Profinet

---

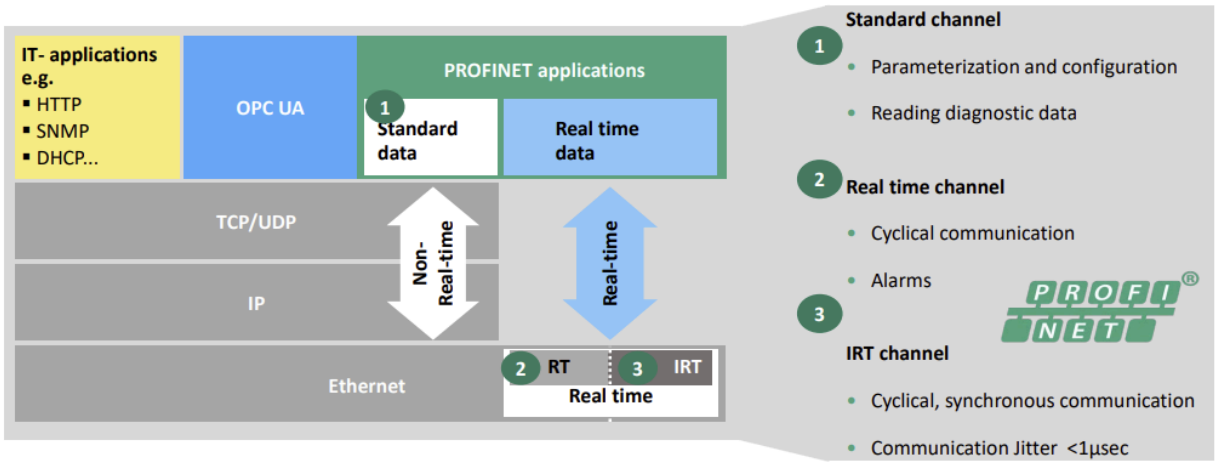
1. Introduction to Profinet:

PROFINET is the industrial Ethernet standard used widely in factory and process automation to connect PLCs, I/O modules, drives, HMIs and gateways. It brings Ethernet/IT concepts to the shopfloor while adding real-time and deterministic features required by automation systems.

Deterministic communication means delivering messages exactly when they are expected. PROFINET must ensure messages are delivered with the appropriate speed and determinism depending on the task

Profinet Delivers data through three communication channels listed below

- TCP/IP (or UDP/IP)
- PROFINET Real-Time (RT)
- PROFINET Isochronous Real-Time (IRT)
- Time Sensitive Networking (TSN)



Source:

https://www.profibus.com/index.php?eID=dumpFile&t=f&f=53290&token=6c8f55bcf01604eb71efb19902a1209c5274b84e

2. Industrial Examples:

- Real-Time Manufacturing Control

- Energy Monitoring and Optimization
- Predictive Maintenance
- Logistics and Material Handling
- Real-Time Quality Inspection

3. Attack Vectors:

3.1 Port Stealing and MAC Spoofing Attack

PROFINET IO devices use MAC addresses to send and receive Real-Time (RT) data directly. An attacker can force the network switch to update its CAM (forwarding) table by spoofing the MAC address of a PLC or IO device and sending repeated fake frames. This will send all future communication to the attacker's device instead of the real one. If the attacker is successful,Intercept PROFINET Real-Time packets , Block PLC-IO communication , Pretend to be the PLC and send fake commands.

“The MITM-based Profinet IO-Device Emulation” (Michel Baud & Max Felser, 2006) study showed this attack in action. The attacker was able to redirect cyclic RT data and take full control of the IO device.

3.2 DCP Reconfiguration Attack (Device Name/IP Manipulation)

DCP is used by PROFINET devices to get device names and IP settings when they start up. An attacker can send fake DCP Set Name or DCP Set IP commands to change the name or address of a field device or sensor because DCP doesn't have authentication.
This results in: The device is no longer connected to the PLC; The Application Relation (AR) cannot be set up; and the device can be reconfigured by an attacker.

The Paper **"No Need to Marry to Change Your Name" (Mehner & König, 2019)** experimentally confirmed this attack. Renaming an IO device ("mallory") permanently broke PLC communication, and the operator couldn't fix it even with engineering software.

3.3 Replay Attack on PROFINET IO Real-Time Frames

In replay attacks, the attacker first passively captures valid PROFINET cyclic RT packets (like motor start/stop) and then sends them to the IO device at a different time. The device accepts and runs the old command because the protocol doesn't protect against replay attacks.

The “**Replay Attacks in Industrial Automation Networks** ” **(Steffen Pfrang and David Meier, 2017)** paper showed how this attack works. It showed how a motor could be started or

stopped from a distance using RT frames that had already been captured, without having to know how to program or access PLC logic.

3.4 RT Command Hijacking

In contrast to replay attacks, which involve reusing old packets, RT command hijacking involves creating new malicious control commands that are similar to valid PLC instructions. The attacker reverse engineers the RT packets to understand the content and format of the payload structure, identifies critical bytes in the payload (e.g., actuator ON/OFF), spoofs the MAC address of the PLC, and injects false RT commands.

This attack was demonstrated in the **RT Command Hijack experiment**, where researchers reverse engineered RT command structure and successfully controlled actuators (motors) using fabricated Ethernet frames.

3.5 Denial-of-Service (DoS) Attack in PROFINET Networks

In PROFINET, continuous cyclic communication between PLC and IO-devices is essential. In the demonstrated experiment (Mehner & König, 2019), the attacker first hijacked communication using MAC spoofing (Port Stealing), and then sent repeated forged DCP Set Name/Set IP messages. PROFINET does not authenticate DCP messages, and as a result, the device renaming or misconfiguration causes the PLC to lose communication with the device permanently. This works as a sustained Denial-of-Service, whereby normal engineering tools or even rebooting the PLC cannot restore the connection.

4. Real life Attacks:

4.1 PROFINET IO Stack Denial-of-Service (CVE-2019-13946)

A critical security flaw, CVE-2019-13946, was discovered in the PROFINET IO (PNIO) communication stack used in Siemens PLCs, drives, and industrial controllers. The vulnerability allowed attackers to send multiple PROFINET diagnostic requests that caused improper memory handling inside the device. Because the PNIO stack lacked resource management controls, repeated requests led to memory exhaustion, resulting in a denial-of-service condition.

During exploitation, the affected device stopped responding to PROFINET cyclic communication, causing loss of connection with IO devices. In several cases, the affected device required a manual restart to restore normal operation. This incident demonstrated that PROFINET-based industrial devices could be remotely disabled using crafted network traffic without authentication, threatening availability and reliability of industrial processes.

Source : https://nvd.nist.gov/vuln/detail/cve-2019-13946

4.2 Hilscher PROFINET IO Device Memory Corruption (CVE-2021-20986)

In 2021, CVE-2021-20986 was discovered in the Hilscher PROFINET IO Device Stack, which is widely used in industrial field devices from Pepperl+Fuchs, ifm, Moxa, Hilscher, and HMS. The vulnerability allowed a remote attacker to send specially crafted PROFINET frames that caused memory corruption in the IO device. This led to failures in cyclic communication, and in many cases, the device completely disappeared from the network and became undetectable by engineering tools. Unlike temporary communication failures, this attack often resulted in **persistent denial-of-service**, as the device did not recover after a system reboot and sometimes required firmware reinstallation. This incident demonstrated how a single malformed PROFINET packet could interrupt industrial process control, isolate devices from the automation network, and cause long-term disruption to operational technology systems.

4.3 Siemens EN100 Module Information Disclosure (CVE-2016-4785)

The Siemens EN100 Ethernet communication module, used in SIPROTEC protection relays and PROFINET-enabled electrical substation equipment, was found to contain a vulnerability allowing unauthorized access to stored configuration and memory data. Identified as CVE-2016-4785, this vulnerability enabled attackers to retrieve sensitive information such as device names, IP addresses, MAC addresses, firmware versions, PROFINET settings, and engineering parameters through the device's web interface without authentication.

Although this vulnerability did not directly interfere with PROFINET cyclic communication, it allowed detailed mapping and profiling of the industrial network. This information could later be used to perform targeted attacks such as device spoofing, DCP reconfiguration, name impersonation, or RT packet manipulation. The incident highlighted the importance of confidentiality and demonstrated that reconnaissance-based attacks can assist in preparing advanced PROFINET attack strategies.

5. Resources:

1.https://us.profinet.com/wp-content/uploads/2020/10/How-PROFINET-Works-Complete.pdf

2. Mehner, S. and König, H., 2019, June. No need to marry to change your name! attacking profinet io automation networks using dcp. Source

3.Pfrang, S. and Meier, D., 2017, February. On the Detection of Replay Attacks in Industrial Automation Networks Operated with Profinet IO. Source

4.Baud, M. and Felser, M., 2006, September. Profinet io-device emulator based on the man-in-the-middle attack Source

5. Blumbergs, B., 2019. Remote Exploit Development for Cyber Red Team Computer Network Operations Targeting Industrial Control Systems.Source

6. Makrakis, G.M., Kolias, C., Kambourakis, G., Rieger, C. and Benjamin, J., 2021. Vulnerabilities and attacks against industrial control systems and critical infrastructures.Source

Modbus

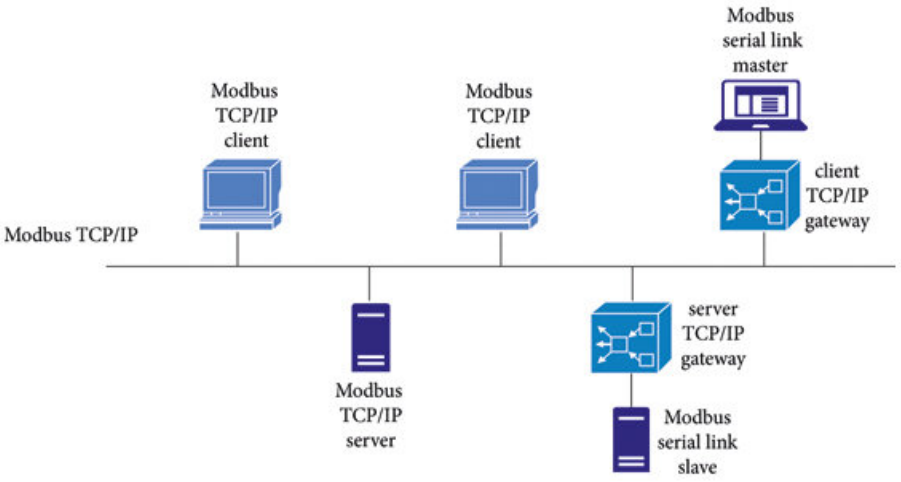
---

1. Introduction to Modbus TCP

Modbus TCP is a widely used industrial communication protocol that enables data exchange between control devices, such as PLCs (Programmable Logic Controllers), RTUs (Remote Terminal Units), sensors, and SCADA systems, over Ethernet networks. It is part of the broader Modbus family, originally developed by **Modicon (Schneider Electric)** in 1979, and later extended to operate on modern IP-based networks.

In Modbus TCP, the traditional **Modbus messaging structure**—which defines how data is read or written to device registers—is encapsulated within **TCP/IP packets** and transmitted over standard Ethernet. This allows Modbus to integrate easily with existing IT and industrial networks without requiring specialized hardware or serial cabling (like RS-485). It uses **port 502** by default for communication. Unlike the older **Modbus RTU**, which operates on serial lines, Modbus TCP supports faster data transfer, greater scalability, and easier network configuration through IP addressing. It follows a **client–server architecture**, where:

- The **client** initiates requests.
- The **server** responds with the requested data or executes commands.



Source: [Modbus TCP protocol network topology. | Download Scientific Diagram]

Because it operates on Ethernet, Modbus TCP can coexist with other standard network services (like HTTP or SNMP) and be routed through typical IT infrastructure such as **switches**, **routers**, and **firewalls**. This flexibility has made it one of the most popular and enduring protocols for Industrial Control Systems (ICS) and Operational Technology (OT) networks.

Modbus TCP was designed for **reliability and simplicity**, not for cybersecurity

2. Industrial Examples

2.1 Industrial Manufacturing Lines

Factories often connect multiple PLCs, motor drives, sensors, and robots using Modbus TCP. PLCs share production data—such as conveyor speed, robot position, motor current, or fault codes—over Ethernet. HMI stations also use Modbus TCP to monitor and control machine states from operator panels.

2.2 Oil & Gas Pipeline Control

Pipeline stations use RTUs with Modbus TCP to monitor pressure, flow rate, and valve status. The SCADA system communicates over long-range Ethernet, fiber, or radio networks using Modbus TCP. Setpoints like pump speed or valve positions are written through Modbus registers.

2.3 HVAC and Building Automation

Building Management Systems manage chillers, AHUs, boilers, meters, and VAV units using Modbus TCP. Each HVAC device acts as a Modbus server exposing registers for temperature, airflow, damper position, energy usage, etc. A central BMS server periodically polls these devices to adjust indoor climate and optimize energy consumption.

3. Attack Vectors

3.1 Sniffing (Eavesdropping)

Because Modbus TCP sends everything in plaintext, attackers capturing network traffic can read sensor values, control commands, and register maps. This helps them plan deeper attacks.

3.2 Spoofing / Impersonation

Modbus TCP has no authentication, so any device on the network can impersonate a legitimate client (HMI/SCADA) and issue commands to PLCs.

3.3 Replay Attacks

Captured Modbus packets can be resent later. Since there are no timestamps or signatures, devices interpret replayed commands as real ones.

3.4 Unauthorized Write Commands

Write function codes (0x05, 0x06, 0x0F, 0x10) can change setpoints, outputs, and critical process values. An attacker could stop pumps, open valves, or modify temperature/pressure limits.

3.5 Broadcast / Trust-of-Network Misuse

Legacy Modbus assumes that everyone on the network is trusted. On serial links, broadcasts can change multiple devices at once. On TCP networks, gateways can unintentionally forward harmful broadcast-like traffic to many devices.

3.6 Device Fingerprinting via Function Codes

Diagnostic and device-identification functions reveal device type, firmware version, and register layout — useful information for attackers.

3.7 Denial of Service (DoS)

Flooding port 502 or sending malformed packets can freeze or crash PLCs or gateways, disrupting operations.

3.8 IT → OT Pivot Attacks

Compromising an office computer, VPN account, or engineering laptop can give attackers a path into the OT network where Modbus devices reside.

4. Real-Life Examples

4.1 FrostyGoop – Lviv Heating Attack (2024)

A newly discovered malware, *FrostyGoop*, targeted a municipal heating operator in Lviv, Ukraine. Attackers sent malicious Modbus TCP write commands to ENCO industrial controllers, disrupting hot-water and heating services for ~600 apartment buildings for over 48 hours during winter. This is one of the first large-scale attacks where Modbus TCP commands directly caused physical service disruption. hub.dragos.com+4Security-Insider+4BornCity+4

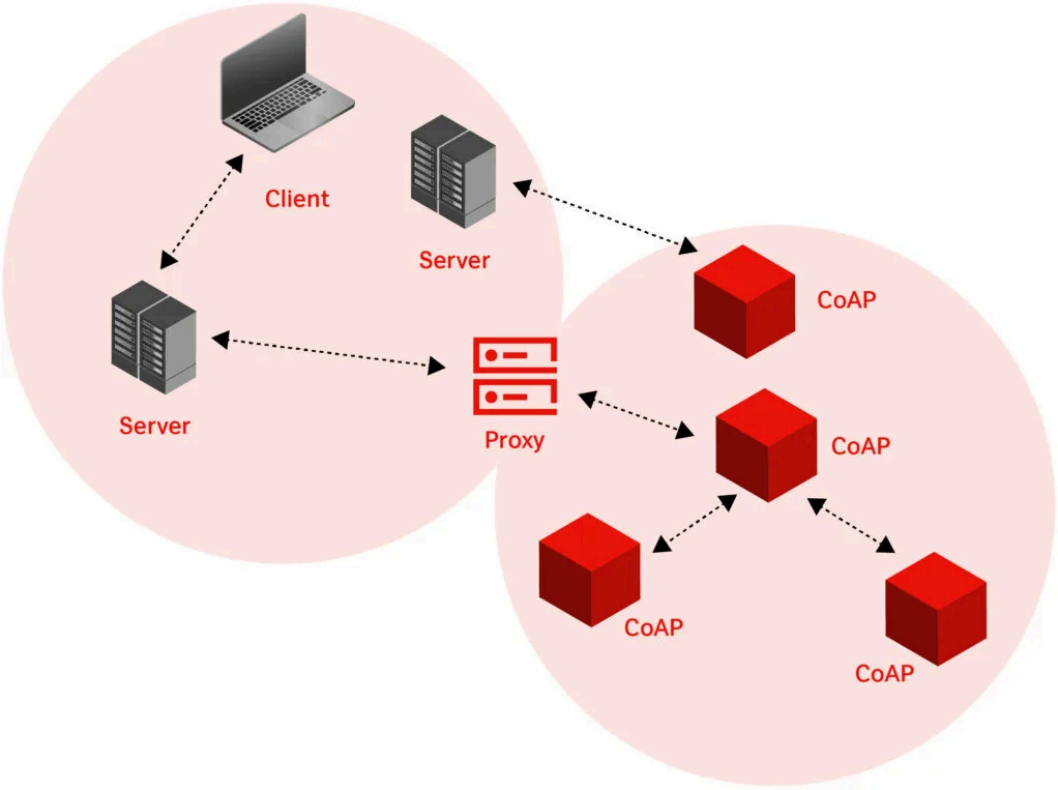
4.2 Simulation and Analysis of Cyber-Attack on Modbus Protocol for Smart Grids (Banik et al.)

This paper simulates several cyber-attacks on Modbus TCP within a smart grid virtual environment. It demonstrates how replay attacks, false data injection, unauthorized writes, and DoS can disrupt grid monitoring and control. The study highlights how vulnerable Modbus-based smart grid components are to manipulation and shows how these attacks affect voltage stability and system performance. https://www.preprints.org/manuscript/202309.0984

5. Resources:

- Analysis of SCADA Security using Penetration Testing: A Case Study on Modbus TCP Protocol (2018) – looks at penetration testing of SCADA systems using Modbus TCP.Source
- Performance Evaluation of Modbus TCP in Normal Operation and Under a Distributed Denial of Service Attack (2020) – benchmarks Modbus TCP devices under DDoS conditions.Source
- Assessment of Hidden Channel Attacks: Targeting Modbus/TCP (2020) – discusses covert channels using Modbus TCP.Source
- Enhanced Modbus/TCP Security Protocol: Authentication and Authorization Functions (2022) – proposes enhancements for Modbus TCP security.Source
- Design and Implementation of a Lightweight Security-Enhanced Scheme for Modbus TCP Protocol. Source

CoAP: Constrained Application Protocol



Source: https://www.a1.digital/knowledge-hub/what-is-the-constrained-application-protocol-coap/

1. Introduction:

- CoAP is a specialized web transfer protocol for low-power and resource-constrained devices in the Internet of Things (IoT).
- It follows a client–server model similar to HTTP but is optimized for devices with very limited processing power, memory, and energy, such as sensors and actuators.
- CoAP usually runs over UDP (User Datagram Protocol), which reduces connection overhead and latency compared to TCP. [4]

Message Structure and Operation:

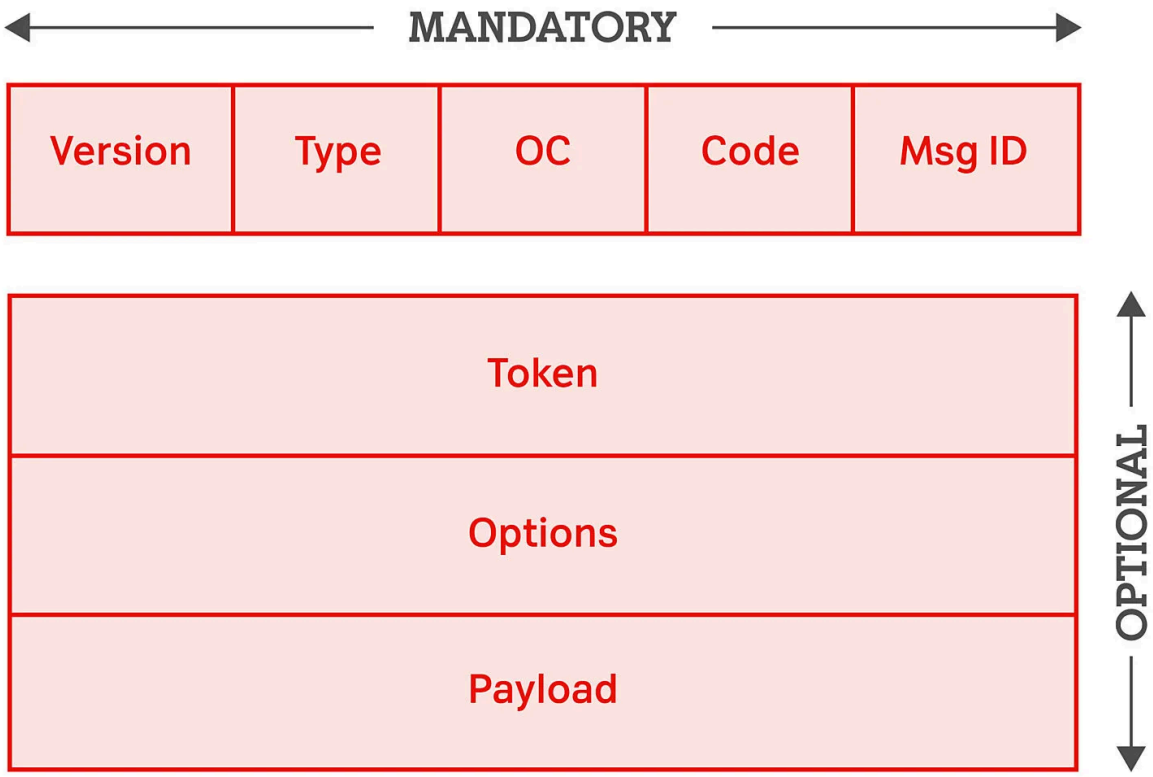
CoAP messages are binary encoded and consist of:

- A fixed 4-byte (32-bit) header
- An optional token field (0–8 bytes)
- Optional options (metadata)

- An optional payload area for application data

Each message includes:

- A **Message ID** (used for matching acknowledgments and detecting duplicates)
- A **Token** (used to link requests and responses at the application level)



Source: https://www.a1.digital/knowledge-hub/what-is-the-constrained-application-protocol-coap/

Message Types:

1. **Confirmable (CON):** Requires acknowledgment (reliable).
2. **Non-confirmable (NON):** No acknowledgment needed (unreliable).
3. **Acknowledgment (ACK):** Confirms a received CON message.
4. **Reset (RST):** Indicates an error or communication issue.

This system provides efficiency and optional reliability despite using UDP while still staying lightweight. [4]

Strengths of CoAP:

- **Lightweight & Low Overhead:** Minimal memory and processing needs.
- **Energy Efficient:** Ideal for **battery-powered devices**.
- **RESTful Design:** Simple integration with existing IoT and web APIs.
- **Multicast Support:** Allows one device to address multiple nodes.

These strengths make CoAP a preferred protocol for **constrained IoT networks**, where bandwidth and energy consumption are critical factors. [4]

Weaknesses and Limitations:

- **Unreliable Transport:** UDP lacks built-in error correction, leading to packet loss especially in noisy wireless environments.
- **Extra Reliability Layer Needed:** Acknowledgment mechanisms must be manually handled.
- **Compatibility and Integration Overhead:** Not directly compatible with HTTP; requires proxies for integration.
- **Security Overhead on constrained nodes:** DTLS (at transport layer) and OSCORE (application layer) introduce computational load unsuitable for low-power nodes. [4]

2. Industrial Example Use Case: Smart City Street Lighting & Metering:

In a smart city, thousands of streetlights and electricity meters are deployed as constrained IoT devices. These devices use a low-power network such as 6LoWPAN over UDP and expose their functions via CoAP:

- Each lamp or meter is modeled as a REST resource, e.g. coap://lamp123/state.
- The city control system can remotely **switch lights on/off**, **dim them**, and **read energy consumption** using CoAP requests.
- A local **gateway** collects CoAP messages from the field devices and translates them to **HTTP or MQTT** for cloud platforms and dashboards.

This architecture keeps communication **lightweight and energy-efficient** in the street network, while still allowing **centralized monitoring and control** in the cloud.

3. Attack Vectors:

Threat	Example	CoAP Mechanism
Eavesdropping / Data Interception	Sensor sending a reading over CoAP without DTLS- an attacker in the same network sniffs the UDP datagrams and read the value [1]	DTLS/ OSCORE for encryption and integrity helps prevent this.
Replay Attack	When a device issues a CoAP POST command like "door-unlock" to a smart lock. Attacker captures the datagram and later re-sends it which causes the door to unlock again maybe at inappropriate time [1]	DTLS sequence numbers/ OSCORE nonces helps guard against reuse of old messages.
Message Tampering / Spoofing / Injection	A CoAP client receives a "temperature = 100 °C" reading but the attacker modified the payload in transit (or injected a fake CoAP packet) to make the controller think there's an overheating fault. [1]	Integrity protection in DTLS/ OSCORE are key and also Authentication like pre-shared keys and certificates helps prevent this.
DDoS Amplification Attack	An attacker sends small CoAP requests using a spoofed source IP (victim's IP) to multiple CoAP servers. Because CoAP supports responses, the servers send much larger responses to the *victim's* IP, overwhelming it. [1]	This attack exploits the protocol's UDP basis and lack of built-in authentication; mitigation includes DTLS, careful server config, and echo/token options.

4. Real-Life Examples

❖ Smart homes / buildings

In smart homes and buildings, low-power devices (sensors and actuators) communicate over **IEEE 802.15.4 / 6LoWPAN** and expose their functions as **CoAP resources**:

- **Sensors** (temperature, motion, door, smoke) → CoAP GET on resources like /temp, /motion, /door/status.
- **Actuators** (lights, HVAC, alarms, locks) → CoAP PUT on resources like /light/level, /hvac/setpoint, /alarm/arm.
- A **border router / gateway** terminates 6LoWPAN+CoAP and translates to **HTTP/MQTT** for the building controller or cloud, often using **DTLS-secured CoAP** on the device side to protect commands and telemetry. [2]

❖ Industrial and Robotics

In industrial and robotic systems, CoAP is used for **remote control and monitoring** of robots and factory devices over constrained networks:

- A 2025 mobile-robot architecture by **Sarkar et al.** uses **CoAP over UDP/Wi-Fi** between a NodeMCU (ESP8266) on the robot and an IoT cloud server (ThingSpeak). The robot's sensors and actuators (motors, ultrasonic sensors, camera) are exposed as **CoAP resources**, and Android/web clients send **CoAP GET/PUT/POST** requests for telemetry and motion control.

- In industrial automation, CoAP is used between **sensors, controllers, and actuators** to implement low-latency monitoring and control of process variables (temperature, vibration, level, etc.) over low-power IP networks. [3]

5. Resources

[1] Attacks on the Constrained Application Protocol (CoAP) (2025) – IETF Internet-Draft that systematically analyzes threats and attack patterns against CoAP deployments. Source

[2] Secure and Energy-Efficient Smart Building Architecture with Emerging Technology IoT (2021) – proposes a smart building design using CoAP + DTLS between sensors/actuators and shows improved energy efficiency. Source

[3] Development of CoAP Protocol for Communication in Mobile Robotic Systems Using IoT Technique (2025) – uses CoAP for remote monitoring and control of a mobile robot and compares its performance to other protocols. Source

[4] What is the Constrained Application Protocol (CoAP)? (A1 Digital) – introductory article explaining CoAP basics, message structure, strengths, and weaknesses. Source