This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Enabling AI-Driven Real-Time Communication and Computation With DDS for IoT-Assisted Home Care

Ching-Han Chen, Chih-Hao Chang, Chih-Wei Peng, and Guan-Wei Lan

Abstract

As aging populations drive demand for smart home–based care, edge-enabled IoT systems must provide real-time responsiveness, reliability, and on-device intelligence. However, Message Queuing Telemetry Transport (MQTT) and the Constrained Application Protocol (CoAP) often fall short for high-frequency, multimodal, AI-driven workloads. We present a practical framework that uses the Data Distribution Service (DDS) as the backbone for scalable, low-latency, and resilient Artificial Intelligence of Things (AIoT) deployments. The design supports dynamic node discovery, automated Quality of Service (QoS) tuning, and containerized rollout across heterogeneous edge devices. It integrates AI-based video and audio analytics to detect critical events in real time, while a brokerless DDS model reduces latency and removes single points of failure. We provide implementation steps and lessons learned from a replicable gateway–sensor architecture. In experiments across four image-delivery scenarios (single- and multi-device), DDS sustained higher throughput and packet rate than MQTT and avoided the 14% packet loss observed for MQTT under multi-device transmission. These results indicate that DDS is an efficient and robust foundation for AI-powered home-care edge networks.

I. Introduction

Smart aging-in-place solutions are reshaping how elderly individuals receive care, enabling continuous health monitoring and timely assistance within their own homes. These systems typically rely on edge-enabled Internet of Things (IoT) devices—such as cameras, microphones, and wearables—that must communicate and respond in real time to ensure safety and improve quality of life [1]. As the complexity of these systems increases, so does the demand for a robust communication and computation infrastructure at the network edge.

Today's mainstream IoT communication protocols, including Message Queuing Telemetry Transport (MQTT) [2] and the Constrained Application Protocol (CoAP) [3], are lightweight and easy to integrate. However, they often fall short when used in AI-powered scenarios that require continuous processing of multimodal sensor streams—such as video, audio, and physiological data. These protocols typically rely on centralized brokers and are not optimized for high-throughput, low-latency, or scalable deployments [4]. In critical care scenarios, such limitations can hinder real-time detection of abnormal events and delay emergency response.

As edge computing becomes more prominent, integrating AI models directly on distributed nodes introduces new challenges in system coordination, resource management, and communication efficiency [5]. Centralized architectures struggle to scale and introduce single points of failure, making them less ideal for long-term care environments where system uptime and responsiveness are paramount.

To address these challenges, this article explores the use of DDS—a real-time, peer-to-peer middleware originally developed for mission-critical systems. DDS offers fine-grained QoS, low-latency publish–subscribe communication, and decentralized operation. It is well suited to intelligent edge-based IoT systems [6]. DDS has also gained momentum in robotics frameworks such as Robot Operating System 2 (ROS 2) and Micro-ROS, underscoring its suitability for time-sensitive, heterogeneous deployments [7], [8].

Despite its technical advantages, deploying DDS in practical care settings remains a complex task. Engineers must manage dynamic node configuration, interoperability across diverse devices, edge containerization, and AI model integration. Ensuring reliable, low-latency transmission of multimodal data—while keeping the system simple and scalable—remains nontrivial [9], [10], [11].

We present a practical framework for building DDS-powered AIoT systems in smart care environments. It supports real-time video and audio event detection at the edge, dynamic QoS tuning, plug-and-play device discovery, and containerized deployment for rapid prototyping and scaling. We also share insights from real-world testing to provide a replicable foundation for deploying edge intelligence in long-term home care applications [12].

II. Background

Modern IoT systems rely on efficient communication protocols to connect sensors, devices, and edge intelligence across distributed environments. Two of the

The authors are with the Department of Computer Science and Information Engineering, National Central University, Taoyuan 320009, Taiwan. Chih-Hao Chang is the corresponding author.

Digital Object Identifier: 10.1109/MIOT.2025.3637839

2576-3180 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

most commonly adopted protocols in low-power IoT applications are MQTT and CoAP [2], [3]. Both are designed for constrained devices and networks, offering lightweight communication with minimal overhead.

MQTT follows a publish-subscribe model and is popular for its simplicity and ease of integration. However, it depends on a centralized broker, which introduces latency, limits scalability, and creates a single point of failure. These characteristics become problematic in real-time applications such as health monitoring, fall detection, or audio event recognition—where immediate reaction is essential [4].

CoAP adopts a request-response pattern similar to HTTP but optimized for constrained environments. While it performs well in simple sensor networks, it struggles to support high-frequency, continuous data streams—such as video or audio feeds from smart care systems. Moreover, both MQTT and CoAP offer limited QoS configuration, which restricts their usefulness in latency-sensitive or mission-critical scenarios.

To overcome these limitations, DDS has emerged as a powerful alternative. Originally developed for defense and aerospace applications, DDS uses a decentralized publish-subscribe model that eliminates the need for a central broker [6]. This peer-to-peer architecture significantly reduces latency, improves system reliability, and enhances scalability—key requirements for edge-based AIoT deployments.

One of DDS's core strengths lies in its fine-grained QoS control. Developers can tune parameters such as data delivery reliability, deadline constraints, resource limits, and message durability. These features make DDS especially well-suited for real-time AI applications where deterministic performance is required, including robotics systems built with ROS 2 and Micro-ROS [7], [8].

Furthermore, DDS facilitates seamless communication among heterogeneous nodes, enabling integration of diverse sensing and computing platforms without custom protocol handling. This interoperability is particularly beneficial in edge-enabled smart environments that combine legacy systems with AI-capable devices.

However, adopting DDS in real-world systems is not without challenges. Engineers must handle dynamic node discovery, edge resource constraints, and the complexity of configuring QoS profiles in distributed deployments [9], [10], [11]. These deployment hurdles can be a barrier for practitioners seeking to implement DDS in healthcare or home monitoring scenarios.

In the next section, we present a practical DDS-based deployment platform designed to address these issues directly. Our framework enables real-time edge analytics with plug-and-play device integration, robust QoS configuration, and containerized deployment for AI-powered long-term care systems.

III. Enabling Technologies for AI-Powered Edge IoT Systems

Modern AIoT systems—where artificial intelligence meets the Internet of Things—rely on the seamless coordination of sensing, computing, and communication components distributed across the network edge. This section introduces key enabling technologies that make such integration feasible: AI-enhanced IoT (AIoT), DDS, ROS 2 and Micro-ROS for heterogeneous device orchestration, and edge computing strategies.

A. AI-Enhanced IoT (AIoT)

The convergence of IoT and AI technologies has led to the emergence of AIoT systems, where smart sensing devices not only capture environmental data but also analyze and act on it locally. Unlike traditional IoT systems that forward all data to the cloud, AIoT pushes intelligence closer to where data is generated—at the edge. This shift reduces latency, enhances privacy, and improves system responsiveness, making it ideal for time-sensitive applications such as fall detection or sound anomaly recognition in home care settings.

To support such intelligence, edge devices must perform real-time preprocessing, while gateways are expected to handle complex multimodal data and apply context-aware decision logic. As noted in [9], gateways in AIoT systems face three critical challenges: system scalability, ease of deployment, and reliable data governance across heterogeneous sources. QoS policies play a key role here, defining latency, reliability, and bandwidth constraints to ensure stable operation under dynamic conditions.

B. Data Distribution Service

DDS is a middleware standard developed by the Object Management Group (OMG) to support real-time data exchange in distributed systems [6]. Its decentralized publish-subscribe model eliminates the need for a central broker, enabling peer-to-peer communication among IoT devices. This architecture improves scalability and removes single points of failure—crucial for mission-critical, latency-sensitive applications.

One of DDS's main advantages is its comprehensive QoS configuration, which lets developers define fine-grained guarantees on reliability, deadline, and resource usage. This flexibility suits robotics and AIoT deployments and underpins platforms like ROS 2 and Micro-ROS [7], [8].

Fig. 1 illustrates the core DDS entities—Publisher, Subscriber, DataWriter, and DataReader—organized under domains and topics. DDS handles data flow implicitly, based on topic subscriptions and configured QoS profiles, reducing development complexity while ensuring predictable performance.

Despite these strengths, DDS deployment can be complex. Developers must configure dynamic node discovery, tailor QoS to each application's real-time needs, and optimize performance across a range of hardware platforms [9], [10], [11].

C. ROS 2 and Micro-ROS

To manage distributed AIoT systems involving heterogeneous devices—from edge servers to microcontrollers—middleware like ROS 2 and Micro-ROS provides a unified development and communication framework. ROS 2 leverages DDS for reliable, low-latency communication across robotic and embedded platforms. However, its full implementation can be too resource-intensive for MCU-class devices.

Micro-ROS addresses this limitation by introducing a lightweight variant of DDS—Micro eXtremely Resource-Constrained Environment Data Distribution Service (Micro XRCE-DDS)—optimized for microcontrollers [13]. It follows a Client-Agent model: the Micro-ROS Client runs on the microcontroller unit (MCU), while the Agent, hosted on a more capable

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

device, handles full DDS communication on its behalf. This structure allows low-power devices to participate in DDS networks without the overhead of running a full DDS stack.

As shown in Fig. 2, the ROS 2 ecosystem supports a layered approach, separating hardware interaction, middleware configuration, and application logic. This modularity simplifies system design and accelerates development in complex AIoT environments [14].

D. Edge Computing for Responsive AIoT Systems

Edge computing plays a vital role in AIoT by enabling data processing to occur near the data source rather than in centralized clouds. This reduces communication latency, lowers bandwidth usage, and enhances privacy—critical factors in healthcare and smart home applications [5].

To support edge intelligence, gateways must handle protocol conversion, local data aggregation, and task scheduling for multiple sensing streams. These devices often serve as the first line of analysis, determining whether to respond locally or escalate data to cloud services.

Modular gateway architectures, as proposed by Chen et al. [10], allow real-time resource allocation based on current workloads, enabling the system to scale while maintaining performance. When combined with DDS and Micro-ROS, these gateways form the backbone of resilient, intelligent, and context-aware AIoT networks.

In the following section, we present our DDS-based deployment framework, which combines the strengths of these technologies into a unified platform. It supports plug-and-play edge devices, automated QoS configuration, containerized deployment, and real-time AI analytics tailored for home-based long-term care scenarios.

IV. System Architecture

To support real-time multimodal AI processing in smart eldercare environments, the proposed system adopts a modular architecture comprising two main components: a central system gateway and a set of peripheral sensor nodes. Peripheral devices—based on ESP32-S3 modules—capture and transmit video or audio data, while the gateway manages data routing, classification, AI inference, and user notification. This architecture allows edge devices to function as intelligent publishers in a fully decentralized DDS network, with the gateway executing real-time event detection and alert delivery.

A. Peripheral Layer: ESP32-S3 Nodes

The peripheral layer consists of ESP32-S3 devices equipped with either image sensors (OV2640) or MEMS microphones. These compact nodes stream raw video or audio data over Wi-Fi using the Micro XRCE-DDS protocol. Each node autonomously loads environment-specific parameters and applies predefined QoS profiles to ensure reliable data delivery. Thanks to a plug-and-play architecture, devices can join the network automatically, minimizing configuration overhead and enabling flexible deployment.

B. Central Gateway: Raspberry Pi 5

The gateway, built on a Raspberry Pi 5, serves as the central processing and communication hub. It hosts a Micro-ROS Agent container that discovers active nodes and routes their DDS messages to internal ROS 2 topics. A multimodal classification module subscribes to these topics, parses the incoming data based on modality, and dispatches it to the corresponding AI inference modules.

Video streams are processed by a You Only Look Once (YOLO) v8-based visual detection module accelerated by a Hailo-8L AI processor, supporting real-time fall detection. In parallel, audio packets are analyzed by a lightweight YAMNet classifier running on the Raspberry Pi central processing unit (CPU), capable of detecting events such as coughing or glass breaking. When an anomaly is detected, alerts are sent to registered caregivers via the LINE Bot application programming interface (API). To support remote access, a network bridging module performs NAT traversal, allowing outbound connectivity and integration with cloud-based services.

This modular gateway-sensor design supports scalable, low-latency edge intelligence and flexible integration of heterogeneous devices—making it well-suited for real-world deployment in home-based healthcare scenarios.

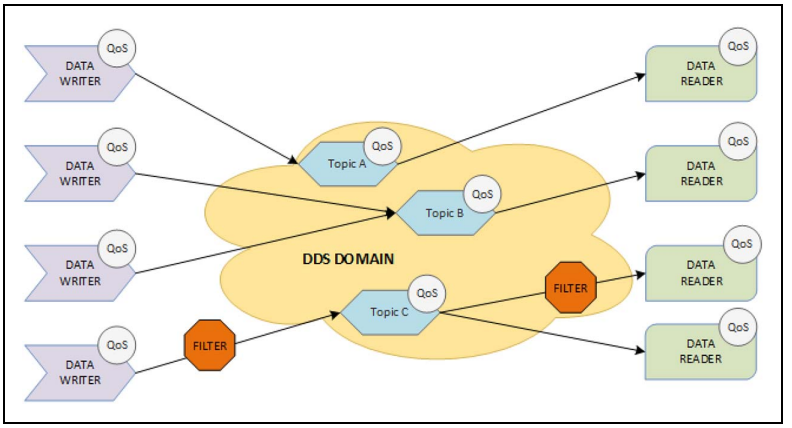


FIG. 1. DDS communication model with QoS and filtering mechanisms.

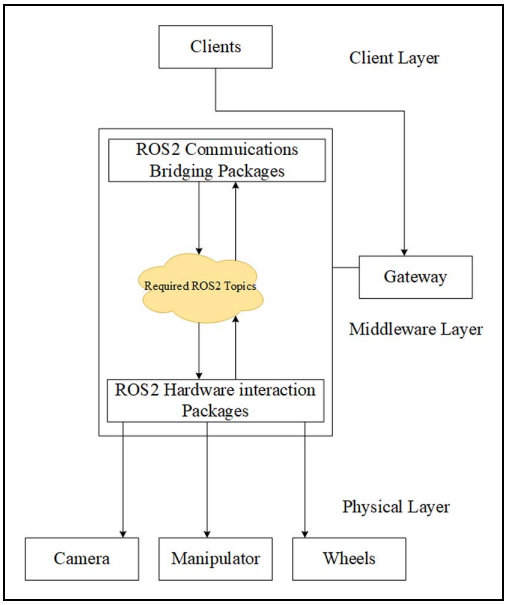


FIG. 2. Layered ROS 2 architecture for distributed AIoT systems.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

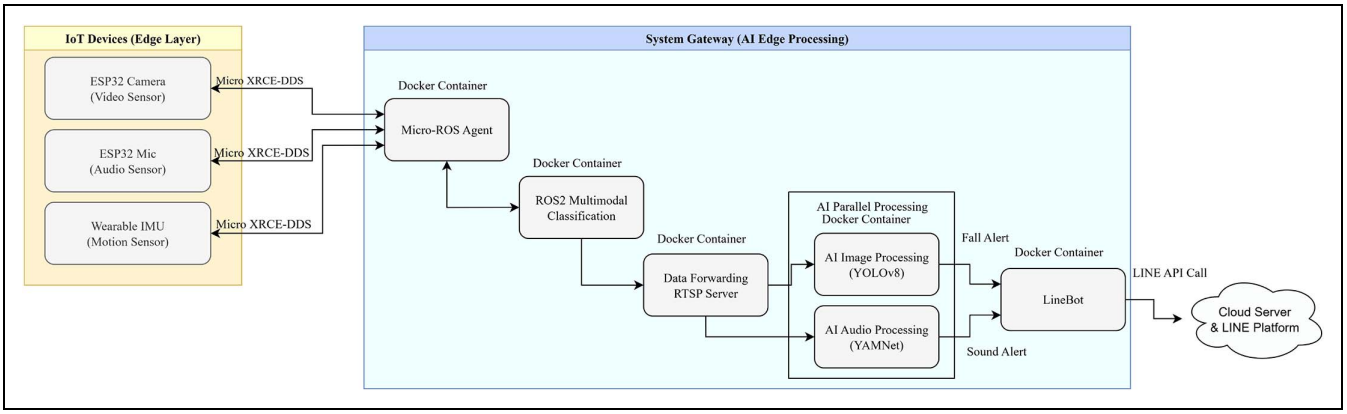


FIG. 3. Modular design of the system gateway, from data reception to user notification.

C. Gateway Functional Modules

The gateway is composed of six key modules, each responsible for a distinct role in the data processing pipeline. As illustrated in Fig. 3, this modular design ensures clear functional separation, scalability, and ease of maintenance.

- **Micro-ROS Agent:** Acts as the interface between the DDS network and the gateway. It automatically discovers Micro-ROS nodes, subscribes to their topics, and stores incoming messages in shared buffers. Payloads remain unmodified, ensuring interoperability with heterogeneous devices.
- **Multimodal Classifier (ROS 2):** Identifies the data type (image or audio) of each packet and reformats it into a consistent structure. It supports optional preprocessing—such as resolution scaling or compression—to improve AI inference efficiency.
- **Data Forwarding Server:** Converts processed data into Real Time Streaming Protocol (RTSP)-compatible streams and establishes separate channels for each modality. These streams can be consumed by visualization dashboards or cloud-based applications in real time.
- **AI Image Processor:** Applies a YOLOv8 model to detect visual anomalies, such as human falls. A postural check verifies predictions before alerts. YOLOv8 offers high inference speed suited to edge hardware.
- **AI Audio Processor:** Uses YAMNet, a lightweight convolutional neural network (CNN) trained on AudioSet, to classify audio events such as distress sounds or glass breaking. Mel-Frequency Cepstral Coefficients (MFCC) from incoming audio enable real-time inference on the Raspberry Pi.
- **LINE Notification Module:** Sends real-time alerts to caregivers using the LINE Bot API. Messages are structured according to event type and severity, and delivered via a reliable messaging pipeline.

D. Workflow Summary

When powered on, each ESP32-S3 node connects to a predefined Wi-Fi network and joins the DDS communication domain. The Micro-ROS Agent on the gateway detects available nodes, subscribes to their topics, and forwards data to the relevant internal modules. Each module operates independently, enabling robust, parallel processing of multimodal streams.

This decoupled architecture supports dynamic node discovery, real-time AI inference, and low-latency coordination across audio and video sources. It provides a scalable, fault-tolerant framework suitable for engineers building intelligent, responsive edge systems for home-based healthcare.

V. System Implementation and Edge-AI Deployment

Fig. 4 presents a conceptual illustration of the system experiment. To demonstrate real-time multimodal sensing and inference at the edge, we implemented a practical gateway–sensor architecture centered on DDS and Micro-ROS. This section outlines the system hardware, configuration approach, and testing methodology to guide replicable deployment.

A. Edge-AI Hardware Configuration

The gateway was built using a Raspberry Pi 5, selected for its balance of performance and form factor. It features a 64-bit quad-core Cortex-A76 processor, 8 GB of random access memory (RAM), Gigabit Ethernet, dual-band Wi-Fi, and USB 3.0 interfaces—making it suitable for high-throughput data acquisition and on-site processing.

For edge sensing, we employed two types of ESP32-S3 nodes: one equipped with an onboard OV2640 camera (ESP32-S3-EYE) for video input, and another with an Inter-IC Sound (I2S) Micro-Electro-Mechanical Systems (MEMS) microphone for audio. The ESP32-S3 offers a dual-core LX7 processor running at 240 MHz with built-in Wi-Fi, enabling efficient multimodal data capture and wireless streaming.

To enable real-time AI inference on the gateway, we integrated a Hailo-8L AI accelerator via the M.2 interface. Supporting up to 13 trillions of operations per second (TOPS), this module runs YOLOv8-based object detection in real time. Audio classification was handled by a pre-trained YAMNet model running on the gateway CPU, using MFCC features extracted from incoming audio streams.

B. Deployment Workflow and Data Routing

Once powered, each ESP32-S3 node scans and connects to a predefined Wi-Fi network. It then loads custom environment parameters from flash memory and applies DDS QoS profiles for consistent data delivery. The system uses a unified development template, enabling automated startup and seamless integration.

After joining the network, each node begins streaming data via Micro-ROS to the gateway. The Micro-ROS Agent container detects new publishers and forwards data to corresponding ROS 2 topics. At boot time, the gateway pre-subscribes to these topics,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

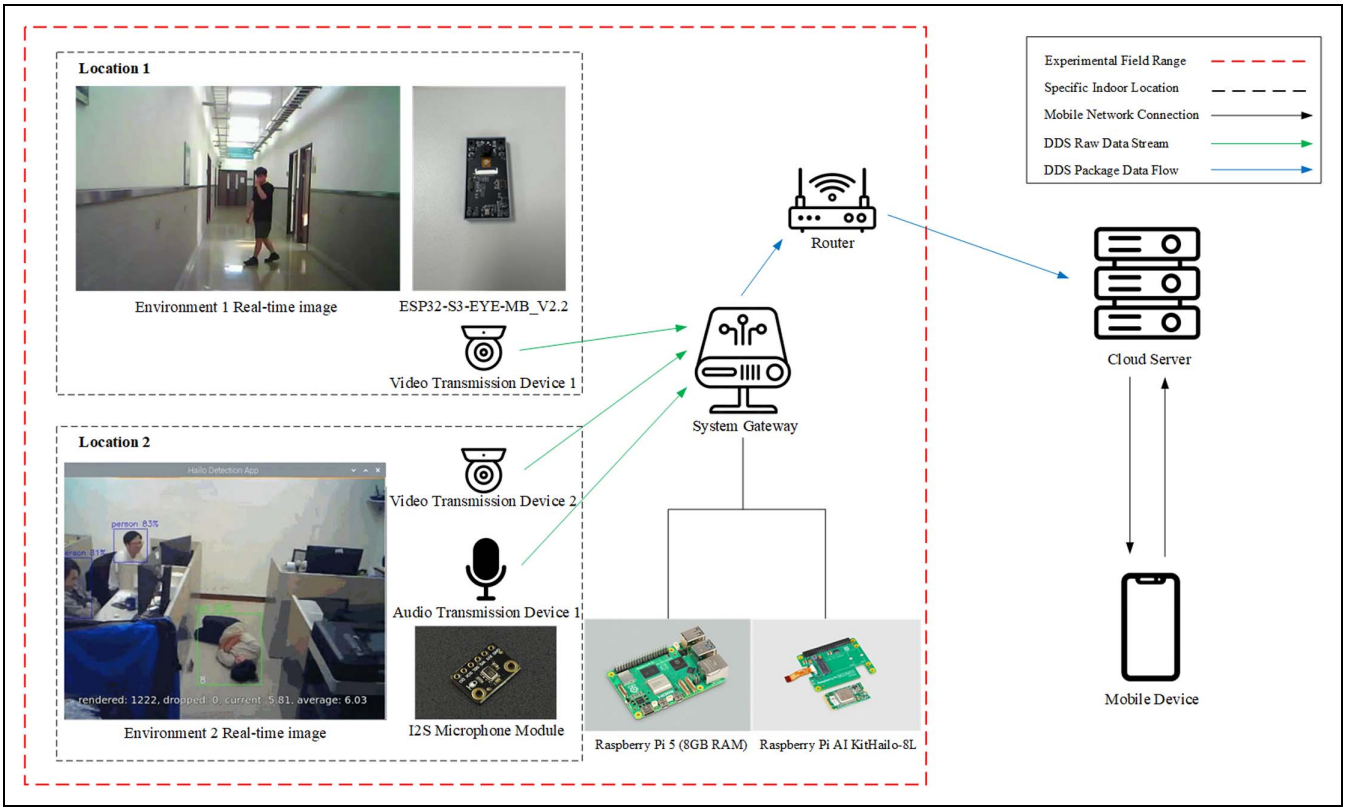


FIG. 4. Conceptual illustration of the system experiment.

allowing AI inference containers to process incoming data without manual configuration.

For video streams, the image data is passed to the YOLOv8 container for fall detection. If a fall is detected, an event is generated and routed through a notification pipeline—including an API call to a cloud service and a LINE Bot alert sent to caregivers. Similarly, audio packets are pre-processed and classified into fall, glass break, or cough events using the YAMNet container. Detected anomalies are forwarded using the same alert workflow.

C. Experimental Setup and Methodology

Network environment: The *Wireless Local Area Network (WLAN)* used an ASUS RT-BE88U access point on 2.4 GHz (channel 9, 20 MHz). The Raspberry Pi 5 gateway and ESP32-S3 nodes associated to the same *service set identifier (SSID)* using WPA2-PSK. Tests were conducted indoors with line-of-sight at approximately 7.5 m. Background traffic averaged about 0.23 MB/s; no intentional interference was introduced.

Node population and roles: Unless otherwise noted, experiments used two sensor nodes concurrently—one ESP32-S3 camera and one ESP32-S3 audio—publishing to a single Raspberry Pi 5 gateway. In the two-node scenario, video and audio were streamed simultaneously. We also validated single-node runs per modality to isolate per-stream behavior.

QoS and configuration: QoS profiles remained constant across trials. Video topics used reliable delivery with bounded history (Keep Last, depth 5). Audio topics used the ROS 2 default QoS.

Measurement procedure: Each scenario ran for 1 minute and was repeated three times. We computed throughput and packet rate (packets per second; pkt/s) at the gateway from received payload sizes and timestamps, and estimated packet loss from backend logs as the fraction of sent packets not received. In the two-node scenario, both streams remained active for the full 1-minute window in each repetition, and metrics were computed per modality.

Notes: We used 20 MHz on 2.4 GHz to improve repeatability and reduce adjacent-channel interference. WPA3-SAE is preferred when supported by all devices; our prototype used WPA2-PSK for MCU compatibility.

D. Multimodal Testing Scenarios

We simulated a typical indoor eldercare environment. ESP32-S3 nodes were mounted at fixed positions while a subject moved throughout the space and performed normal and emergency actions. The gateway received synchronized video and audio, routed data through a forwarding service, and executed real-time inference across processing containers.

We conducted tests in two stages:

- **Single-Node Validation:** Each ESP32-S3 was evaluated independently to verify connectivity and data stream integrity. Camera nodes successfully delivered image packets to the YOLOv8 module, while audio nodes transmitted 20 ms audio chunks to the YAMNet module.
- **Multimodal Operation:** With both video and audio nodes active, we verified the system's ability to handle concurrent streams, execute parallel inference, and coordinate event detection and response through a shared backend.

E. Communication Benchmark: DDS vs. MQTT

To evaluate the impact of communication protocols on real-time performance, we replaced DDS with MQTT while keeping all other system parameters constant. As shown in Fig. 5, the results are summarized as follows:

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

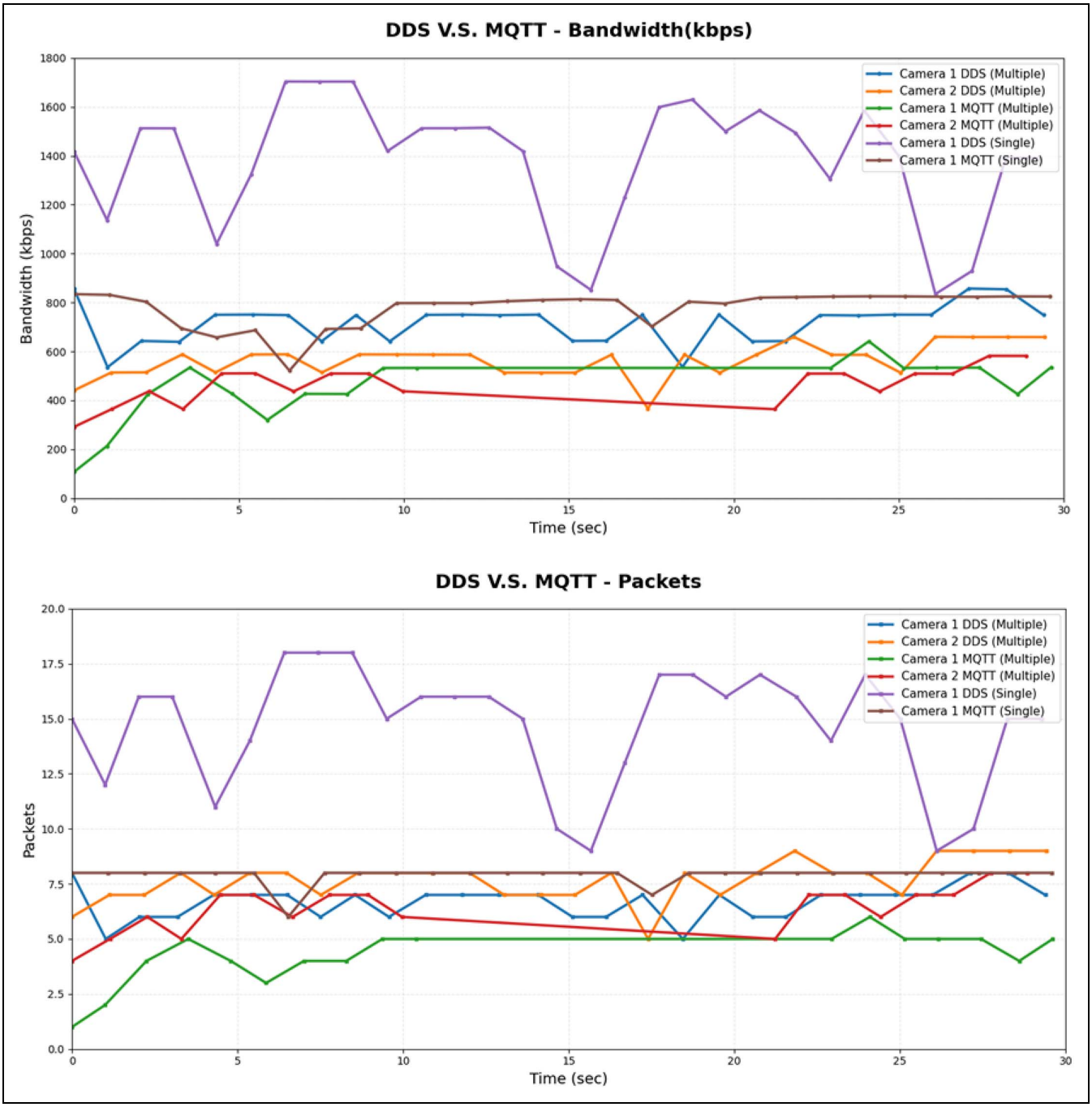


FIG. 5. DDS vs. MQTT under single- and multi-device workloads (S1–S4). Mean throughput (kbps) and packet rate (pkt/s). DDS outperforms MQTT; under multi-device, MQTT shows ĩ4% loss.

- **Single-device (S1/S2):** DDS achieved an average throughput of 1370 kbps with 14.73 packets per second (pkt/s), while MQTT reached 789 kbps with 7.93 pkt/s.
- **Multi-device (S3/S4):** When multiple devices transmitted concurrently, DDS maintained an average throughput of 640.6 kbps and approximately 7.2 pkt/s. MQTT averaged 497.18 kbps and 5.9 pkt/s and exhibited 14% packet loss at the backend.

These findings emphasize DDS's higher efficiency and robustness, especially under multi-device contention; DDS sustained higher throughput and packet rate and avoided the packet loss observed with MQTT.

F. Scalability and Practical Insights

With self-configuring nodes, new ESP32-S3 devices join by simply powering on within the network. The gateway discovers and integrates them without manual pairing, which simplifies large-scale deployment. This plug-and-play design fits real-world eldercare scenarios that require flexibility, reliability, and rapid scaling.

VI. Privacy and Security for Home-Care Edge-AIoT

- On-gateway inference and data minimization: The system executes inference on the gateway and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

forwards only event-level semantics (for example, labels and bounding boxes) instead of raw media. Raw video/audio are not uploaded by default.
- DDS Security via *ROS 2 Security (SROS2)*: We enable authenticated, encrypted sessions and per-topic permissions through governance and permission artifacts, preventing unauthorized readers/writers and enforcing least privilege.
- Secure media transport across hosts: When media streams must traverse multiple hosts, we protect real-time transport using *Secure Real-time Transport Protocol (SRTP)* or *RTSP over TLS (RTSPS)*.
- Anonymized investigative export: For investigative exports, we apply optional anonymization (for example, face blurring) to minimize identifiability beyond the edge environment.

A. Threat Model

Adversaries may eavesdrop on home WLAN links, perform man-in-the-middle or replay of sensor streams, subscribe to unauthorized topics, steal devices, or compromise the gateway. Our controls focus on least-privilege topic permissions, encrypted transport, and the ability to revoke compromised identities.

B. Data Lifecycle and Minimization

Raw media remain local in a rolling buffer of 48 hours for incident verification and are not uploaded by default. Outbound telemetry contains event-level results (for example, labels and bounding boxes) rather than raw frames or waveforms. When frames must be exported for investigative review, optional anonymization (for example, face blurring) is applied.

C. Transport, Identity, and Access Control

SROS2 is enabled with authenticated, encrypted sessions and per-topic permissions via governance and permission artifacts, preventing unauthorized readers/writers. For media streams that traverse multiple hosts, *SRTP* or *RTSPS* protects real-time transport. Outbound notifications use scoped, short-lived API tokens or mutual TLS (mTLS). Sensors use Micro-ROS (XRCE-DDS) to publish to the gateway agent (SROS2-enforced).

D. Recommended Enhancements

For stronger assurances, enable at-rest encryption on the gateway (for example, encrypted volumes), rotate keys periodically with certificate revocation for compromised nodes, and extend audit retention with role-based access control (RBAC) reviews. Quantifying the performance overhead of these controls is left to future work.

Fig. 6 summarizes life-cycle controls.

E. Compliance Notes

Our handling aligns with HIPAA/GDPR principles—data minimization, purpose limitation, security of processing, and accountability. Full compliance requires organization-specific policies (for example, consent and access requests), public-key infrastructure for credential lifecycle, and periodic RBAC reviews beyond the scope of this prototype.

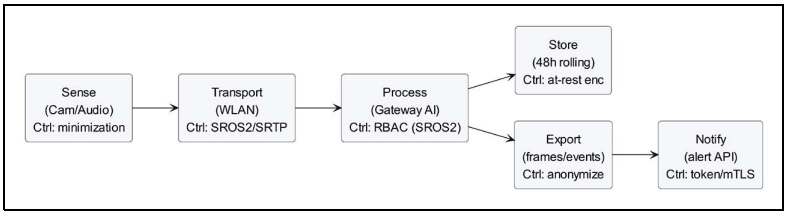


FIG. 6. Life-cycle privacy and security controls.

VII. Conclusion

This article demonstrates how a DDS-based AIoT gateway can efficiently manage real-time multimodal sensing for smart eldercare environments. By combining Micro-ROS, automated device onboarding, and edge AI processing using YOLOv8 and YAMNet, the system supports scalable, low-latency integration of video and audio streams. Our unified development template enabled plug-and-play deployment of ESP32-S3 sensor nodes, greatly simplifying configuration.

Experimental results highlight DDS's superior performance over MQTT in high-throughput scenarios, sustaining higher frame rates and lower transmission delays. The proposed architecture not only improves runtime efficiency but also reduces engineering complexity in managing heterogeneous IoT nodes. Limitations: our evaluation used a 2.4 GHz WLAN, a small number of devices, and short trial windows; we prioritized throughput and packet rate, with latency/jitter distributions left for follow-up. These results validate a practical and extensible approach for deploying intelligent, responsive edge systems in real-world healthcare applications.

References

[1] A. Koohang, C. S. Sargent, J. H. Nord, and J. Paliszkiewicz, "Internet of things (IoT): From awareness to continued use," *Int. J. Inf. Manage.*, vol. 62, Apr. 2022, Art. no. 102442.
[2] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol," in *Proc. Int. Conf. Eng. MIS (ICEMIS)*, Monastir, Tunisia, 2017, pp. 1–6.
[3] D. Sacramento, G. Schütz, and N. Correia, "Aggregation and scheduling in CoAP/Observe based wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, UK, 2015, pp. 654–660.
[4] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, and B. Stiller, "Landscape of IoT security," *Comput. Sci. Rev.*, vol. 44, Apr. 2022, Art. no. 100467.
[5] W. Yu et al., "A survey on the edge computing for the internet of things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.
[6] G. Pardo-Castellote, "OMG data-distribution service: Architectural overview," in *Proc. 23rd Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW'03)*, Providence, RI, USA, 2003, pp. 1–7.
[7] L. J. Dust, E. Persson, M. Ekström, S. Mubeen, C. Seceleanu, and R. Gu, "Experimental evaluation of callback behavior in ROS 2 executors," in *Proc. IEEE 28th Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, 2023, pp. 1–8.
[8] R. Carreira, N. Costa, J. Ramos, L. Frazão, and A. Pereira, "A ROS2-based gateway for modular hardware usage in heterogeneous environments," *Sensors*, vol. 24, no. 19, p. 7, Sep. 2024.
[9] A. Nugur, M. Pipattanasomporn, M. Kuzlu, and S. Rahman, "Design and development of an IoT gateway for smart building applications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9020–9029, Oct. 2019.
[10] C.-H. Chen, M.-Y. Lin, and C.-C. Liu, "Edge computing gateway of the industrial internet of things using multiple collaborative microcontrollers," *IEEE Netw.*, vol. 32, no. 1, pp. 24–32, Jan. 2018.
[11] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.
[12] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.
[13] A. Barciś, M. Barciś, and C. Bettstetter, "Robots that sync and swarm: A proof of concept in ROS 2," in *Proc. Int. Symp.*

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Multi-Robot Multi-Agent Syst. (MRS), New Brunswick, NJ, USA, 2019, pp. 98–104.

[14] J.-B. Castillo-Sánchez, E. González-Parada, and J.-M. Cano-García, "A novel testbed for evaluating ROS 2 robot swarm wireless communications," in *Proc. IEEE 22nd Mediterr. Electrotechnical Conf. (MELECON)*, Porto, Portugal, 2024, pp. 68–73.

Biographies

Ching-Han Chen received the Ph.D. degree in electrical engineering from the National Tsing Hua University, Taiwan, in 2014. He is currently an Associate Professor with the National Central University. His research interests include intelligent control systems, autonomous robots, and AI applications in control engineering.

Chih-Hao Chang received the B.S. and M.S. degrees from the National Quemoy University, Taiwan, in 2019 and 2021, respectively. He is currently working toward the Ph.D. degree with the National Central University. His research interests include IoT, autonomous control, and industrial protocols for smart manufacturing.

Chih-Wei Peng received the master's degree in computer science from the National Central University, Taiwan, in 2025. His research interests include embedded systems, IoT, and edge computing, especially in drone control applications.

Guan-Wei Lan received the M.S. degree in CSIE in 2013 from the National Central University, where he is currently working toward the Ph.D. degree. His research interests include embedded systems, robotics, and the integration of AI in smart manufacturing.