Accepted for presentation in: IEEE International Conference on Factory Communication Systems (WFCS), Offenburg, Germany, April 2026.

Teleoperating Mobile Robots via VDA 5050 – A First Middleware Evaluation within Open Industrial Networks

Niklas A. Wagner*[1], Lars Toenning*[2], Dennis Lünsch[2], Jana Jost[2,3], Christian Wietfeld[1], and Peter Detzner[2,4]

***Abstract*—Teleoperation is a practical fallback for mobile robots in intralogistics when rare but safety- and throughput-critical edge cases exceed autonomous capabilities. While Fleet Management Systems (FMSs) increasingly rely on vendor-neutral interfaces such as VDA 5050 to coordinate heterogeneous robots, teleoperation is not natively represented in the standard and is typically implemented as an ad hoc, robot-specific add-on. This paper closes this gap by proposing a VDA 5050 teleoperation extension that enables human-in-the-loop control as part of the same interface used for autonomous task and route coordination. To facilitate reproducible evaluation, we further contribute a configurable VDA 5050 traffic generator that reproduces realistic message patterns for middleware-based FMS communication. Employing the Open Radio Access Network (Open RAN) concept, cellular 5G and future 6G networks can be flexibly deployed in industrial environments to provide reliable, low-latency wireless connectivity for mobile robot teleoperation. Using an end-to-end stack consisting of VDA 5050 messaging, middleware transport, and Open RAN wireless link, we evaluate latency and communication overhead to characterize the performance implications of standardized teleoperation integration.**

I. Introduction

Smart factories significantly increase flexibility, intelligence, and efficiency by operating as distributed cyber-physical networks capable of autonomous decision-making and dynamic coordination. [1, 2]. In this context, intralogistics emerges as a key enabler of flexible material flow, where Fleet Management Systems (FMSs) supervise and optimize fleets of mobile robots and thus strongly influence throughput and service quality. Acting as cyber-physical coordination layers, FMSs continuously collect state, localization, and performance data from distributed agents and apply operations research and AI-based methods for tasks such as dynamic dispatching, routing, traffic management, and maintenance planning [3, 4].

At the same time, smart factories are becoming increasingly complex, with growing numbers of heterogeneous mobile robots for which isolated, robot-centric autonomy is often insufficient to ensure robust and scalable operation [5]. Therefore, FMSs increasingly rely on standardized, vendor-neutral interfaces such as VDA 5050 to enable mixed-fleet operation across vehicle types by defining common message structures for orders, navigation, and status exchange [6]. By abstracting vehicle-specific control from fleet-level decision making, VDA 5050 improves interoperability, reduces vendor lock-in, and provides a scalable foundation for advanced multi-agent coordination and optimization in modern cyber-physical production systems and intralogistics [7].

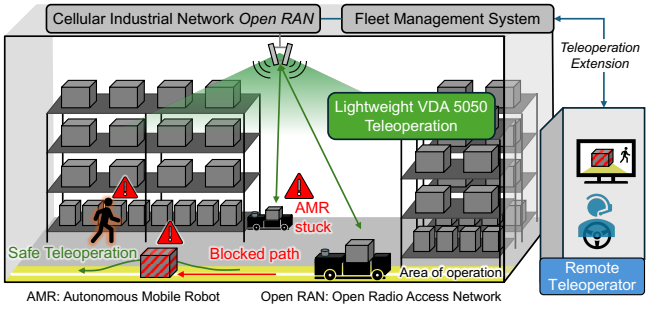


Fig. 1: Human-in-the-loop teleoperation of mobile robots via an FMS and VDA 5050 with video feedback and low-latency remote control in wireless open industrial networks.

Despite these advances, the autonomy of (mobile) robots often remains insufficient for complex, real-world operations. Teleoperation complements autonomy by providing a practical fallback for rare but safety- and throughput-critical edge cases such as blockages or navigation failures, where human intervention is required to maintain operational flow [8]. This requires teleoperation support within FMSs, particularly for safe operation in human-shared environments that extend beyond dedicated zones, as illustrated in Fig. 1.

Hence, robust low-latency wireless connectivity between the FMS and the mobile robots [9] is required. While Wi-Fi is common in industrial deployments, it often faces scalability issues in congested environments, making 5G and future 6G networks preferable due to their robustness [10]. Cellular industrial networks can be flexibly deployed using the vendor-neutral Open Radio Access Network (Open RAN) concept, enabling integration of use-case specific 6G innovations [11].

Unlike standard FMS traffic, teleoperation introduces asymmetric, high-demand communication flows: continuous video uplink for situational awareness and low-latency control downlink, forming a closed human-in-the-loop control loop [12]. As a result, wireless connectivity becomes a performance-critical dependency [6, 13] as latency, jitter, reliability, and scalability directly affect the operator experience and the ability of the FMS to maintain stable fleet-level coordination in dense intralogistics environments. To fully exploit the flexibility of standardized, vendor-neutral fleets for handling rare, critical edge cases, teleoperation needs to be tightly integrated with VDA 5050 for human intervention within the same unified interface. This integration motivates end-to-end evaluation of

* Contributed equally to this paper.
[1]N. A. Wagner (niklas.wagner@tu-dortmund.de) and C. Wietfeld are with the Chair of Communication Networks, TU Dortmund University, Germany
[2]D. Lünsch and J. Jost are and L. Tönning and P. Detzner were with the Fraunhofer Institute for Material Flow and Logistics, Germany
[3]J. Jost is with the Institute of Logistics Engineering, Hamburg University of Technology (TU Hamburg), Germany
[4]P. Detzner is with the Westphalian University of Applied Science, Germany

the complete stack (protocol, middleware, and radio link), rather than assessing communication technologies in isolation.

Consequently, the main contributions of this work are:

- **Extension of VDA 5050 to support teleoperation** within FMSs, enabling manual control while preserving interoperability across heterogeneous mobile robot types.
- **Design of a configurable VDA 5050 traffic generator** to reproduce realistic message patterns for evaluation of middleware-based FMS communication.
- **Evaluation of an end-to-end teleoperation stack**, quantifying the overhead and latency characteristics via middleware transport and Open RAN cellular network.

The remainder of this paper is structured as follows. Section II reviews related work on teleoperation and provides relevant background on the VDA 5050 standard. Section III details the system design and presents our VDA 5050 extensions for teleoperation support. Section IV evaluates the approach in a simulation-based study, and Section V concludes the paper and outlines future research directions.

II. Related Work and Background

With rapid advances in Artificial Intelligence (AI), the global trend in robotics points toward increasing autonomy in real world deployments. Against this backdrop, teleoperation may seem counterintuitive, but remains an important bridge to robust autonomy, as it enables safe human support in borderline cases and generates rich situational data for continuous improvement. Accordingly, this section reviews related work on teleoperation via cellular networks with a focus on end-to-end wireless communications, and on standardized interfaces for heterogeneous fleets with a focus on interoperability layers like VDA 5050 representing human-in-the-loop operation.

A. Teleoperation within Open Industrial Networks

Multi-modal wireless teleoperation is a key use case driving research and standardization efforts for future 6G networks [14]. Teleoperation addresses both safety-critical scenarios where autonomous systems require human oversight, and intentional human control for complex tasks. It has found applications across multiple domains including autonomous vehicles [15], excavation tasks [16], and remote medical examinations [17] or telesurgery. Here, control loops can exist directly between the operator and Human Machine Interface (HMI), or within the robot if some functionalities are operating semi-automatically [12].

Various experimental studies have demonstrated mobile-robot teleoperation over mobile networks. For instance, one prototype setup enables remote and on-robot controllers via 5G at 28 GHz, reporting end-to-end latencies of low milliseconds [18], emphasizing that measured performance depends on the full end-to-end path. Teleoperation of Automated Guided Vehicles (AGVs) has also been demonstrated over 5G Non-Standalone (NSA) public mobile networks [13]. In [19], the authors propose joint-level automated teleoperation of humanoid robots via edge computing for collaborative tasks including advanced Open RAN resource scheduling. Beyond industrial robotics, teleoperation has become increasingly relevant for autonomous vehicles, where remote operators provide fallback control in challenging scenarios [15]. While such evaluations are typically conducted in public networks rather than dedicated private or industrial campus networks, results can nonetheless inform robustness considerations for industrial teleoperation use cases as proactive bitrate adaption [15]. In intralogistics environments specifically, the auhors of [20] analyzed teleoperation of robotic forklifts over Ethernet, and public 5G NSA networks, finding that approximately 60% of glass-to-glass latency originated from sensor capture delays, highlighting the importance of optimizing the entire end-to-end chain, beyond pure network latency.

B. Standardized Interfaces for Heterogeneous Fleets

Ensuring interoperability across heterogeneous fleets, reducing vendor lock-in, and providing a scalable integration foundation are core requirements in cyber-physical intralogistics systems [21]; as a result, VDA 5050 emerged as a common interface standard. VDA 5050 defines a standardized, vendor-neutral communication interface between a central master control (e.g., an FMS) and AGVs/Autonomous Mobile Robots (AMRs) fleets to enable mixed-vendor operation. It specifies the structure and semantics of the exchanged information (e.g., mission/orders, navigation-related data, and vehicle status) and is accompanied by machine-readable artifacts (e.g., schemas) to support consistent implementation across vendors.

All VDA 5050 communication is realized via an MQTT broker in a publish–subscribe manner: the master control publishes orders (topic `order`) and immediate commands (topic `instantActions`), while the mobile robot publishes execution and health information (topic `state`) and, optionally, higher-rate pose data for monitoring (topic `visualization`); a dedicated `connection` topic supports broker-level connection checks. All messages are exchanged as JSON payloads (validated against the provided schemas) on well-defined topics, enabling vendor-neutral interoperability across heterogeneous fleets.

The most common message types of the VDA 5050 are listed in Table I. The *Connection* and *Factsheet* messages

TABLE I: Frequency, direction, minimum, and estimated message sizes of the VDA 5050 messages of the FMS, including the VDA 5050 header. Up- (↑) and downlink (↓) direction is specified from the perspective of the mobile robot.

Message Type	Frequency	Direction	Min. size	Estimated Size
Connection	once	↑	130 B	≈ 144 B
Factsheet	once	↑	593 B	≈ 1551 B
Order	aperiodic	↓	156 B	≈ 206 B + 154 B/node + 434 B/edge
Instant Action	aperiodic	↓	116 B	≈ 293 B
Visualization	aperiodic, min. every 30 s	↑	185 B	≈ 201 B
State	periodically 10 Hz	↑	343 B	≈ 522 B + 75 B/node + 91 B/edge

are typically transmitted only once and therefore contribute negligibly to the overall network load. *Order* and *Instant Action* messages are downlink traffic towards the mobile robot, and their frequency depends strongly on the use case. For example, an FMS demo may operate without any *Instant Actions*, whereas complex transport processes often require additional instant actions, e.g., to integrate safety-related devices or to actuate load-handling units. For a single transport order, a single *Order* message is sufficient if the full path is already free; otherwise, the path must be sent piecewise as segments become available, resulting in multiple *Order* messages. The *Visualization* message is sent periodically at a configurable rate (here: 10 Hz) to ensure smooth monitoring. The *State* message is event-driven (aperiodic) on state changes, including progress along nodes and edges, but is sent at least every 30 s.

III. System Design of Teleoperation in FMSs

In this section, we present a VDA 5050–compliant teleoperation architecture that augments standard fleet-to-mobile-robot communication with wireless teleoperation channel. The system, as shown in Fig. 2, augments the conventional VDA 5050 interaction between FMS and AGV/AMR by introducing a teleoperation service that *i*) injects teleoperator control commands aligned with the VDA 5050 to enable teleoperation, *ii*) and streams video data from the robot to the teleoperator as well as additional sensor data.

A. Interoperable Teleoperation Architecture

Our architecture is organized in three layers that together enable interoperable teleoperation of heterogeneous mobile robots. On the *Shopfloor Layer*, the physical mobile robot communicates wirelessly; the standard VDA 5050 interface is extended because its original scope is insufficient for teleoperation, while preserving interoperability even for robots with proprietary teleoperation interfaces (sensor publishing / command receiving). On the *Application Layer*, a dedicated *Teleoperation Extension* is introduced as the single external entry point for teleoperation functions. When teleoperation is activated, the VDA 5050 *master control* ownership is transferred from the *Physical Mobile Robot* to the *Teleoperation Extension*, which then directly communicates with the robot and interacts with the FMS's routing service on the robot's behalf (e.g., route requests and topology allocations for collision avoidance). For situational awareness, video (and optionally additional sensor streams) is provided via a streaming server acting as an intermediate to avoid tight coupling between robot and HMI and to reduce load when multiple viewers consume the stream. To handle interrupted transport tasks caused by teleoperation, the FMS is extended with policies for robot-side behavior (e.g., wait/park/accept-only-specific-task) and task-side recovery options (skip, restart, or continue by reassigning remaining orders to the same robot), enabling completion of tasks even when a loaded robot must be teleoperated mid-mission. On the *User/Business layer*, a web-based *Teleoperation Interface* connects to the *Teleoperation Extension* to start/stop teleoperation, manage FMS-routing-service interactions (e.g., allocate topology, delete routes), and provide an HMI for the video stream and manual control, while remaining extensible to different robot types and protocols.

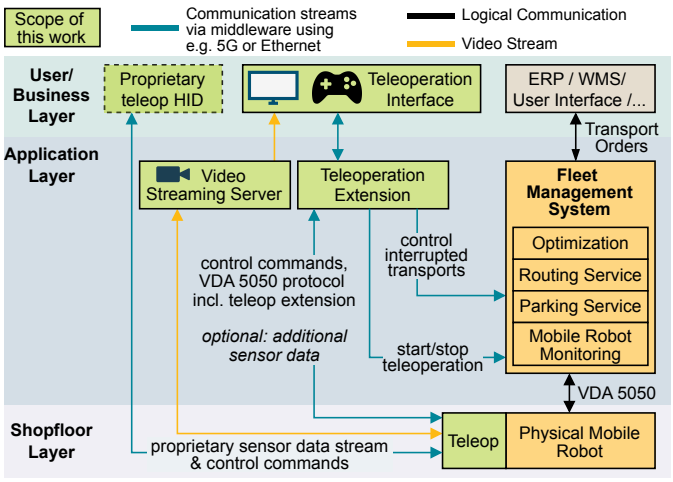


Fig. 2: Extended FMS architecture for teleoperation via open industrial networks (logical interactions only; no physical data flow). Green components are within the scope of this work.

B. Teleoperation for VDA 5050-based Mobile Robots

We propose a backward-compatible extension to VDA 5050 by introducing teleoperation-specific instant actions and status messages under a dedicated namespace. Robots that do not support teleoperation will be excluded from teleoperation by the FMS, as they do not provide the teleoperation instant actions. As shown in Fig. 3, the proposed *libTeleop* encapsulates robot-independent teleoperation functionality, while robot-specific aspects remain within the respective robot implementations. To account for heterogeneity across robot types, we introduce *teleoperation capabilities* analogous to the VDA 5050 factsheet (cf. Listing 1). These capabilities (e.g., control protocol and sensor configurations) are specified per robot in the *libVDA5050++* configuration and can be queried at runtime via a VDA 5050 instant action

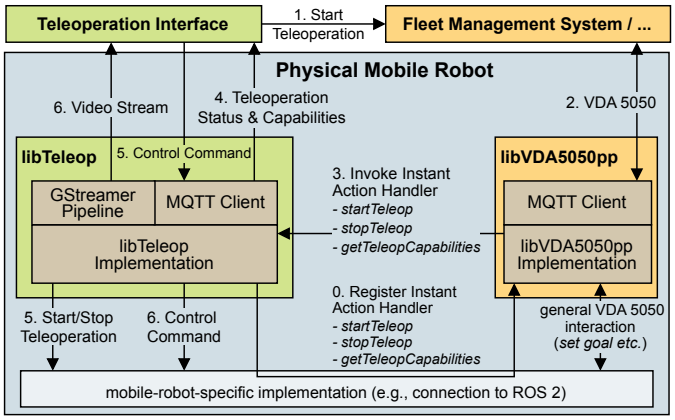


Fig. 3: *libTeleop* integration with libVDA5050++ in the mobile robot stack; control commands are routed from the Teleoperation Interface via the FMS to the robot.

```
[teleoperation.capabilities]
json = '''{
"control_protocol": "VDA5050_MQTT",
"sensors": [{
  "id": 1,
  "name": "cam",
  "type": "camera",
  "url": "http://<IP>:8889/teleop",
  "gstreamer_pipeline": "videotestsrc is-live=
      true ! video/x-raw,format=NV12,field=none,
      framerate=60/1,width=1280,height=800 !
      nvh264enc zerolatency=true bitrate=2000 !
      queue leaky=downstream max-size-time
      =20000000 ! rtspclientsink rtp-blocksize
      =1300 location=rtsp://10.53.1.4:8554/
      teleop"
}]}'''
```

Listing 1: Example *libTeleop* configuration as part of the libVDA5050++ configuration, using the integrated control protocol and providing a test source video stream.

of type `teleopCapabilitiesRequest`. Upon reception, *libTeleop* automatically registers and handles the instant action and publishes the capabilities using the standard VDA 5050 topic structure.

C. Teleoperation Extension

The *Teleoperation Extension* bridges the *Teleoperation Interface* with the FMS components. It exposes a WebSocket interface towards the web-based *Teleoperation Interface* and uses MQTT for intra-system communication. WebSockets were selected for the UI link because they are natively supported by contemporary browsers and provide an efficient bidirectional 1:1 channel without the topic-management overhead inherent to publish–subscribe communication. Furthermore, the *Teleoperation Extension* aggregates system state, including the status of mobile robot, transport orders, and route reservations/allocations, and disseminates this information to the teleoperation interface. In the opposite direction, the *Teleoperation Extension* processes commands originating from the teleoperation interface, covering *i*) activation and deactivation of teleoperation, *ii*) route-management operations (e.g., allocation and deallocation of topology resources), *iii*) and handling of interrupted or aborted transport orders. For the basic control protocol considered in this work, incoming manual-control commands are validated and forwarded to the physical robot. To increase operational robustness, the extension is designed to minimize persistent internal state, enabling restarts during runtime. This is achieved through MQTT retained messages (providing the most recent state to late-joining subscribers) and request–reply interactions to explicitly retrieve required system context.

D. Video Streaming

Video streaming is realized using MediaMTX, an open-source streaming server. On the physical robot (NVIDIA Jetson with an Intel RealSense D455 camera), a GStreamer pipeline is deployed. The pipeline captures video frames via `v4l2src`, performs memory-format conversion using `nvvidconv`, encodes the stream with the hardware-accelerated `nvv4l2h264enc` H.264 encoder, and publishes the stream to MediaMTX via Real-Time Streaming Protocol (RTSP). The media payload is carried using Real-Time Transport Protocol (RTP) over UDP. On the client side, the *Teleoperation Interface* consumes the video stream via WebRTC protocol providing the lowest evaluated latency. Since teleoperation is evaluated in simulation, an equivalent pipeline is instantiated using a test video source that emulates the RealSense D455 video characteristics.

TABLE II: Frequency, direction, minimum, and estimated message sizes of teleoperation messages of the FMS, including the VDA 5050 header. Up- (↑) and downlink (↓) direction is specified from the perspective of the mobile robot.

Message Type	Frequency	Direction	Min. size	Estimated Size
Factsheet	once	↑	36 B	≈ 405 B
Status	aperiodic	↑	22 B	≈ 22 B
Control Cmd	periodically 20 Hz	↓	4 B	≈ 14 B

E. Message Frequency and Time Criticality

Table II summarizes the typical message frequencies exchanged between the FMS / *Teleoperation Extension* and the *Physical Mobile Robot* over a middleware such as MQTT, separated into base VDA 5050 messages and teleoperation-extension messages. Similar to the VDA 5050 factsheet, the *Teleoperation Factsheet* is only sent once. The *Teleoperation Control* command to the mobile robot is currently sent at 20 Hz. This frequency has to be adjusted according to the needs of the mobile robot control.

Although VDA 5050 communication via the middleware is not part of a closed low-level control loop, low end-to-end latency remains important. We therefore rank the base VDA 5050 and teleoperation-extension message flows by latency criticality: *Teleoperation Control* commands are the most time-sensitive, as they directly affect robot motion and thus teleoperation performance [22]. *Visualization* messages (e.g., robot pose: position and orientation) support the operator's situational awareness and should be delivered promptly alongside other sensor streams. *Order* messages provide piecewise path clearances; timely delivery helps avoid stop-and-go behavior, since even small delays (e.g., around 50 ms) can trigger braking and reduce traffic-flow efficiency and system throughput. Similarly, *State* messages should reach the FMS quickly to prevent unnecessary waiting times in task progression and order assignment.

IV. Evaluation

For practical analysis of the VDA 5050 and teleoperation protocol, gathering metrics by running real FMSs produces hardly reproducible results. This is due to timing requirements and unknown influencing factors on the network metrics. Therefore, we contribute an open-source synthetic traffic generator[1], simulating the traffic patterns of the VDA 5050 and the

[1]https://github.com/tudo-cni/VDA-5050-Traffic-Generator

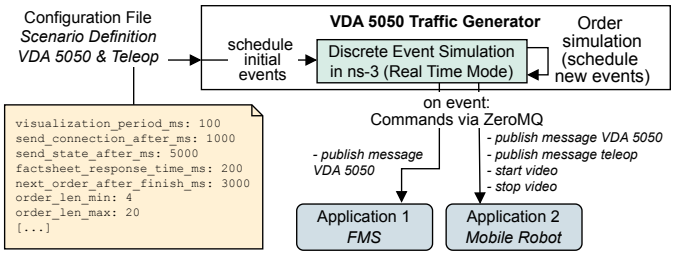


Fig. 4: Working method of the proposed open-sourced VDA 5050 Traffic Generator to simulate FMS and teleoperation interaction and generate real-time traffic flows.

teleoperation protocol. This traffic generator is used to evaluate the end-to-end latency and overhead of different middleware configurations and the proposed teleoperation extension over arbitrary networks, including industrial Open RAN networks.

The working method of the VDA 5050 traffic generator is illustrated in Fig. 4. The generator takes a configuration file specifying the scenario, including timing information for message transmissions. Using this timing, it schedules events via the discrete event simulator ns-3 in real-time mode [23], leveraging ns-3 solely for deterministic event scheduling, not network simulation. Each event triggers actions such as publishing VDA 5050 or teleoperation messages, or starting/stopping the teleoperation video stream. Commands are then forwarded to client applications for execution using ZeroMQ sockets. This abstraction keeps the traffic generator middleware-agnostic. Based on the requirements in the VDA 5050 communication, the traffic generator supports two client applications, representing the mobile robot and the FMS.

For simulating the VDA 5050 message exchange, the full setup of a VDA 5050 connection, including the *Connection* message and a *Factsheet* request and response is simulated. Further, regular *State* and *Visualization* messages are published. Executing orders is simulated by generating orders with semi-random length and sending them, also randomly partwise as within the FMS, to the mobile robot. The order is "executed" by the simulated mobile robot by finishing one step after every configured time interval and sending a new state update according to the VDA 5050 protocol. After completion, a new order process is started after a given delay. For teleoperation, additionally, the teleoperation state and simulated control inputs are published with a configurable frequency. This traffic generator is used in later parts of this work for evaluation.

A. Evaluation Setup for MQTT and Zenoh via Open RAN

The end-to-end evaluation setup depicted in Figure 5 includes a full-stack Open RAN network, containing a simulated mobile robot with a User Equipment (UE) as well as applications running in the wired part of the network. Evaluation apps were implemented in C++ and use a middleware to publish and receive messages. On mobile robot side, the evaluation app contains two middleware clients as used within the FMS for the VDA 5050 and teleoperation extension, as well as a GStreamer pipeline for video streaming. The evaluation app

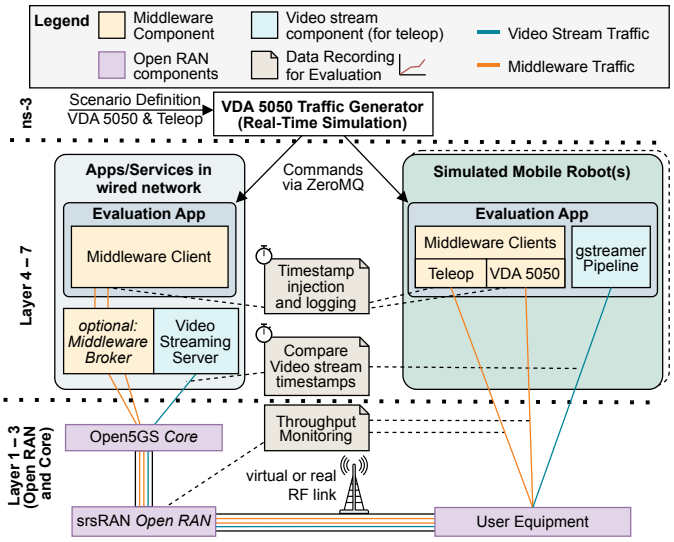


Fig. 5: Software and hardware in the simulation loop evaluation setup to run middleware and video stream traffic over different industrial networks, here via wireless Open RAN.

on the wired network side contains a single middleware client. As presented earlier, the traffic generator orchestrates the evaluation apps with respect to when they should publish messages or enable teleoperation. When middleware servers/routers are used, they are deployed on the wired side of the network. The video streaming server MediaMTX, used for the teleoperation video stream, is also operated within the wired network.

For MQTT, we use the Eclipse Paho MQTT C++ client together with a Mosquitto broker. For Zenoh, we employ the C++ API (via its wrapper) and a Docker-deployed Zenoh router. Zenoh is operated in client and peer modes over the transport protocols UDP, TCP, and Quick UDP Internet Connections (QUIC). To avoid discovery-induced background traffic, Zenoh's multicast-based auto-discovery is disabled and endpoints are configured explicitly.

The end-to-end Open RAN network stack is implemented using Open5GS as the core network and srsRAN as Open RAN stack. The rigorous controlled testing is based on a virtual-radio testbed for highest repeatability. As UE, we employ the software-based srsUE capable of 5G Standalone, exposing a TUN network device for applications to transmit data over. This TUN device operates in a separate Linux network namespace to ensure isolation. The air interface is virtualized by transferring I/Q samples via ZeroMQ. The benchmark is

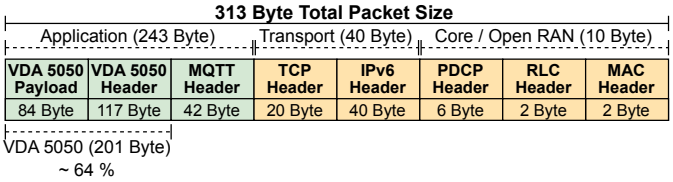


Fig. 6: Protocol overhead for one VDA 5050 *Visualization* MQTT message across all layers via the Open RAN network. Approx. 64% of the packet size is the actual VDA 5050 data.

TABLE III: Consolidated Evaluation Parameters of the Open RAN Networks and the VDA 5050 Traffic Generator.

Cellular Open RAN Network Parameters		
Parameter	**Virtual-Radio Testbed**	**Real-World Testbed**
Subcarrier Spacing	15 kHz	30 kHz
Bandwidth	10 MHz	100 MHz
Frequency, Band	1.8425 GHz, n3	3.75 GHz, n78
Duplexing Mode	Frequency Division	Time Division
User Equipment	srsUE (software)	Quectel RM520N-GL
Max. Transm. Unit	1400 B	
VDA 5050 Scenario Traffic Generator Parameters (excerpt)		
Message Type	**Timing**	**Value**
Connection	Start offset	1 s
Factsheet Request	Start offset	6 s
Factsheet Response	After request	200 ms
Visualization	Interval Initial offset	100 ms 1200 ms
Order	Start offset Order Length	8 s 4 to 20 (unif. random)

executed on a single machine running Ubuntu 22.04 with an Intel i7-4790K processor and 16 GB of RAM. For real-world transfer of the teleoperation, we utilize a private Open RAN testbed with Benetel RAN550 radio unit and physical modem, as illustrated in Fig. 7. Key network parameters are summarized in Table III. A sample VDA 5050 message via the Open RAN is shown in Fig. 6.

To measure end-to-end latency, the evaluation app injects synchronized timestamps into the payload when sending messages. By replacing parts of the actual payload, the message size is not altered. When receiving a message, the evaluation app tracks the send and receive timestamps for later evaluation. Measuring the latency of the video stream is handled by recording the traffic at the network interface on UE side and on the socket of the core network when delivered to the video streaming server. Sequence numbers within the header of the used RTP stream are used to correlate traffic and calculate the latency. The throughput is measured by recording the actual payload of all messages transmitted, which is compared to the

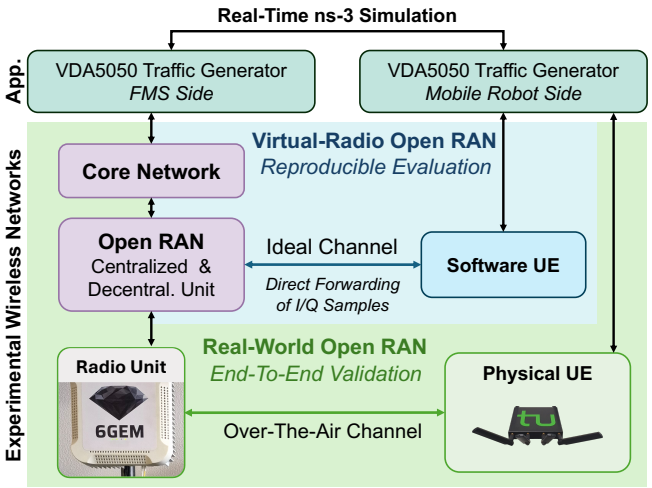


Fig. 7: Testbed architecture of controlled virtual-radio and real-world Open RAN network testbed for evaluation of VDA 5050 and teleoperation in open industrial networks.

TABLE IV: Parameters of the evaluated middleware configurations (bold indicates values used in the evaluation).

Property	**MQTT**	**Zenoh**		
Transport	**TCP**	**TCP**	**UDP**	**QUIC**
Patterns	**Pub/Sub**	**Pub/Sub**, Store/Query		
Application Layer QoS	**at most once (0)**, at least once (1), exactly once (2)	not explicitly		
System Architecture	**Brokered**	**Brokered, Peer-to-Peer**, Hybrid		
Encryption	TLS - **disabled**	TLS - **disabled**	–	**Native (QUIC)**
Discovery Mode	N/A; **static**	Dynamic (mcast); full; static; **hybrid: discovery static central**		
Latency parameters	**TCP_NODELAY enabled**	Express Flag		

packet sizes on Radio Link Control (RLC) layer of the Open RAN, excluding signalling messages on RLC layer.

B. Evaluation Scenarios

To reflect different usage behaviors in the FMSs, two evaluation scenarios were defined:

In the ***VDA 5050 Scenario***, a single mobile robot executes orders using the VDA 5050 protocol, and the evaluation focuses exclusively on the network traffic transmitted via the middleware. The scenario comprises 60 s of order execution. For the evaluation, we use the traffic generator presented earlier, configured with the parameters listed in Table III.

In the ***Teleoperation Scenario***, a single mobile robot executes orders via the VDA 5050 protocol, managed by the FMS. Control commands are sent to the mobile robot via the middleware at 20 Hz. Teleoperation mode is activated for 20 s within the time window from 20 s to 40 s, following the implementation introduced in Sec. III. During teleoperation, the mobile robot transmits a video stream over the Open RAN network with WXGA resolution (1280×800), 60 frames per second, H.264 encoding, and a bitrate of 2 Mbit/s. This scenario also uses the traffic generator with the same parameters used for the VDA 5050 scenario, except for the additional teleoperation interval, where no orders are executed.

C. Evaluation of of MQTT and Zenoh via Open RAN

Fig. 9 presents the measured end-to-end latencies in the VDA 5050 and teleoperation scenario for different middleware choices. We compare MQTT, used as the recommended middleware in VDA 5050 with Zenoh, which is increasingly adopted in robotics [24] as shown in Table IV. To ensure comparability, all middleware configurations publish identical payloads at identical rates.

When large fleets of mobile robots are operated simultaneously over a shared wireless network, protocol overhead has to be minimized to maximize the available capacity for payload data. Hence, we perform an overhead comparison of the VDA 5050 scenario using MQTT and different Zenoh configurations over a cellular network. The overhead is evaluated on the Open RAN RLC layer operating in acknowledged mode, which is the standard mode for data transmission, providing fast retransmissions of lost packets

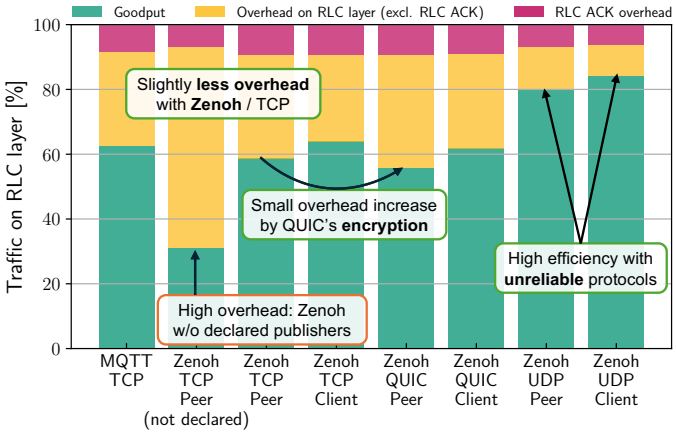
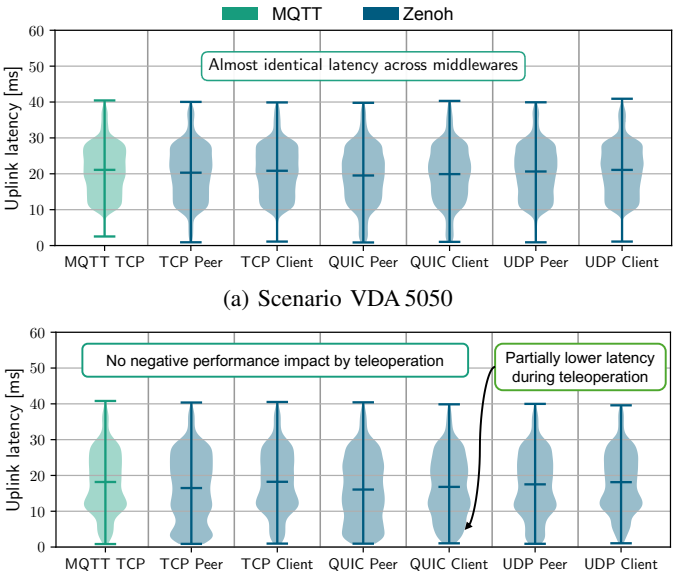


Fig. 8: Protocol overhead for VDA 5050 communication over MQTT and Zenoh at the Open RAN RLC layer under various configurations. Zenoh allows allows for flexible selection of transport options based on the required reliability.



(a) Scenario VDA 5050

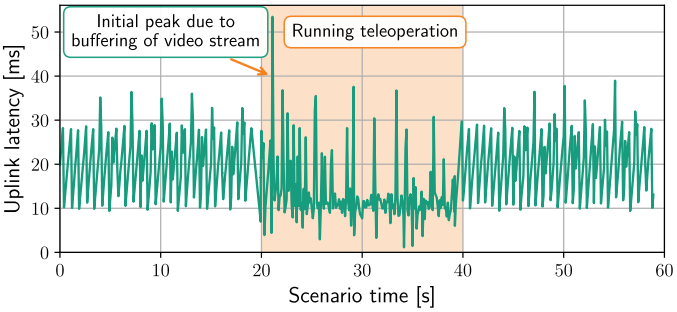
(b) Scenario Teleoperation (Initial video buffer flush omitted)

Fig. 9: End-To-End uplink latency over a reproducible virtual-radio Open RAN link across all middleware configurations.

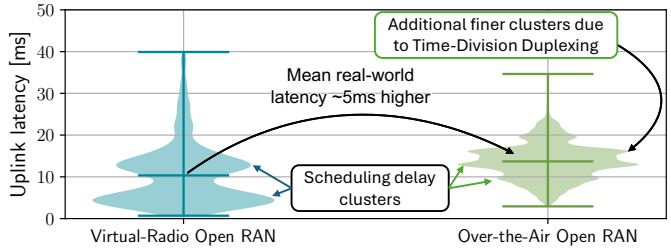
to ensure reliable data delivery. The overhead analysis is shown in Fig. 8 comprising the goodput and the additional overhead by headers as shown in Fig. 6 and acknowledgments. When using MQTT, each message incurs overhead due to the reliable TCP transmission over a reliable wireless network. When moving to Zenoh, different transport protocols can be additionally used. While using TCP as transport protocol in Zenoh incurs comparable overhead due to the reliable transmission, the overhead is slightly reduced when operating Zenoh in client mode. The encrypted QUIC transport creates small additional overhead compared to non-encrypted TCP, as expected. Transmissions over unreliable UDP in a reliable network reduces the overhead significantly, as no transport-layer acknowledgments are required. The lowest overhead is achieved using Zenoh in client mode over UDP reducing the overhead by nearly half compared to MQTT. Here, no transport-layer acknowledgments are required. Critically, declaring publishers in Zenoh before pushing data significantly reduces protocol overhead [25], as routing and subsription information is exchanged only once during the declaration phase instead of with every message.

Within the VDA 5050 scenario, Fig. 9a, all middleware setups perform nearly the same. The average latency is around 20 ms, while at maximum, a latency of 40 ms occurs. This same behavior across all middlewares, even with different transport protocols occurs as the network has sufficient capacity and therefore the packets are propagated without congestion or packet loss. Compared to the latency distribution of the teleoperation scenario in Fig. 9b, the observed latencies are similar in range but approximately 3 ms lower than those in the VDA 5050 scenario. To analyse this behavior, Fig. 10a displays the MQTT uplink latency over time of a single teleoperation scenario run.

Teleoperation, including the video stream, runs from around 20 s to 40 s within the scenario. When starting the teleoperation, a latency peak can be observed, which is omitted in Fig. 9b as the transient state does not influence normal teleoperation. This behavior is due to an initial burst of the video stream, flushing the video stream buffer, which was filled before connecting to the media server. While the teleoperation is running, the average latency is reduced to around 11 ms. This likely happens as the regular video stream traffic ensures that no explicit Scheduling Requests (SRs) are required to schedule uplink traffic caused by the regular and large grants



(a) End-to-end uplink latency for the middleware traffic over time; teleoperation is active approximately from 20 s to 40 s.



(b) End-to-end video-stream latency from UE to media server via UDP. The initial latency peak caused by video-buffer flush is omitted.

Fig. 10: Teleoperation Scenario over the virtual-radio and the over-the-air Open RAN testbed with VDA 5050 and MQTT.

for video traffic. Therefore, the size of the RLC layer buffer ready to be sent can be sent to the base station via Buffer Status Reports (BSRs) instead of using SRs, which can only be sent within SR occasions, set to every $20\,\mathrm{ms}$ in the virtual-radio testbed. Hence, the latency improves if a moderately increased, constant uplink traffic flow for a UE is provided.

The video stream latency of a single run in the teleoperation scenario with MQTT is shown in Fig. 10b. In the $20\,\mathrm{s}$ video stream, an initial latency peak of up to $60\,\mathrm{ms}$ was observed, occurring during an initial traffic burst by aggregated buffered video data. For statistical analysis of the long-term latency, this transient spike is excluded in Fig. 10b. The mean latency reaches around $10.35\,\mathrm{ms}$. Much of the delay is clustered at around $4\,\mathrm{ms}$ and $14\,\mathrm{ms}$ due to scheduling delays, depending on SR necessity. In comparison with the real-world over-the-air Open RAN testbed for validation of the proposed VDA 5050 teleoperation, the video stream latency is slightly increased to $15\,\mathrm{ms}$ on average due to additional radio components, processing, and the use of time-division duplexing.

V. Conclusion and Future Work

This work motivates teleoperation as a practical complement to autonomy in intralogistics, which should be integrated at the FMS interface level to remain interoperable in heterogeneous fleets. We extend VDA 5050 with teleoperation support, provide a configurable VDA 5050 traffic generator for reproducible evaluation, and evaluate the end-to-end teleoperation stack including middleware transport and Open RAN industrial wireless connectivity. The results characterize latency and overhead implications of standardized teleoperation integration with different middleware configurations and provide a foundation for designing robust remote human-in-the-loop operation of mobile robots in VDA 5050-based FMSs.

Future work will focus on a joint optimization of application and Open RAN network for low-latency teleoperation, e.g., by advanced uplink scheduling strategies such as adaptive configured grants for periodic streams to reduce latency and jitter under load. Beyond link-level optimization, further work can explore adaptive traffic shaping and QoS mapping within the middleware for the mixed workload of VDA 5050 messages, control commands, and video uplink, as well as validation in larger-scale multi-robot scenarios with an over-the-air Open RAN network in realistic industrial channel conditions.

Acknowledgment

This work has been partially supported by the Federal Ministry of Research, Technology and Space (BMFTR) via the *6GEM+* transfer hub under the funding reference 16KIS2412.

References

[1] L. Monostori, P. Valckenaers, A. Dolgui, H. Panetto, M. Brdys, and B. C. Csáji, "Cooperative control in production and logistics," *Annual Reviews in Control*, vol. 39, pp. 12–29, 2015.

[2] J. Wan, X. Li, H.-N. Dai, A. Kusiak, M. Martínez-García, and D. Li, "Artificial-intelligence-driven customized manufacturing factory: Key technologies, applications, and challenges," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 377–398, 2021.

[3] T. Peitscher, D. Lünsch, and P. Detzner, "Multilayer graph partitioning to enable a decentralized path planning for large and heterogeneous AGV fleets," in *IEEE Int. Conf. on Automation Science and Engineering*, 2024.

[4] G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journal of Operational Research*, vol. 294, no. 2, pp. 405–426, 2021.

[5] M. De Ryck, M. Versteyhe, and F. Debrouwere, "Automated guided vehicle systems, state-of-the-art control algorithms and techniques," *Journal of Manufacturing Systems*, vol. 54, pp. 152–173, 2020.

[6] M. Brilhante, P. M. Rebelo, P. M. Oliveira, H. Sobreira, and P. Costa, "Control of a mobile robot through VDA5050 standard," in *Robot 2023: Sixth Iberian Robotics Conference*, 2024, pp. 541–552.

[7] S. Franke, D. Lünsch, J. Jost, and M. Roidl, "Identification of requirements and opportunities for new types of standardized interfaces for AGV systems based on the VDA 5050 concept," *Logistics Journal: Proceedings*, vol. 2023, no. 1, 2023.

[8] M. Moniruzzaman, A. Rassau, D. Chai, and S. M. S. Islam, "Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey," *Robotics and Autonomous Systems*, 2022.

[9] M. Zhan and K. Yu, "Wireless communication technologies in automated guided vehicles: Survey and analysis," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018.

[10] C. Arendt, S. Böcker, H. Schippers, T. Ploch, M. Kuhn, V. Venjakob, S. Hunger, and C. Wietfeld, "Towards future industrial connectivity: Evaluation of private 5G and Wi-Fi networks in professional industrial environments," in *IEEE Int. Conf. on Factory Commun. Systems*, 2025.

[11] N. A. Wagner, J. Eßer, I. F. Priyanta, F. Kurtz, M. Roidl, and C. Wietfeld, "Real-time predictive scheduling for networked robot control using digital twins and OpenRAN," in *IEEE Globecom Workshops*, 2024.

[12] T. Sheridan, "Telerobotics," *Automatica*, vol. 25, no. 4, 1989.

[13] T. B. Levin, J. M. Oliveira, R. B. Sousa, M. F. Silva, B. S. Parreira, H. M. Sobreira, and H. S. Mendonça, "Image and command transmission over the 5G network for teleoperation of mobile robots," in *2024 7th Iberian Robotics Conference (ROBOT)*, 2024, pp. 1–8.

[14] one6G Association, "6G technology overview," 2025. [Online]. Available: https://one6g.org/one6g-releases-the-5th-edition-of-the-6g-technology-overview-white-paper/ [Accessed: Mar. 6, 2026]

[15] H. Schippers, T. Gebauer, K. Heimann, and C. Wietfeld, "RISE: Multi-link proactive low-latency video streaming for teleoperation in fading channels," in *IEEE Veh. Technol. Conf. (VTC2025-Spring)*, 2025.

[16] J. S. Lee, Y. Ham, H. Park, and J. Kim, "Challenges, tasks, and opportunities in teleoperation of excavator toward human-in-the-loop construction automation," *Automation in Construction*, vol. 135, 2022.

[17] S. M. Sajadi, K. Mathiassen, H. Solvin, H. Brun, P. H. Lehne, and O. J. Elle, "Ultrasound robotic-assisted teleoperation over networks: Performance evaluation of commercial 5G, Wi-Fi, and Ethernet," in *Int. Conf. on Mechatronics and Robotics Engineering (ICMRE)*, 2025.

[18] K. Yaovaja, P. Bamrungthai, and P. Ketsarapong, "Design of an autonomous tracked mower robot using vision-based remote control," in *IEEE Eurasia Conf. on IOT, Communication and Engineering*, 2019.

[19] N. A. Wagner and C. Wietfeld, "O-RACES: proactive AI-driven scheduling in Open RAN for 6G-networked humanoid robots," in *IEEE Conf. on Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2026.

[20] M. B. Couto, M. R. Petry, A. Mendes, and M. F. Silva, "Virtual Reality-Based Teleoperation System for Robot Forklifts," in *IEEE Int. Conf. on Autonomous Robot Systems and Competitions (ICARSC)*, 2025.

[21] D. Lünsch, P. Detzner, A. Ebner, and S. Kerner, "SWAP-IT: A scalable and lightweight Industry 4.0 architecture for cyber-physical production systems," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022, pp. 312–318.

[22] S. B. Kamtam, Q. Lu, F. Bouali, O. C. L. Haas, and S. Birrell, "Network latency in teleoperation of connected and autonomous vehicles: A review of trends, challenges, and mitigation strategies," *Sensors*, vol. 24, no. 12, 2024.

[23] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34.

[24] ROS 2 Core Team, "2023-09 ROS 2 RMW alternate," 2023. [Online]. Available: https://discourse.openrobotics.org/uploads/short-url/o9ihvSjCwB8LkzRklpKdeesRTDi.pdf [Accessed: Mar. 6, 2026]

[25] A. Corsaro, L. Cominardi, O. Hecart, G. Baldoni, J. E. P. Avital, J. Loudet, C. Guimares, M. Ilyin, and D. Bannov, "Zenoh: Unifying communication, storage and computation from the cloud to the microcontroller," in *Euromicro Conference on Digital System Design*, 2023.