół M., Platforma integracji systemów zarządzania z produkcją (cz. 2). Głównie dla orłów, *Control Engineeging Polska*, Październik 2008, str. 16-24.

[9] Skura K., Zagadnienia integracji systemów informatycznych w automatyzacji procesów produkcyjnych w oparciu o technologię OPC. *Napędy i sterowanie*, nr 10, Październik 2007r.

[10] Postół M., Platforma integracji systemów zarządzania z produkcją (cz. 1). W poszukiwaniu złotego środka, *Control Engineeging Polska*, Wrzesień 2008, str. 16-22.

[11] Kwiecień R., Szychta L., Figura R.: Skryptowy informatyczny system sterowania urządzeniami automatyki przemysłowej, Przegląd Elektrotechniczny Electrical Review, s. 285 – 288, nr 2/2010, (ISSN 0033-2097)

[12] Mahnke W., Leitner S.-H.: Damm M., OPC Unified Architecture, ISBN 978-3-540-68898-3, 2009 Springer-Verlag Berlin Heidelberg

***Authors**: dr Roman Kwiecień, Eng., Technical University of Radom, Institute of Transportation Systems and Electrical Engineering, ul. Malczewskiego 29, 26-600 Radom, E-mail: r.kwiecień@pr.radom.pl;*

prof. Leszek Szychta, Eng., Technical University of Radom, Institute of Transportation Systems and Electrical Engineering, ul. Malczewskiego 29, 26-600 Radom, E-mail: l.szychta@pr.radom.pl