OPEN ACCESS

International Journal Of Modern Engineering Research (IJMER)

Optimizing Industrial Communication: Comparing Fieldbus Protocols

Gordana Ostojic[1], Stevan Stankovski[1]

[1]PhD, Full professor, University of Novi Sad, Faculty of Technical Sciences

Corresponding Author: Gordana Ostojic

ABSTRACT: *In modern industrial automation, efficient and reliable communication between devices and systems is crucial for productivity, flexibility, and safety. Fieldbus protocols have become the backbone of industrial communication, enabling real-time data exchange across sensors, actuators, controllers, and supervisory systems. This paper presents a comparative analysis of prominent Fieldbus protocols—including Profibus-DP, CANopen, DeviceNet, Modbus RTU, Modbus TCP, and AS-Interface—with the objective of identifying their strengths, limitations, and optimal use cases. Key parameters such as communication speed, topology, determinism, and interoperability are examined. The analysis highlights how each protocol addresses the needs of different industrial environments, from time-critical applications to simple device connectivity. A comparative table is provided to support decision-making when selecting a communication standard tailored to specific industrial scenarios. Ultimately, this paper aims to guide engineers, integrators, and system designers toward choices that enhance system performance and integration in industrial networks.*

KEY WORDS: *Protocols, Data Rate, Topology, Openness, Communication Channels*

Date of Submission: 11-07-2025 Date of acceptance: 24-07-2025

I. INTRODUCTION

In the rapidly evolving landscape of industrial automation, the choice of communication protocol plays a pivotal role in ensuring reliable, real-time data exchange between field devices, controllers, and supervisory systems [1]. As automation systems become more complex and interconnected, efficient communication becomes a key requirement for maintaining performance, flexibility, and scalability.

Fieldbus communication protocols were introduced to replace traditional point-to-point wiring systems, offering reduced complexity, improved diagnostics, and better integration of devices. Over the years, a wide array of industrial communication standards has emerged, each tailored to specific application domains, performance requirements, and network architectures.

This paper presents a comparative study of six widely adopted fieldbus protocols: Profibus-DP (DP Decentralized Periphery), CANopen, DeviceNet, Modbus RTU (Remote Terminal Unit), Modbus TCP (Transmission Control Protocol), and AS-Interface (Actuator-Sensor Interface). These protocols represent diverse design philosophies and technical specifications, varying significantly in terms of data rates, topology, determinism, real-time performance, ease of configuration, and interoperability. Profibus-DP, developed by Siemens and widely adopted in process and factory automation, offers high-speed deterministic communication and strong diagnostic capabilities [2]. CANopen, based on the Controller Area Network (CAN) standard, is optimized for modular devices and real-time embedded applications with robust error handling [3]. DeviceNet, also built on CAN, is tailored for industrial device-level communication, emphasizing plug-and-play configuration and multi-vendor interoperability [4]. Modbus RTU, one of the oldest and simplest protocols, operates over serial links and remains popular for its ease of implementation and low overhead [5]. Modbus TCP, the Ethernet-based extension of Modbus, offers compatibility with IT infrastructures and greater bandwidth, enabling broader integration [6]. AS-Interface focuses on simple, cost-effective connections for binary sensors and actuators in the lowest field level, with minimal wiring and quick setup [7].

The objective of this paper is to provide a side-by-side comparison of these protocols based on key performance and technical criteria. By analyzing their strengths, limitations, and suitable application scenarios, the paper aims to guide engineers, system integrators, and decision-makers in selecting the most appropriate communication protocol for their specific automation needs.

Through this comparative approach, the study highlights how the proper selection of a fieldbus protocol can impact not only system performance but also lifecycle costs, maintainability, and future scalability of industrial networks.

II. FIELDBUS PROTOCOLS

Fieldbus protocols that are going to be analyzed are protocols mostly used in industrial applications. Protocols are: Profibus-DP, CANopen, DeviceNet, Modbus RTU, Modbus TCP and AS-interface.

Profibus-DP operates primarily at the field level of automation architectures, where it connects programmable logic controllers (PLCs), distributed I/O modules, sensors, and actuators. It follows the master-slave communication model, where a central controller (master) manages communication with one or more field devices (slaves). Multiple masters can exist in a network, but each segment can only have one active master at a time.

The communication protocol is based on the RS-485 standard, which provides robust, noise-resistant communication over twisted-pair cables. This allows for transmission rates of up to 12 Mbps, depending on the cable length (e.g., 12 Mbps at up to 100 meters). The network can support up to 126 devices per segment, including masters and slaves, although repeaters and segment couplers can be used to increase distance and device count [2].

Profibus-DP uses cyclical data exchange for real-time control and acyclical communication for configuration, diagnostics, and parameterization. During the cyclic operation, the master continuously polls the slaves in a predetermined sequence, exchanging small packets of process data. This ensures deterministic behavior, which is crucial in time-sensitive industrial applications such as motion control or high-speed production lines [8].

The acyclic communication channel runs in parallel and enables advanced functions like device diagnostics, firmware updates, or parameter changes without interrupting the cyclic data flow. This dual-channel design contributes to Profibus-DP's efficiency and flexibility in complex industrial environments.

Despite its advantages, Profibus-DP has some limitations. As a serial, master-slave protocol, it lacks the bandwidth and flexibility of newer Ethernet-based systems. Additionally, the physical limitations imposed by RS-485 restrict maximum network length and the number of devices per segment. Nevertheless, Profibus-DP remains in widespread use due to its reliability, real-time capabilities, and massive installed base.

CANopen has a complete communication architecture that includes device profiles, communication services, and application layers. This makes it much easier to develop interoperable and reliable systems.

One of the key features of CANopen is its communication model, which includes several types of messages optimized for different purposes. Process Data Objects (PDOs) are used for real-time data exchange. These are short, time-critical messages (up to 8 bytes) that can be sent cyclically, based on events, or synchronized to a common time base. For configuration, diagnostics, and other less time-sensitive tasks, Service Data Objects (SDOs) are used. These allow devices to read from or write to one another's memory using a client-server model. Additional message types include emergency messages (EMCY), used to signal critical faults, and synchronization messages (SYNC), which coordinate the timing of PDO transmissions [3].

Each CANopen device contains an internal data structure known as the Object Dictionary. This is a standardized list of all the device's parameters, such as configuration settings, process variables, and status indicators. The Object Dictionary provides a consistent way for other devices or control systems to interact with a device, regardless of its specific function or manufacturer. Parameters are accessed by their 16-bit index and subindex, making the system both organized and highly flexible.

To ensure compatibility between devices, CANopen defines a number of device profiles. These profiles specify how certain types of devices—such as digital I/O modules, motion controllers, or encoders—should behave and communicate on the network. For example, CiA 401 defines standard functionality for general-purpose I/O devices, while CiA 402 covers drives and motion control systems. Application profiles also exist to define entire network behaviors for specific sectors, such as elevators (CiA 417) or maritime electronics (CiA 422) [9].

CANopen has many advantages. It is lightweight and efficient, which is especially important in systems with limited computing resources. It supports real-time communication, making it suitable for motion control and other time-sensitive applications. It is also very reliable, with built-in error handling and diagnostic capabilities, including heartbeat messages, node guarding, and emergency messages. Furthermore, CANopen is scalable, allowing networks to range from just a few nodes to several dozen, and its flexibility in topology and device configuration means it can be adapted to a wide range of use cases.

However, CANopen is not without limitations. The maximum payload of 8 bytes per message can be restrictive for data-heavy applications, and the maximum communication speed of 1 Mbps may be insufficient for high-bandwidth needs. In addition, the physical characteristics of the CAN bus limit network length, especially at higher speeds—for instance, at 1 Mbps, the recommended bus length is around 40 meters. For these reasons, newer Ethernet-based protocols are becoming more common in applications that require high data throughput and long distances.

DeviceNet uses the CAN protocol (ISO 11898) at its core but adds a higher-layer protocol for defining how data is structured, how devices are configured, and how communication is managed. It operates on a producer-consumer model rather than a strict master-slave model, allowing more flexibility and better efficiency in communication. This model enables a device to produce a message once and have it consumed by multiple devices that need it, rather than sending it multiple times to each destination [4].

The communication speed of DeviceNet can be set to 125 kbps, 250 kbps, or 500 kbps. However, the maximum cable length depends on the speed selected. For example, at 125 kbps, the network can span up to 500 meters; at 500 kbps, the maximum length is reduced to about 100 meters.

DeviceNet supports cyclical, change-of-state, and polled messaging, providing flexible options for real-time data transmission. Cyclical messaging sends data at regular intervals, change-of-state only sends updates when values change, and polled messaging allows the controller to request data when needed [10].

DeviceNet offers several significant advantages that have contributed to its widespread adoption in industrial automation, particularly at the sensor and actuator level. One of its most notable benefits is the ability to carry both communication and power over a single five-wire cable. This simplifies wiring, reduces installation time, and lowers overall system cost. By eliminating the need for separate power and communication lines, DeviceNet minimizes the complexity often associated with field device connectivity.

DeviceNet also benefits from the robust and reliable CAN physical layer, which provides high noise immunity and error detection capabilities. The protocol uses mechanisms like cyclic redundancy checks (CRC), message acknowledgment, and retries to maintain data integrity, making it highly reliable even in electrically noisy industrial environments.

Standardization is another strength of DeviceNet. Its object-oriented architecture, including consistent data structures across devices, facilitates easier device integration, configuration, and diagnostics. Tools that use Electronic Data Sheets (EDS) can automatically recognize and configure devices, reducing commissioning time and simplifying maintenance. In addition, ODVA certification ensures that DeviceNet-compliant devices from different manufacturers can interoperate smoothly, giving users a wide choice of compatible products [11].

Despite its strengths, DeviceNet has some limitations that must be considered during system design. One of the primary drawbacks is its limited communication speed and bandwidth, so DeviceNet is not suitable for applications that require high-speed data transfer or the transmission of large amounts of data. As a result, it is not ideal for vision systems, advanced motion control, or complex data analysis applications.

Another limitation is network length, which is directly affected by the selected communication speed. This restriction can pose challenges in large facilities, unless repeaters or additional segments are used, which adds cost and complexity.

Scalability is also a concern. While DeviceNet supports up to 64 nodes per network segment, expanding beyond that requires careful planning and additional infrastructure. Compared to modern Ethernet-based protocols, which support thousands of nodes and much higher data rates, DeviceNet may not meet the scalability requirements of large or fast-growing systems.

Modbus RTU operates over serial communication lines, typically RS-485, though RS-232 and RS-422 can also be used. RS-485 is preferred in most applications due to its ability to support multiple devices on the same communication line (multi-drop), noise immunity, and longer cable lengths—up to 1,200 meters depending on baud rate and cable quality.

Modbus RTU uses a master-slave architecture, where a single master controls communication on the network and one or more slave devices respond to the master's requests. Only the master can initiate communication, and slaves respond only when addressed. This simple model makes implementation straightforward but limits flexibility and peer-to-peer communication.

Each device on the Modbus network is assigned a unique slave address (ranging from 1 to 247), and the master uses this address to direct queries to a specific device. Communication occurs using binary (RTU) frames, which are compact and efficient.

One of the primary advantages of Modbus RTU is its simplicity and openness. It is not proprietary, and the protocol specifications are freely available, which has led to widespread support by vendors and integrators. It is relatively easy to implement on microcontrollers and embedded devices, making it a practical choice for both low-cost and complex systems.

Modbus RTU is also lightweight, with low overhead and small packet sizes. This makes it efficient for slower serial lines and suitable for applications where bandwidth is limited. Its use of binary encoding (RTU) is more compact and faster than the ASCII version of Modbus.

Another key benefit is its reliability and robustness in noisy environments. When implemented over RS-485 with differential signaling, it can operate effectively over long distances and in industrial environments prone to electrical interference.

Despite its strengths, Modbus RTU has several limitations. The master-slave model means that communication is strictly one-way initiated from the master, and slaves cannot initiate data transfers. This makes it unsuitable for event-driven or peer-to-peer communication.

Modbus RTU is also limited in bandwidth and speed, typically supporting baud rates from 1,200 to 115,200 bps. While sufficient for many applications, it does not scale well for high-speed data acquisition or systems with many devices and frequent updates [5].

Another drawback is the lack of a standard data model—each device may use different register mappings, requiring custom configuration and documentation. Additionally, Modbus RTU does not support advanced features like device discovery, time synchronization, or diagnostics, which are common in more modern protocols.

Modbus TCP (also known as Modbus TCP/IP) is a network protocol used in industrial automation systems for communication over Ethernet. It is an extension of the traditional Modbus protocol, which was originally developed for serial communication. While Modbus RTU runs over serial lines like RS-485 or RS-232, Modbus TCP operates over standard Ethernet networks using the TCP/IP protocol stack. This makes it a modern and flexible solution for integrating field devices, controllers, SCADA (Supervisory Control and Data Acquisition) systems, and human-machine interfaces (HMIs) into IP-based networks.

Modbus TCP follows the client-server model (equivalent to the master-slave model in Modbus RTU), where the client (often a PLC, SCADA, or HMI) initiates requests and the server (a field device such as a sensor, actuator, or I/O module) responds. What makes Modbus TCP particularly appealing is that it retains the core structure and function codes of the original Modbus protocol, which ensures compatibility and ease of migration from serial to Ethernet-based systems.

Modbus TCP uses TCP port 502 for communication and encapsulates Modbus frames within a standard TCP/IP packet. The Modbus Application Data Unit (ADU) consists of: MBAP (Modbus Application Protocol) Header that contains transaction ID, protocol ID, length, and unit ID (replacing the slave address in RTU); and PDU (Protocol Data Unit) that contains the function code and associated data (same as in Modbus RTU) [12].

Because Modbus TCP uses the full Ethernet stack, it can be integrated into existing LANs and WANs. Devices can be connected via switches, routers, and even wireless access points, allowing for easy scalability and remote access capabilities.

Modbus TCP benefits from using standard Ethernet networks, allowing easy integration with existing infrastructure and eliminating the need for special converters. It offers higher communication speeds compared to serial Modbus, enabling faster data exchange and better scalability across large or distributed systems. Its simplicity and wide vendor support make it easy to implement, and remote access capabilities facilitate monitoring and control from anywhere.

However, Modbus TCP lacks built-in security features like encryption and authentication, making networks vulnerable without external protection. It also uses a simple request-response model without advanced communication functions such as event-driven messaging or multicast support. Additionally, inconsistent device data mapping requires manual configuration, and network performance can suffer from latency or congestion in large systems.

AS-Interface focuses on the lowest level of automation, where sensors and actuators operate. Its primary goal is to simplify the wiring and communication between these devices and the control system, thereby reducing installation costs, minimizing wiring complexity, and speeding up commissioning and maintenance.

It is mainly used in applications such as conveyor systems, packaging machines, automotive assembly lines, and other machinery that involve a large number of binary input/output devices.

One of AS-Interface's key features is its unique two-conductor flat cable that carries both power and data simultaneously. This simple cabling system drastically reduces wiring effort compared to traditional point-to-point wiring, where each sensor or actuator would need individual power and signal wires.

The standard AS-Interface cable consists of two flat copper conductors arranged in parallel, enclosed in a durable sheath. It supports distances up to 100 meters per segment and can be extended up to 300 meters using repeaters or bridges. The cable topology is typically a simple bus or tree structure, which makes expansion and troubleshooting straightforward [7].

Power (up to 30 V DC) and communication signals share the same cable, with a maximum current of around 2 A available per segment, sufficient for powering a wide range of field devices directly from the AS-i network.

AS-Interface uses a master-slave communication protocol in which the master controller manages communication with multiple slave devices (sensors and actuators). Typically, one AS-i master can manage up to 62 slave devices per segment, although this number can be increased by adding more segments.

Data transmission is serial and occurs in a time-multiplexed manner over the two-wire cable at a speed of 167 kbps. Each slave device is assigned a unique address (1 to 62), allowing the master to poll devices cyclically for their input states and to send output commands.

The protocol supports digital I/O data exchange and can also handle simple analog values with additional addressing techniques or special devices. The data frames are short and deterministic, making AS-Interface suitable for real-time control of sensors and actuators.

AS-Interface is often integrated as a lower-level network connected to higher-level industrial networks such as Modbus TCP. Gateways and bridges translate AS-i signals to these protocols, enabling seamless integration into broader automation systems.

The simplicity, reliability, and low cost of AS-Interface have made it popular worldwide, especially in discrete manufacturing and material handling sectors.

AS-Interface offers a simple and cost-effective solution for connecting sensors and actuators by using a unique two-wire cable that carries both power and data, greatly reducing wiring complexity and installation time. It supports up to 62 devices per segment, provides reliable real-time communication, and features easy device installation and diagnostics. Its robustness makes it suitable for harsh industrial environments, and it integrates well with higher-level networks via gateways [7].

However, AS-Interface is limited to mostly binary signals and low-speed communication (167 kbps), making it unsuitable for complex or high-data applications. Its cable length and device count per segment impose scalability constraints, and it is not designed to support advanced devices like drives or vision systems.

III. DISCUSSION AND CONCLUSION

In order to better understand the capabilities and limitations of various industrial communication protocols, a structured comparison has been conducted. The selected protocols — Profibus-DP, CANopen, DeviceNet, Modbus RTU, Modbus TCP, and AS-Interface — are among the most commonly used in modern automation systems, each with unique characteristics suited for different applications and environments.

This comparative analysis focuses on several key parameters that influence protocol selection and network performance, including: data rate, topology, deterministic protocol type, openness, and typical application areas (Table 1).

By organizing these characteristics into a comparative table, similarities and differences between the protocols become more apparent, allowing for clearer insights into which protocol is most appropriate for specific industrial scenarios. The table serves as a practical tool for engineers and system designers to evaluate trade-offs between performance, complexity, and scalability.

***Table I.* Parameters of fieldbus protocols**

Protocol	Data Rate	Topology	Deterministic	Openness	Typical Application Areas
Profibus-DP	Up to 12 Mbps	Bus, Tree	Yes	Open (IEC 61158)	Factory automation, drives
CANopen	Up to 1 Mbps	Bus	Yes	Open	Embedded systems, robotics, mobile machinery
DeviceNet	Up to 500 kbps	Trunk & drop	Moderate	Open	Sensors and actuators
Modbus RTU	9600 bps – 115.2 kbps	Line (RS-485)	No	Open	Low-speed industrial devices
Modbus TCP	Ethernet (10/100 Mbps)	Star (Ethernet)	No	Open	Industrial Ethernet devices
AS-Interface	167 kbps	Line, Tree	Limited	Open	Simple sensors/actuators

Profibus-DP is a mature, robust fieldbus protocol tailored for fast, reliable communication between central controllers and field-level devices. Its real-time capabilities, standardized integration, and diagnostic features have made it a cornerstone in industrial automation for over three decades. While Ethernet-based protocols are gaining ground, Profibus-DP continues to offer significant value in legacy systems and applications where real-time, deterministic communication remains a priority.

CANopen is a reliable and efficient fieldbus protocol ideal for real-time, distributed control in embedded and industrial systems. Its lightweight architecture, standardized device profiles, and robust error handling make it well-suited for applications requiring deterministic communication and low resource usage. While limited in data throughput compared to Ethernet-based protocols, CANopen remains a popular choice for automation systems where simplicity, reliability, and interoperability are key.

DeviceNet is a robust, cost-effective communication protocol designed for industrial automation at the device level. Its combination of CAN-based reliability, integrated power and communication, and standardized device structure makes it ideal for connecting sensors, actuators, and control devices on the factory floor. Although newer technologies offer higher speeds and scalability, DeviceNet remains a dependable choice for applications where simplicity, reliability, and cost-efficiency are critical.

Modbus RTU remains one of the most widely used and supported communication protocols in industrial automation due to its simplicity, low cost, and open nature. It is especially effective in small to medium-sized systems that require basic data exchange over serial lines. While it may lack the features and scalability of modern Ethernet-based protocols, its ease of use, reliability, and extensive vendor support ensure that Modbus RTU continues to be a practical choice in many industrial environments.

Modbus TCP is a powerful and flexible evolution of the classic Modbus protocol, bringing the simplicity and reliability of Modbus to modern Ethernet networks. With higher speed, easier scalability, and widespread support, it plays a critical role in bridging traditional industrial devices with modern control systems and IoT platforms. While it lacks native security and some advanced features found in newer protocols, its openness, ease of use, and broad adoption continue to make it a trusted solution in industrial communication.

AS-Interface is a specialized, cost-effective fieldbus designed for connecting sensors and actuators in industrial automation. Its unique two-wire cable that carries both power and data simplifies installation, reduces costs, and supports reliable real-time communication for up to 62 devices per segment. While its data capacity and speed are limited compared to more complex protocols, its ease of use, robustness, and wide vendor support make AS-Interface an enduring choice for low-level device networking in many industrial applications.

ACKNOWLEDGMENT

The research for this paper was funded by the project „Unapređenje kvaliteta nastave na sudijskim programima Departmana kroz implementaciju rezultata naučno istraživačkog rada u oblasti Industrijskog inženjerstva i menadžmenta“.

REFERENCES

[1]. Ostojić, G. and Stankovski, S., (2024). “IoT Protocols and Cybersecurity Threats”, Journal of Mechatronics, Automation and Identification Technology – JMAIT, 9 (1), 1-4.

[2]. Vitturi, S., (2003). “DP-Ethernet: the Profibus DP protocol implemented on Ethernet”, Computer Communications, 26 (10), 1095-1104.

[3]. Barbosa, M., Farsi, M., Allen, C. and Carvalho, A.S., (2003). “Formal validation of the CANopen communication protocol”, IFAC Proceedings Volumes, 36 (13), 225-230.

[4]. Biegacki, S. and VanGompel, D., (1996). “The application of DeviceNet in process control”, ISA Transactions, 35 (2), 1996, 169-176.

[5]. Urrea, C., Morales, C. and Muñoz, R., (2016). “Design and implementation of an error detection and correction method compatible with MODBUS-RTU by means of systematic codes”, Measurement, 91, 266-275.

[6]. Machaka, V., Figueroa-Lorenzo, S., Arrizabalaga, S., Elduayen-Echave, B. and Hernantes, J., (2025). “Assessing the impact of Modbus/TCP protocol attacks on critical infrastructure: WWTP case study”, Computers and Electrical Engineering, 126, 110485.

[7]. https://www.rtautomation.com/technologies/as-interface/

[8]. Vitturi, S., (2004). “On the effects of the acyclic traffic on Profibus DP networks”, Computer Standards & Interfaces, 26 (2), 131-144.

[9]. https://www.can-cia.org/can-knowledge/cia-402-series-canopen-device-profile-for-drives-and-motion-control

[10]. Reynders, D., Mackay, S., Wright, E. and Mackay, S., (2004). “13a - DeviceNet overview”, Editor(s): Deon Reynders, Steve Mackay, Edwin Wright, Steve Mackay, Practical Industrial Data Communications, Butterworth-Heinemann, 279-295.

[11]. https://www.odva.org/technology-standards/document-library/

[12]. Urrea, C., Morales, C. and Kern, J., (2016). “Implementation of error detection and correction in the Modbus-RTU serial protocol”, International Journal of Critical Infrastructure Protection, 15, 27-37.