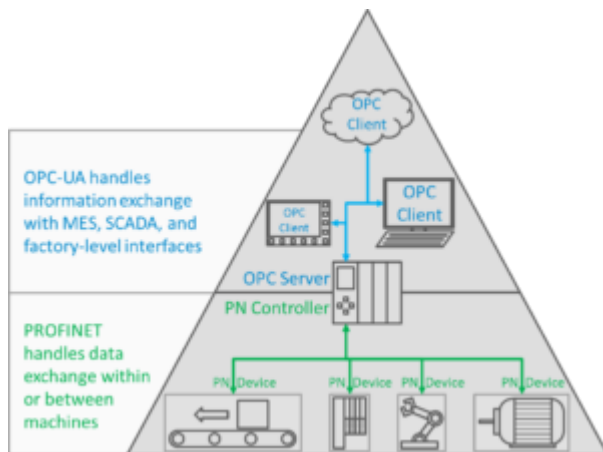# OPC UA and PROFINET

P ROFINET and OPC UA are two common protocols that have some overlap in the automation and process industries, and understanding which protocol to use in a particular part of a network can be confusing. Should a new plant use PROFINET or OPC UA to implement local control loops? What about performance data for an automation cell? Which protocol is the best fit to transmit real-time production status up to the corporate office?

To answer these questions, you have to understand some of the low-level differences between the protocols. Both PROFINET and OPC UA exchange the same types of data, but they do it in two very different ways. PROFINET exchanges time-critical data in a tightly controlled format, designed to facilitate fast processing, minimize processor overhead and allow control loops with relatively small time constants (~1ms) to run across the network. OPC UA, however, can exchange nearly any type of information. It was designed from the ground up to enable flexible communication, at the expense of slower processing, higher latency and more processor overhead.

Historically, the two protocols have had two distinct roles. PROFINET is typically used for real-time data communication between field devices and local controllers. Conversely, OPC UA is usually used to communicate between those controllers and higher-level historians, MES, and SCADA systems.

OPC UA Network Hierarchy

PROFINET networks consist of Controllers and Devices. Similarly, OPC UA networks consist of Clients and Servers. A Server provides some data, and Clients consume the data. Usually, that means that a PROFINET Controller may also implement an OPC UA Server to pass data up to OPC UA Clients, such as HMIs, Engineering Systems, or even the cloud:

OPC UA and PROFINET address two different needs in the industrial communication ecosystem. OPC UA used for system monitoring and diagnostics, while PROFINET is used to implement local control loops and processing.

However, that clear distinction starts to blur when PROFINET devices implement their own OPC UA Servers, and PROFINET Controllers implement OPC UA Clients in addition to Servers. Interconnection from the HMI or Engineering system directly to field devices becomes possible, and the clear hierarchy of the automation system starts to get more confusing:
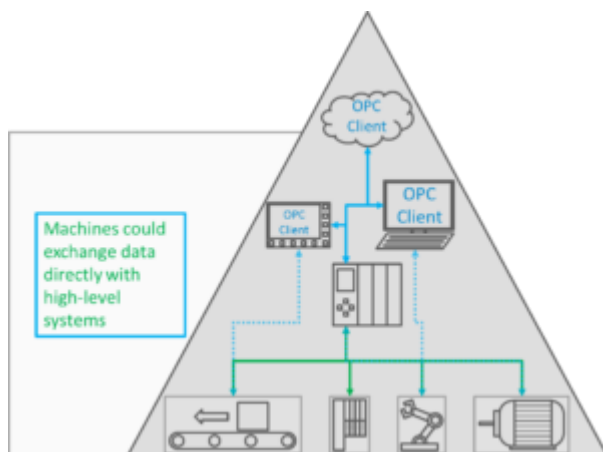
Pervasive implementation of OPC UA Clients and Servers blurs the line between local and remote control systems.

Real Performance Differences

This confusion is at the heart of questions about OPC UA. Is it an automation protocol? Is it a SCADA protocol? Can it exchange real-time data and information? Will it replace PROFINET, Ethernet/IP, SERCOS, EtherCAT, or any other Ethernet-based automation protocol? The answer to each of these questions is "maybe," but in some cases, "not likely" is a better answer. Let's take a look at what both OPC UA and PROFINET do well to draw a line between their use cases:

	PROFINET	**OPC UA**
Cycle Time	Typically between 0.125 - 512 ms	~200ms, depending on server capabilities.
Jitter	0.001 - 1 milliseconds	10-100 milliseconds
Data Format	Tightly-defined data format with only a few types of structures.	Object-oriented data format with the flexibility to match almost any type of data.
Network Boundaries	PROFINET is not forwarded by routers and cannot cross between networks.	OPC UA is a routed protocol and can be used between networks, or even over the internet.
Cyclic Data Exchange	Guaranteed Performance	Best Effort Performance

Looking at the data, OPC UA doesn't make the grade for a true real-time control protocol. It's just too slow to implement control loops for most motion applications. So while you may see OPC UA appear on datasheets for IO Devices alongside PROFINET, it's a feature that fills a distinctly different role. OPC UA is great for moving information to higher-level systems, but PROFINET forms the backbone for distributed I/O and control.