Journal Pre-proof

Edge robotics: are we ready? An experimental evaluation of current vision and future directions

Milan Groshev, Gabriele Baldoni, Luca Cominardi, Antonio de la Oliva, Robert Gazda

PII: S2352-8648(22)00088-8

DOI: https://doi.org/10.1016/j.dcan.2022.04.032

Reference: DCAN 421

To appear in: *Digital Communications and Networks*

Received Date: 19 November 2020

Revised Date: 21 April 2022

Accepted Date: 28 April 2022

Please cite this article as: M. Groshev, G. Baldoni, L. Cominardi, A.d.l. Oliva, R. Gazda, Edge robotics: are we ready? An experimental evaluation of current vision and future directions, *Digital Communications and Networks* (2022), doi: https://doi.org/10.1016/j.dcan.2022.04.032.

Digital Communications and Networks(DCN)

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/dcan

Edge robotics: are we ready? An experimental evaluation of current vision and future directions

Milan Groshev*[a], Gabriele Baldoni[b], Luca Cominardi[b], Antonio de la Oliva[a], Robert Gazda[c]

[a]**University Carlos III of Madrid, Leganes 28028, Madrid**
[b]**ZettaScale Technology SARL, Saint-Aubin 91190, France**
[c]**InterDigital Inc., Wilmington DE 19809, USA**

Abstract

Cloud-based robotics systems leverage a wide range of Information Technologies (IT) to offer tangible benefits like cost reduction, powerful computational capabilities, data offloading, etc. However, the centralized nature of cloud computing is not well-suited for a multitude of Operational Technologies (OT) nowadays used in robotics systems that require strict real-time guarantees and security. Edge computing and fog computing are complementary approaches that aim at mitigating some of these challenges by providing computing capabilities closer to the users. The goal of this work is hence threefold: *i)* to analyze the current edge computing and fog computing landscape in the context of robotics systems, *ii)* to experimentally evaluate an end-to-end robotics system based on solutions proposed in the literature, and *iii)* to experimentally identify current benefits and open challenges of edge computing and fog computing. Results show that, in the case of an exemplary delivery application comprising two mobile robots, the robot coordination and range can be improved by consuming real-time radio information available at the edge. However, our evaluation highlights that the existing software, wireless and virtualization technologies still require substantial evolution to fully support edge-based robotics systems.

KEYWORDS:
Edge, Robotics, Experimental, Fog, MEC, Virtualization, ROS, Fog05

1. Introduction

Historically, Information Technology (IT) and Operational Technology (OT) have been two separate domains addressing distinct scenarios. While the former focuses on providing services in a cyber-only environment (e.g., content-delivery networks, video-on-demand), the latter targets applications in a cyber-physical environment (e.g., robotics systems, industrial automation) where the virtual and the real worlds are closely entangled [1]. With these different visions in mind, it is understandable that IT and OT have evolved in the past in different, yet related, directions.

As of today, IT is primarily characterized by large-scale and multi-tenant deployments (e.g., datacenters), short lifetime-cycles of homogeneous resources (e.g., servers, mobile phones), and the usage of best-effort technologies (e.g., Ethernet, IP) coupled with high-availability techniques (e.g., load-balancing, replicas) [2]. In contrast, OT is heavily characterized by confined and well-controlled environments (e.g., manufacturing plants), long lifetime-cycles of heterogeneous resources (e.g., assembly line), hard-real time guarantees (e.g., closed control loop), ad-hoc components (e.g., microcontrollers) and

*Milan Groshev (Corresponding author) (email: milan.groshev@uc3m.es).
[1]Gabriele Baldoni (email: gabriele@zettascale.tech).
[2]Luca Cominardi (email: luca@zettascale.tech).
[3]Antonio de la Oliva (email: aoliva@it.uc3m.es).
[4]Robert Gazda (email: robert.gazda@interdigital.com).

reliable technologies (e.g., Profibus, CAN bus) [3]. In recent years however, the OT has started to move towards a more IT-oriented approach driven by the proven benefits of cloud computing in terms of flexibility and cost reduction.

One clear example is cloud robotics [4], which aims to integrate cloud computing resources in the robotics systems so as to increase the re-configurability as well as to decrease the complexity and cost of the robots. Moreover, by offloading the control logic from the robot to the cloud, it is easier to share services and information from various robots or agents to achieve better cooperation and coordination. However, the centralized nature of cloud computing poses critical challenges: Cloud facilities usually reside far away from the robots (i.e., separated by one or more uncontrolled transit networks), making it hard to achieve high-bandwidth, low-latency and bounded jitter [5].

In recent years, edge [6] computing and fog [7] computing have emerged as paradigms to alleviate these problems by placing computing and storage resources deep into the network. This enables applications to execute closer to the users (i.e., the robots), resulting in a more predictable communication and overall better system performance. Moreover, real-time information about user connectivity is expected to be available at the edge, enabling the dynamic adaptation of the application's logic to the actual status of the communication (e.g., radio channel) [8]. As a consequence, edge computing and fog computing are very well-positioned for overcoming today's cloud robotics challenges. In light of this, the main contributions of this work are:

1. We analyze the current landscape of edge computing and fog computing in the context of robotics systems, with a particular focus on architectures from standards and industrial committees.
2. We design an exemplary end-to-end robotics system, for the edge environment where context radio information is available.
3. We implement an exemplary robotics application, envisaging the coordinated movement of two robots to experimentally assess the suitability of the IT and OT technologies available today with regard to edge and fog architectures.
4. We formulate the main challenges and gaps that we have identified in integrating IT and OT technologies and we delineate some future directions towards IT and OT harmonization.

The rest of the paper is organized as follows. Section 2 reviews the related work and discusses the benefits of integrating edge computing and fog computing for robotics systems. In Section 3, the edge-based reference design for our robotic application is presented. In Section 4, prototype implementation of the designed robotic application is described. Results based on experiments are presented and analyzed in Section 5. Section 6 discusses the potential research challenges and future research directions. Finally, Section 7 summarizes our findings and draws the conclusions.

2. Related work

Moving resource demanding computation (used to build specific applications) to the cloud makes traditional robotics systems more service-oriented, interoperable, distributed and programmable. However, such cloud offloading implies problems regarding availability [36], low latency [37], bounded jitter, high bandwidth [38], [39], security and data filtering [40], which limit the use of cloud robotics for applications that require real-time sensitivity and precision. Edge computing and fog computing with their implementations have emerged to fill this gap by placing computational resources closer to the user. While European Telecommunications Standards Institute (ETSI) provides an implementation for edge computing through a framework for Multi-access Edge Computing (MEC) [41] over static substrates (e.g., data centers or servers), the OpenFog [42] working group of the Industrial Internet Consortium (IIC) [43] extends this definition to include less powerful, constrained and mobile computational substrates, including the end-user devices. ETSI MEC and IIC with their main features have the potential to provide major benefits for robotics systems. These benefits, together with some recent work for efficient integration of edge computing and fog computing into robotics systems, are summarized in Table 1. It is worth mentioning that some of the existing studies do not specifically mention the use of ICC or MEC for robotics systems, but their work complies with the corresponding reference architectures.

Existing studies on using MEC and IIC for robotics systems mainly focus on the low-latency and computation offloading features that offer the potential to overcome the real-time constraints of cloud based solutions. By placing the time-sensitive robotics applications at the edge of the network, MEC and IIC can ensure high computation capabilities while leveraging low-latency communication and high bandwidth. Most of the recent experimental work focuses on distributing computation between the robots and the edge of the network, in particular, deep object recognition and grasp planning [16], visual odometry [17], [25], real-time control [18], [27], [28], [19], [21] and object detection [26], [20]. Besides the experimental work, several studies have motivated the application of MEC and IIC for different robotics systems by proposing collaborative architectures (e.g., healthcare [9], autonomy [10], tele-surgery [11], manufacturing [12–14] and multi robot systems [24]). The MEC and ICC enabled implementations allow the co-location of independent applications on a shared edge/fog node through virtualized abstraction. As illustrated in Ta-

Table 1: Related works targeting ETSI MEC and IIC OpenFog architectures in the context of robotics systems categorized by different Edge-related characteristics.

Characteristic	ETSI MEC		IIC OpenFog	
	Architecture	Experimental	Architecture	Experimental
Low-latency	[9] [10] [11] [12] [13] [14] [15]	[16] [17] [18] [19] [20] [21]	[12] [14] [22] [23] [24]	[16] [25] [26] [27] [28] [29]
Computation offload	[10] [11] [12] [13] [14]	[16] [17] [18] [30] [20] [21]	[12] [14] [23] [24]	[16] [25] [26] [27] [28] [29] [30]
Context awareness	[11] [15]	[18]	-	-
Localization	[31]	[17] [30] [31]	-	[25] [29] [30]
Efficiency	[9] [10] [11] [12] [13] [14]	[16] [18] [27] [32] [19] [21]	[12] [14] [23] [24]	[16] [28] [29]
Multi-RAT	[12] [33]	[32]	[12] [23] [24] [33]	-
Proximity	[12] [13] [14]	[34]	[12] [14] [22] [24]	[34] [35]

ble 1, some of the existing studies where the main focus is on the low-latency and computation offload [9–14], [16], [18], [19], [21], [23], [24], [27–29] also adapt the concept of virtualization that allows robotic services to reuse the surrounding hardware and deploy applications on demand.

The close proximity of MEC and IIC has been studied in some recent existing experimental studies to offload the location-based robotics services from mobile robots with limited computational and low-energy resources [17], [25], [29–31]. Additionally, [12–14], [22], [24] elaborate on the reduced network pressure and improved security and privacy of the robot sensor data that can be offered by restricting the access within a trusted private infrastructure. For security, cloud-fog-edge security risk prediction method has been proposed in [34] to meet the current needs of the industrial Internet systems. In [35], the authors propose a fog-driven method to detect GPS spoofing of unmanned aerial vehicles.

IIC and MEC have the potential to enable robots to communicate directly in a Device-to-Device (D2D) fashion through a network-assisted approach [44]. Tight coordination and cooperation between different Radio Access Technologies (RATs) in the edge can be used to enable multi-RAT communication in robotics systems. Moreover, the network under these two implementations is expected to be completely context-aware [45], allowing robotics applications to obtain real-time information for the radio network condition. Based on this information, robotics applications can adapt their operations or optimize the network to improve the user experience. However, as Table 1 shows, most of the recent studies that address multi-RAT communication [12], [33], [23], [24] and network contextual information applicability to robotics systems [11], [15] focuses on proposing IIC and MEC compliant communication architectures. For what concerns experimental studies, in our previous work [18], we used the locally available context information to adapt the driving speed of a mobile robot. In [32], the end-to-end reliability of mission-critical traffic was investigated in softwarized 5G networks where a multi-RAT and multi connectivity access network is considered.

Relation with our work: With the growing need for context-aware real-time collaboration, especially under mission-critical scenarios in robotics systems, it is strongly desirable to experimentally evaluate the implementations of such context-aware platforms (MEC and ICC) and the suitability of the currently available IT/OT technologies in order to truly materialize edge computing and fog computing visions. However, as mentioned above, few existing studies adapting network context information focused on improving the overall performance of robotics systems. Therefore, this paper designs and experimentally validates an edge robotics application that leverages on real-time context-aware collaboration to optimize the end-to-end robotics systems.

3. An end-to-end edge robotics system design

Based on the state-of-the-art overview performed in Sec. 2, in this section we propose the design of an exemplary edge robotics system. Our design uses the ETSI MEC definition of applications and context-aware services as a set of autonomous and virtualized microservices [46] that can be distributed and orchestrated in the system. However, the applicability of MEC specifications to support end terminals (UEs, robots) in the infrastructure is not considered by ETSI. As Fig. 1 illustrates, we also adapt the IIC concept of end terminals as part of the infrastructure. We believe that the integration of IIC and MEC can fully enable a distributed end-to-end edge robotics system.

3.1. System structure

The proposed edge robotics system illustrated in Fig. 1 consists of 3 subsystems: *i)* robotics system (blue modules), *ii)* edge computing system (red modules), and *iii)* orchestration system (green modules). In general, raw sensor data is acquired by robots and

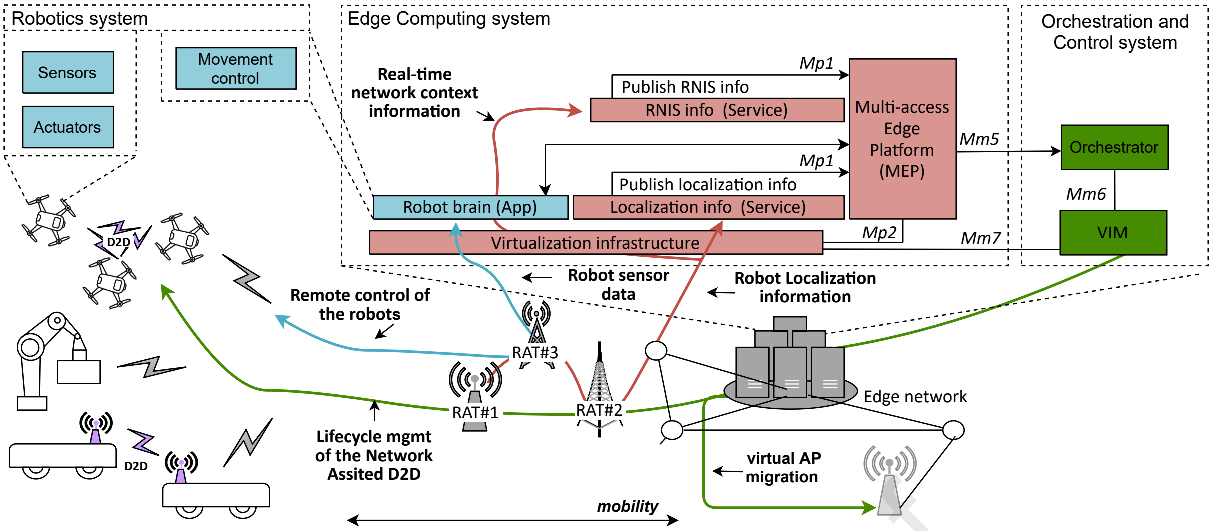


Fig. 1: End-to-end Edge robotics scenario

sent to the edge computing system, removing the need for any data processing by the robots. Real-time navigation decisions are taken in the edge computing system to minimize the latency and share the data between different connected robots. The orchestration system provides additional services such as managing and controlling the underlying infrastructure.

3.1.1. Robotics system

The robotics system (Fig. 1 - blue modules) is composed of software modules distributed between the robots and the edge computing system. In our proposed design, the robots act only as sensors and actuators. They have embedded within them only the minimal components in charge of: *i)* reading data from the sensors (e.g., odometry, camera, lidar) and sending them to the robot brain; and *ii)* executing navigation instructions received from the robot brain. The robot brain resides in the edge computing system and is responsible for the coordination and navigation of robots. Control algorithms are run in the robot brain, where real-time network contextual information and localization data are used to adapt robot operations. The robots can be equipped with the most suitable RAT (e.g., WiFi, LTE, 5G) based on the use case requirements. Additionally, Device-to-Device (D2D) connectivity is envisioned between the robots to offer higher data rates, lower transfer delays, and improve power efficiency [47].

3.1.2. edge computing system

The edge computing system shown in Fig. 1 (red modules) depicts the integration of applications, together with different services such as radio-network information and location services. Both services are described in more details later on. In particular, the edge computing system takes care of coordination and control of robot movements based on network contextual information and location algorithms. The virtualization infrastructure provides computation, storage, and network resources for the applications. It also includes the basic networking capabilities for routing the traffic between services, applications, external networks, and multi-access edge platform. The multi-access edge platform offers an API-based domain for advertising, discovering and consuming services. We can consider the multi-access edge platform as a 'middleman' in charge of storing and distributing the subscription data of a service to the data subscribers via a suitable messaging protocol (e.g., MQTT [48], REST [49], DDS [50], Zenoh[5]).

Localization algorithms are run in the edge computing system, where real-time sensor data is matched with an area of an existing map. Additionally, for situations when the robot operates in partially or totally unknown environments, the signaling information received from the robot can be utilized to estimate its precise location. Simultaneous localization and mapping algorithms run as services. The real-time localization information is shared with the robot brain using the multi-access edge platform.

The Radio-Network Information Service (RNIS) provides information regarding the quality of the radio channel. Regardless of the radio access technology (e.g., LTE, WiFi, Bluetooth, etc.), the RNIS monitors the connectivity and publishes real-time information about the signal strength, MAC layer parameters, packet loss, etc., of each robot. The use of RNIS service can facilitate the robot's mobility in an indoor environment and optimize the robot's navigation. For example, based on the signal level and the MAC layer parameters, the robot brain can detect quality degradation on the wireless channel and adapt the robot speed accordingly.

[5]Zenoh project: http://zenoh.io/ [Accessed: 20 April 2022]

3.1.3. Orchestration system

The orchestration and control system (Fig. 1 - green modules) is in charge of the application life-cycle management. It includes allocation, monitoring and enforcement, over the IT/OT convergent infrastructure while ensuring dynamism and elasticity for the robotics application. Its responsibilities are separated in two different modules: *i)* the Virtual Infrastructure Manager (VIM); and *ii)* orchestrator.

On the one hand, the VIM interacts with the infrastructure and offers unified abstractions over the heterogeneous, mobile and volatile resources. It offers monitoring, allocation and management of resources across the infrastructure and shares this information with the upper layers.

On the other hand, the orchestrator is responsible for: *i)* storing the catalog information of the robotics application components, including checking the integrity and authenticity of the application packages; *ii)* keeping records of the on-board components and enforcing the allocation; and *iii)* re-allocating components based on the information received from the robot and the edge computing system as well as from the VIM. Compared with the cloud-like orchestrators, more advanced placement algorithms are required in order to select the appropriate host(s) for instantiating or migrating the components based on constraints defined by the application, such as latency, computing requirements, availability of resources and services etc. Enforcement instructions are transmitted to the robots and the underlying infrastructure through the VIM.

3.2. Connectivity challenges

In our system design, the radio connectivity is used to form a closed control-loop between the robots and the robot brain. The quality of the radio channel can be reduced by Radio-Frequency (RF) interference [51] inducing deep fading, resulting in increased jitter, low throughput and high losses. As a result, we can experience a significant degradation of the smoothness and precision of our robotics system. For example, delayed packets in the robot brain can result in robot coordination imprecision, while delayed packets in the robot will be translated to navigation smoothness degradation.

To tackle this problem, we can use the real-time RNIS information to design algorithms that are able to detect or predict the RF interference. Based on this information, the orchestrator can establish network-assisted D2D communication between the robots to reduce the latency to the point where precise robot coordinated movements are achievable. As shown in Fig. 1, the D2D communication channel can be used on-demand for exchanging synchronization related robot information.

Furthermore, radio access technologies (e.g., WiFi, ZigBee, Bluetooth) in indoor environments have a limited radio coverage range. As some mobile robots target large indoor environments, a wireless range of 10 to 500 meters with a single access point can be a limiting factor, notably for the 5GHz and above frequency bands. Additionally, considering the low latency requirements of our robotics system, an advanced handover mechanism will be needed for continuous mobility without an interruption of the robotics service. This will require a dedicated robot architecture design and deployment.

To address this challenge, we can exploit the virtualization infrastructure that runs on top of our distributed system and instantiate a dedicated virtual Access Point (vAP) infrastructure. With the help of the locally available RNIS and location services, it is possible to detect when a mobile robot is moving out of the coverage area. Placement algorithms use these services to select the best location to offload the access point capabilities in order to extend the driving range of the robots. Moreover, in the absence of network activity, the allocated edge resources can be released to save energy and computational budget.

4. Edge-based robotics application for mobile robots

To address the aforementioned considerations and evaluate the available IT/OT technologies, we have built an experimental testbed along a hallway in Universidad Carlos III de Madrid and deployed a prototype of the system presented in Sec. 3.

The goal of the experimental testbed is: *i)* to use the contextual information to extend the driving range of mobile robots; and *ii)* to evaluate the network-assisted D2D communication to improve the mobile robot's coordination. We also successfully demonstrated a similar edge robotics system for remote autonomous navigation in Hsinchu, Taiwan.[6]

4.1. Mobile robots system implementation

Our robotics application is motivated by a realistic use case where a fleet of mobile robots are remotely controlled and coordinated to perform different tasks in a multi-access indoor environment. This use case allows us to easily detect the new business actors that are enabled by using an end-to-end edge robotics system. Let's consider a business center as a potential deployment scenario. In this case, the infrastructure (e.g., the micro-datacenter at the edge) may be owned by the business center owner, or even a third party that provides and manages the infrastructure for the business center owner. The clients of the business center require a robotics service, such as video surveillance, and cleaning or transport of goods. This service is provided by a robotics service provider that delivers its

[6] A demonstration video is available at: https://youtu.be/zzjxDSLGdas [Accessed: 20 April 2022].

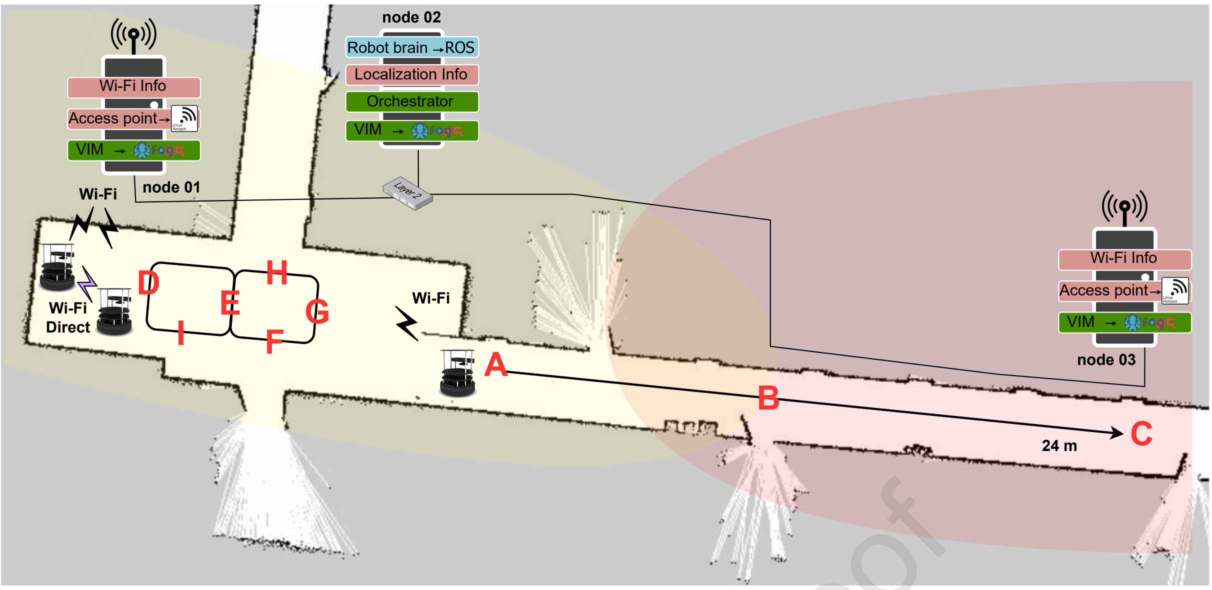


Fig. 2: Grid map of the deployed experimental testbed obtained using lidar and odmoetry data. In the map two separate experimental scenarios are presented, namely (A,B,C) and (D,E,F,G,H,I). The WiFi coverage in the hallway is denoted with yellow and red ovals

applications through the infrastructure located in the business center.

The testbed presented in Fig. 2 is comprised of: *i)* three mini PCs with 4 vCPUs and 16 GB of RAM, namely, node 01, node 02 and node 03, interconnected by a 10 GB/s ethernet connection; and *ii)* two mobile robots. Two of the mini PCs (node 01 and node 03 in Fig. 2) are equipped with IEEE 802.11ac capable interfaces. As shown in Fig. 2, these two mini PCs are placed in two different locations in order to ensure WiFi coverage of the testbed area.

For the mobile robots, we used the Kobuki Turtlebot S2 robots equipped with 2 WiFi antennas and a laptop with 8 GB of RAM and 5 vCPUs. One antenna is used for the robot-to-infrastructure communication and one for the robot-to-robot communication. Additionally, a RPLIDAR A2[7] lidar is mounted on the robots in order to perform a 360-degree omnidirectional laser range scanning. The Turtlebot maximum speed is 0.75 m/s, while its minimum speed is 0.1 m/s with a ROS control frequency of 10 Hz (i.e., 100 ms). The sampling frequency for reading the odometry sensor data from the robot's wheels is 16.6 Hz (i.e., 60 ms) and the rotating frequency of the lidar is 10 Hz with a guaranteed 8-meter ranger distance.

We deployed a version of the edge robotics system shown in Fig. 1 on this testbed. The robotics system is based on Robot Operating System version 1 (ROS-1)[8] and is distributed across the robot's and the edge computing system. Basic robot sensors (lidar, odometry) and actuators handling are provided by the ROS-1 nodes deployed in the robot's laptops. The robot brain is placed in node 02 and hosts the ROS-1 navigation stacks for both robots together with the static map of the covered testbed indoor environment.

The robots are connected to the infrastructure through a WiFi link. We decided to use IEEE 802.11 technology mainly because of the high bandwidth, low latency and widespread use that make it suitable for the requirements of our time-sensitive robotics application. An application implements the WiFi access point capabilities and can be deployed on-demand as a Hostapd[9] Linux container. It is important to mention that our vAP instances are configured to use IEEE 802.11r [52] over the Distributed System (DS) in order to permit fast and secure handovers. Since we use WiFi as our RAT, a RNIS service, namely, WiFi information service, is deployed as a container together with the WiFi access point and provides real-time context information about the connected robots. In addition, a location service is hosted by node 02. This service is implemented as a ROS node and provides probabilistic robot localization based on the lidar sensor. Such location and context information is therefore published via an MQTT broker that acts as a multi-access edge platform and is hosted by node 01.

Regarding the orchestration system, we implemented a custom orchestrator deployed as a container in node 02. The orchestrator implements the base life-cycle management functionalities such as instantiation and termination of the application. More details regarding the functionalities of the implemented orchestrator are presented in Sec. 5. For the VIM component, we employed the open-source project

[7] https://www.slamtec.com/en/Lidar/A2 [Accessed: 20 April 2022]

[8] Widespread framework for developing and testing multi-vendor robotics software.

[9] User space software that provides access point capabilities.

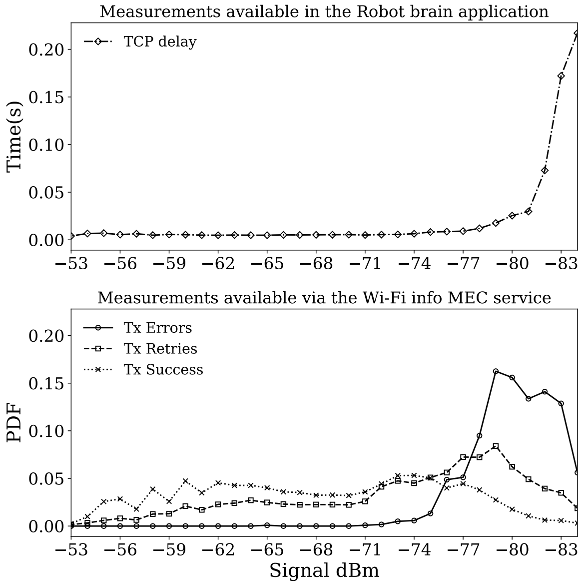


Fig. 3: Signal and delay behaviour [18]

Eclipse Fog05[10] [53], which embodies the principles described in Sec. 3. Eclipse Fog05 addresses the requirements and characteristics of edge computing by providing a decentralized infrastructure that logically unifies computing, networking, and storage fabrics end-to-end, while addressing the challenges imposed by resource heterogeneity (e.g., virtual machines, containers, native applications).

5. Experimental evaluation

This section describes the experimental validation of the edge robotics system proposed in Sec. 3. To that end, we implemented a prototype of our edge robotics system as described in Sec. 4.1. This testbed is used for a step by step experimentation aimed at evaluating the context-aware vAP offloading (Sec. 5.2) and the network-assisted D2D robot coordination (Sec. 5.3). Additionally, in Sec. 5.1, we analyse the WiFi context information available for our experimental area.

5.1. Wireless channel behaviour

In our previous work [18], in order to better understand the system and its limitations, we took into consideration how the WiFi context information impacts the delay in controlling the robot, as perceived by the robot brain. This consideration is also valid for our experimental evaluation because we used the same experimental area and hardware as described in Sec. 4. The university hallway layout is provided in Fig. 2. The robot was positioned at location A throughout the experiment and the robot brain drove the robot on a straight line from A to B. The length of the hallway AB is 15 m. The robot is connected to a vAP deployed in node 01. At the end of the driving, the robot stops approximately 22 m from the vAP. It's important to mention that the robot starting position is approximately 7 m away from the vAP. The WiFi information obtained via the WiFi Info service was recorded in the robot brain, while on the robot itself we measured the ROS-1 navigation delay.

Fig. 3 depicts the downlink retransmissions (Tx Retries), the failed transmissions (Tx Errors), downlink successful transmissions (Tx Success) and ROS-1 downstream delay (TCP delay) in our experimental area. The Probability Density Function (PDF) in Fig. 3 shows that a low WiFi signal level (below -71 dBm), results in a higher probability of a failed transmission. This probability surpasses the probability of successful transmission at signal levels lower than -77 dBm. TCP delay measurements confirm this with values as high as hundreds of milliseconds. For signal levels below -80 dBm (the last 2 meters of the drive), it is very difficult to have a successful transmission, which results in non-smooth and bouncy movements of the robot. This WiFi signal level can be considered as a borderline between a good and bad coverage area for our deployed edge robotics system.

5.2. Context-aware virtual access point offloading

5.2.1. Algorithm design

Algorithm 1 Context-aware virtual access point offloading algorithm

info ← *GetCurrentWiFiInfo()*
buffer ← *buffer.removeOldestWiFiInfo()*
buffer ← *buffer.add(info)*
signalLevel ← *buffer.average()*
if *signalLevel* ≤ -65 dBm **then**
    if *signalLevel* ≥ -69 dBm **then**
        currentAP ← *info.GetAPinfo()*
        newAP ← *GetBestAvalaibleAP()*
        deployNewAP(newAP)
        if *newAP.ssid==avalaible* **then**
            RobotHandover(newAP)
            StopOldAP(currentAP
        end if
    end if
end if

Based on the conclusions drawn from Sec. 5.1, we present the design of a context-aware vAP offloading algorithm. The aim of this algorithm is to provide an uninterrupted robot delivery service while extending the network WiFi coverage in indoor environments. Through this algorithm, we showcase the advantages of consuming context information for managing vAP. Nonetheless, we acknowledge that more complex and optimal algorithms than the one proposed in this section can be designed.

Algorithm 1 shows the pseudo-code of the vAP offloading algorithm. The orchestrator, in real-time, extracts the current signal level from the WiFi info ser-

[10]Fog05 project: https://fog05.io/ [Accessed: 20 April 2022]

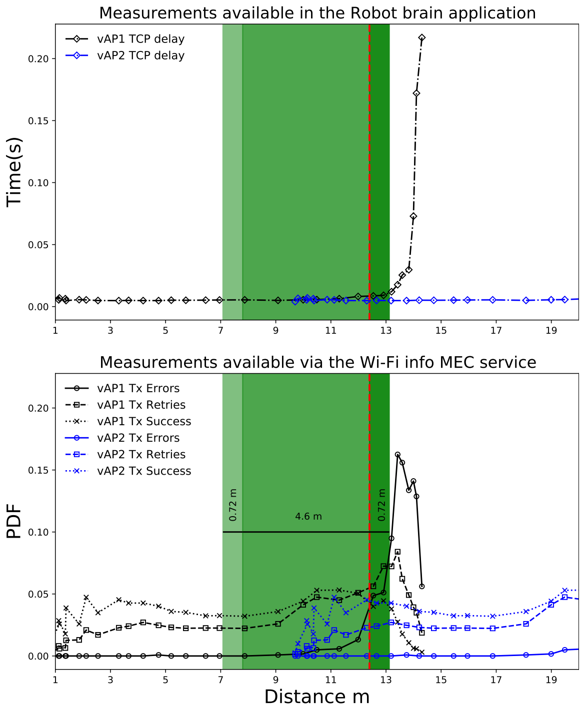


Fig. 4: Offloading of virtual Access Point

vice, stores it in a circular buffer and computes the moving average of the WiFi signal levels. In Sec. 5.1 we observed that our edge robotics system has significant degradation in performance for signal values below -80 dBm. In order to avoid these values, the offloading algorithm must react proactively when deploying a new vAP. Based on this assumption, the offloading algorithm obtains information about the new AP and performs the deployment for an average signal level in the interval between -65 dBm and -69 dBm. Once the new vAP is up and running, the orchestrator triggers the robot handover from the currently associated to the newly instantiated vAP. In the end, the orchestrator stops the old vAP and releases the resources on the corresponding node. It is important to indicate that the vAP instances need to be configured to use IEEE 802.11r over DS so as to perform fast and secure handovers.

5.2.2. *Experimental results*

The testing of the proposed context-aware algorithm was accomplished in the university hallway shown in Fig. 2. The robot was positioned at location A and connected to the vAP instantiated in node 01. The WiFi range of this AP is presented in Fig. 2 with yellow. The robot, controlled by the robot brain, accelerates from position A to the target velocity of 0.2 m/s. During the drive, the orchestrator implements the vAP offloading algorithm and performs the vAP offloading from node 01 to node 03. After having traveled for 24 m, the robot stops at position C. In the orchestrator, we recorded the total vAP offloading time, which is the time elapsed from when it is detected that vAP offloading is needed until the old vAP is stopped. Moreover, to reduce the deployment time, we already made a copy of the vAP container image available in the nodes. By doing this, the orchestrator does not need to copy the vAP over the network. In a separate laptop placed in the corridor, we recorded the IEEE 802.11 link layer overall downtime.

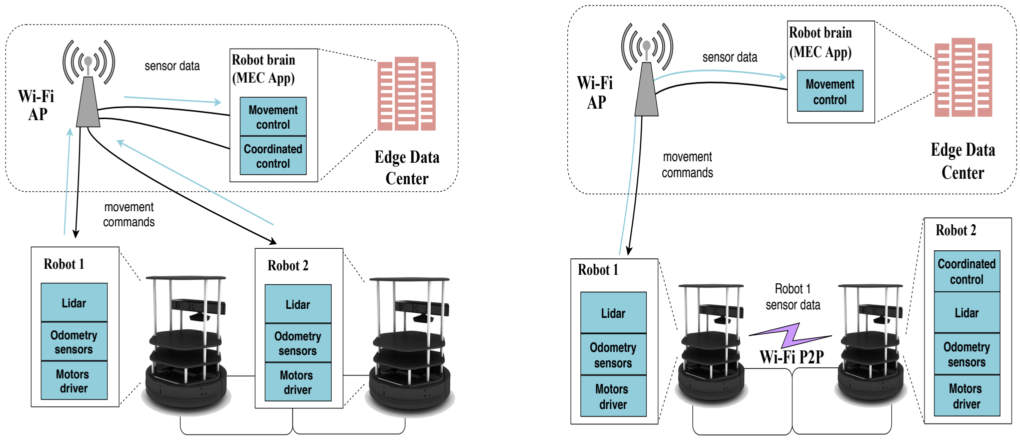
Fig. 4 shows the available WiFi context information from the departing vAP (in black) and the arriving vAP (in blue). The breakdown of the total offloading distance is reported in green. From left to right, Fig. 4 depicts the traversed robot distance during the Eclipse Fog05 Linux container vAP instantiation. Next, the distance traveled by the robot while the vAP is been provisioned (the moment when the SSID is available) and the overall handover distance is shown. Finally, the results for the robot's traveled distance during the Eclipse Fog05 Linux container vAP termination are presented.

Keeping in mind that the vAP function is deployed as a Hostapd Linux container, the results show that the robot's traveled distance is approximately 5.5 m during the vAP offloading procedure if driving at 0.2 m/s. Three factors contribute to this traveled distance: *i)* approximately 0.7 m of the robot's traveled distance is for Eclipse Fog05 to deploy the vAP Linux container; *ii)* approximately 4.6 m of the robot's traveled distance is for the Hostapd Linux container to boot and the SSID to become available; and *iii)* approximately 0.7 m of the robot's traveled distance for Eclipse Fog05 to stop the vAP Linux container. During our experiments, we recorded a value of 38 ms, as the time interval in which our robots are without WiFi connectivity (link-layer handover downtime).

5.3. *Network-assisted D2D robot coordination*

One of the problems detected while coordination is performed between the robots is the jitter introduced by the WiFi link. Sometimes a coordinated order given to both robots, in separate messages addressed to each of the robots, results in different execution times, leading to the dis-coordination of the movement. To reduce this effect, our edge robotics system implements D2D communication to improve the reliability. Network-assisted D2D communication testing was performed in the university hallway. The floor layout is provided in Fig. 2. The robots were positioned at location D, one behind the other with an approximate distance of 0.3 m between them. The first robot starts the experiment drive at a constant speed of 0.2 m/s. The second robot follows the first robot, trying to keep a constant distance of 0.65 m. The robots move from D to E, E to F, F to G, G to H, H to E and E to I. The length of the hallway DFHI is 5 m and the EG is 4m.

Fig. 5 depicts two separate experiment scenarios. Both experiments are composed of ROS-1 movement and coordinated control applications. The movement control application navigates Robot 1 through the known map. The coordinated control navigates Robot



(a) Centralized robotic control

(b) Network-assisted D2D robotic control

Fig. 5: Network-assisted D2D experiment scenarios

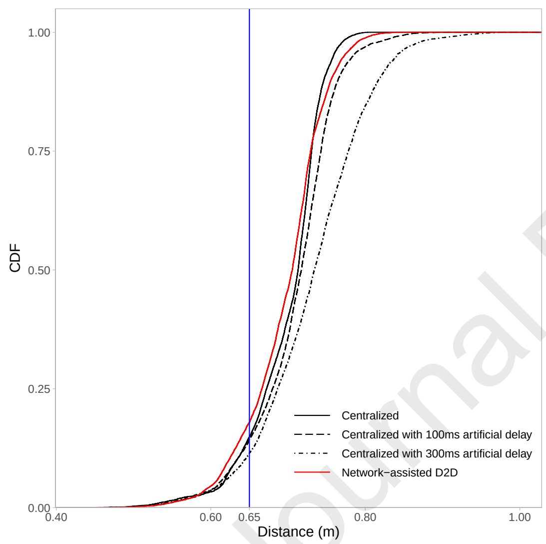


Fig. 6: eCDF of distance between the two robots

2 by following the driving path of Robot 1 while maintaining a constant distance. In the centralized robotic control (see Fig. 5a), the robot brain hosts both applications. The sensor data received from the robots (odometry, lidar) is used to calculate and execute the driving instructions for the navigation of both robots. In the network-assisted D2D robotic control, we have D2D communication between the two robots based on WiFi Direct. The coordinated control application is now placed in Robot 2. Consequently, Robot 2 is now navigated by the coordinated control app that consumes the sensor data of Robot 1 via the D2D communication channel. In order to emulate the effects of network interference, we introduced an artificial delay of 100ms and 300ms on the WiFi link in the centralized robotic control. The selected values for the artificial delay are influenced by our robot sensor sampling frequency and the control loop described in Sec. 4.

5.3.1. Experimental results

Fig. 6 depicts the Euclidean distance between the robots. It is worth mentioning that, in our tests, the robots are making turns. Since we are measuring the straight line distance, this leads to shorter distances in our measurement set. The Cumulative Density Functions (CDF) shows that, by using the D2D communication channel, we can arrive closer to our target distance of 0.65 m during the experimental drive. This is because in the centralized robotic control experiments, when we introduce an artificial delay, a robot location mismatch is triggered in the robot brain. The location mismatch then results in decreased precision when the robot brain tries to maintain a constant distance between the robots. Naturally, as shown in Fig. 6, the Euclidean distance between the robots increases as we increase the artificial delay. However, Fig. 6 also shows that the Euclidean distance between the two robots in the non-artificially delayed centralized control is very close to the network-assisted D2D. This is reasonable since we used WiFi as radio access technology for both experimental scenarios.

6. Discussion and future directions

Although the edge and fog ecosystem can bring advantages to the real-time robotics system by integrating IT and OT technologies, there are inherent gaps and challenges that are yet to be adequately addressed. This section presents the main issues that we have identified and recommends some future trends towards IT and OT convergence.

6.1. Operational technologies

Offloading the computation tasks from the robots to the edge of the network requires a distributed, scalable and fast software framework. One of the open issues of ROS-1 is that the centralized design uses best-effort technologies (e.g., HTTP, IP) and depends highly on excellent network connectivity. Therefore, it is not suitable for multi-robot real-time embedded systems. To address this issue, the ROS community recently started ROS-2 which builds its communication system around Data Distributed Service (DDS) to achieve system decentralization. DDS is a widely deployed OT communication protocol for industrial and real-time critical system. However, DDS as a middleware protocol tends to have scalability and reliability issues when implemented over" *i)* wireless networks; and *ii)* non-local area networks.

In order to deal with such issues, initial efforts have been made to bring Zenoh in ROS-2[11] to solve the reliability issues of DDS over error-prone and large scale networks (e.g., WLAN and Internet) whilst keeping a significant level of time and space efficiency.[12]. Therefore, in the coming years, we can expect ROS-2 to gradually integrate Zenoh as a middleware so as to achieve: *i)* more hard real-time and reliable robotics applications; and *ii)* a higher degree of decentralization and distribution of robotics applications in more heterogeneous network scenarios.

6.2. Wireless local area network

WLAN is a part of the IEEE 802 set of LAN protocols, and, as a non-deterministic protocol, is unsuitable for hard real-time applications. The media access control protocol with its backoff algorithm prevents the network from supporting hard real-time communication due to its random delays and potential transmission failures. To address this issue, Time-Sensitive Networking (TSN) and IEEE 802.11 groups are working together towards equipping IEEE 802.11 wireless networks with real-time capabilities [54] [55]. Hence, we recommend that future researchers consider this extension of TSN to ensure the performance of the robotics system over WLAN.

6.3. Information Technologies

This work makes use of the virtualized edge to run access points, which is one example of an application requiring hardware support in the edge. Another application with such needs is to provide AI capabilities on-demand (GPU or hardware accelerators needed). Hypervisors (e.g., Hyper-V, KVM) and container systems support such virtual replication of specific hardware by enabling pass-through or user namespaces. However, in our case, using a hardware-specific (e.g., TPU, GPU, WiFi/Bluetooth adapters) virtualization involves a lot of installation and integration which leads to high configuration complexity. Recently, Wireless Network Virtualization (WNV) has been proposed as an exciting innovation that enables physical wireless network infrastructure resources to be abstracted into virtual resources and shared by multiple parties [56]. As a future research direction, this technology may be used to introduce the RAT-as-a-Service (RaaS). Customers can run and manage access point capabilities in the edge without the complexity of developing and testing RAT applications.

6.4. Orchestration and control

Today's mainstream orchestration solutions are designed with an IT environment in mind, in which all resources are similar and reside in data-centers, and resources are interconnected, leveraging the highly-reliable and high-throughput network technologies. In contrast edge and fog robotic environment is made of heterogeneous, volatile and mobile resources interconnected by OT protocols and unreliable wireless technologies. Such differences require research work to improve the existing orchestration solutions. There is a gap in the area of describing resources and their inter-connectivity in a heterogeneous environment. This can be addressed by designing a unified information model that can describe both IT and OT infrastructures. There is a need to work on infrastructure monitoring. Existing solutions contain stream-based monitoring, thus requiring high-bandwidth networks to update the orchestrators. There is a need to develop new monitoring paradigms that can work over unreliable and low-bandwidth networks. Finally, volatility and mobility introduce a different life-cycle for the resources. Research work is required to track the state changes for such resources to improve the reliability of applications over such unstable infrastructure.

6.5. Development and IT operations

Moving robotics algorithms in the edge of the network requires frameworks that facilitate this transition. Frameworks for robotics systems have been introduced in the past (e.g., Rapyuta, RoboEarth, RoboBrain, etc.). Their focus is mostly to ease and automate the software development. However, when it comes to testing the end-to-end robotics system, we need to follow the stepwise refinement approach that slows down the development process mainly because it is tightly coupled with the physical infrastructure (e.g., access points, robots). To address this issue, simulation environments have been geared specifically towards robotics (Bullet, OpenRAVE, Gazebo, CoppeliaSIM) to improve the sim-to-real domain adaptation. Therefore, a future research direction is to develop models which can consider the edge robotics production environments. The reason is that, an end-to-end edge robotics system is highly heterogeneous and has various types of resources in the physical infrastructure. A digital twin solution that supports such heterogeneous infrastructure can be involved in the development process to provide faster realization of an end-to-end edge robotics system.

[11] https://github.com/osrf/zenoh_evaluation [Accessed: 20 April 2022]

[12] The minimum wire overhead introduced by Zenoh is 4 bytes. The tiniest Zenoh implementation can run in 300 bytes of RAM on an 8-bit microcontroller. Information retrieved from the Zenoh project website: http://zenoh.io/ [Accessed: 20 April 2022]

7. Conclusions

This research presents an exemplary design of an edge robotics system that leverages on real-time context-aware collaboration. The proposed edge robotics system is experimentally evaluated, and the results show how the network-assisted D2D communication can improve the coordination between mobile robots. Additionally, we prove the usefulness of network contextual information when migrating the virtual access point functionality.

The discussion on the future extension of this research is based on our experience of integrating the existing IT/OT technologies, covering the suitability of WLAN, robotics, orchestration, and virtualization solutions and testing platforms for edge robotics. As for our future work, we plan to explore the applicability of TSN to IEEE 802.11 for the remote control of mobile robots over WLAN. We also plan to investigate the usage of the RAT-as-a-Service concepts for edge and fog environments to ease the development, deployment, and management of virtualized RAT applications.

Acknowledgements

This work has been partially funded by European Union's Horizon 2020 research and innovation programme under grant agreement No 101015956, and the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO 5G I+D 6G-EDGEDT and 6G-DATADRIVEN

References

[1] P. K. Garimella, It-ot integration challenges in utilities, in: 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), 2018, pp. 199–204.

[2] F. Tao, Q. Qi, New it driven service-oriented smart manufacturing: Framework and characteristics, IEEE Transactions on Systems, Man, and Cybernetics: Systems 49 (1) (2019) 81–91.

[3] A. Hahn, Operational Technology and Information Technology in Industrial Control Systems, Springer International Publishing, Cham, 2016, Ch. Operational Technology and Information Technology in Industrial Control Systems, pp. 51–68.

[4] J. Wan, S. Tang, H. Yan, D. Li, S. Wang, A. Vasilakos, Cloud robotics: Current status and open issues, IEEE Access 4 (2016) 1–1.

[5] B. Kehoe, S. Patil, P. Abbeel, K. Goldberg, A survey of research on cloud robotics and automation, IEEE Transactions on Automation Science and Engineering 12 (2) (2015) 398–409.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet of Things Journal 3 (5) (2016) 637–646.

[7] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, P. A. Polakos, A comprehensive survey on fog computing: State-of-the-art and research challenges, IEEE Communications Surveys Tutorials 20 (1) (2018) 416–464.

[8] M. Peng, S. Yan, K. Zhang, C. Wang, Fog-computing-based radio access networks: issues and challenges, IEEE Network 30 (4) (2016) 46–53.

[9] S. Wan, Z. Gu, Q. Ni, Cognitive computing and wireless communications on the edge for healthcare service robots, Computer Communications 149 (2020) 99 – 106.

[10] J. P. Queralta, L. Qingqing, Z. Zou, T. Westerlund, Enhancing autonomy with blockchain and multi-access edge computing in distributed robotic systems, in: 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), 2020, pp. 180–187.

[11] Q. Zhang, J. Liu, G. Zhao, Towards 5g enabled tactile robotic telesurgery (2018). arXiv:1803.03586.

[12] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, Q. Zhang, Edge computing in iot-based manufacturing, IEEE Communications Magazine 56 (9) (2018) 103–109.

[13] L. Hu, Y. Miao, G. Wu, M. M. Hassan, I. Humar, irobot-factory: An intelligent robot factory based on cognitive manufacturing and edge computing, Future Generation Computer Systems 90 (2019) 569 – 577.

[14] Q. Qi, F. Tao, A smart manufacturing service system based on edge computing, fog computing, and cloud computing, IEEE Access 7 (2019) 86769–86777.

[15] S. Nunna, A. Kousaridas, M. Ibrahim, M. Dillinger, C. Thuemmler, H. Feussner, A. Schneider, Enabling real-time context-aware collaboration through 5g and mobile edge computing, in: 2015 12th International Conference on Information Technology - New Generations, IEEE Computer Society, USA, 2015, p. 601–605.

[16] A. K. Tanwani, N. Mor, J. Kubiatowicz, J. E. Gonzalez, K. Goldberg, A fog robotics approach to deep robot learning: Application to object recognition and grasp planning in surface decluttering (2019). arXiv:1903.09589.

[17] L. Qingqing, J. P. Queralta, T. N. Gia, H. Tenhunen, Z. Zou, T. Westerlund, Visual odometry offloading in internet of vehicles with compression at the edge of the network, in: 2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU), 2019, pp. 1–2.

[18] K. Antevski, M. Groshev, L. Cominardi, C. J. Bernardos, A. Mourad, R. Gazda, Enhancing edge robotics through the use of context information, in: Proceedings of the Workshop on Experimentation and Measurements in 5G, ACM, 2018, p. 7–12.

[19] I. A. Tsokalo, H. Wu, G. T. Nguyen, H. Salah, F. H.P. Fitzek, Mobile edge cloud for robot control services in industry automation, in: 2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC), 2019, pp. 1–2.

[20] M. Fu, S. Sun, K. Ni, X. Hou, Mobile robot object recognition in the internet of things based on fog computing, in: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2019, pp. 1838–1842.

[21] D. Pakkala, J. Koivusaari, P. Pääkkönen, J. Spohrer, An experimental case study on edge computing based cyber-physical digital service provisioning with mobile robotics, in: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020, pp. 1165–1174.

[22] J. Soldatos, S. Kyriazakos, P. Ziafati, A. Mihovska, Securing iot applications with smart objects: Framework and a socially assistive robots case study, Wireless Personal Communications (02 2020).

[23] M. S. Shaik, V. Struhár, Z. Bakhshi, V.-L. Dao, N. Desai, A. V. Papadopoulos, T. Nolte, V. Karagiannis, S. Schulte, A. Venito, G. Fohler, Enabling fog-based industrial robotics systems, in: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vol. 1, 2020, pp. 61–68.

[24] N. Mohamed, J. Al-Jaroodi, I. Jawhar, Utilizing fog computing for multi-robot systems, in: 2018 Second IEEE International Conference on Robotic Computing (IRC), 2018, pp. 102–105.

[25] L. Qingqing, F. Yuhong, J. P. Queralta, T. N. Gia, H. Tenhunen, Z. Zou, T. Westerlund, Edge computing for mobile robots: Multi-robot feature-based lidar odometry with fpgas, in: 2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU), 2019, pp. 1–2.

[26] B. V. Bhausaheb, P. S. Saikrishna, Control algorithms for a mobile robot application in a fog computing environment, in: Proceedings of the 2019 3rd International Conference on Automation, Control and Robots, ACM, 2019, p. 30–36.
[27] A. Kattepur, H. Dohare, V. Mushunuri, H. K. Rath, A. Simha, Resource constrained offloading in fog computing, in: Proceedings of the 1st Workshop on Middleware for Edge Clouds amp; Cloudlets, ACM, 2016, pp. 1–6.
[28] J. Zhu, Y. Chen, M. Zhang, Q. Chen, Y. Guo, H. Min, Z. Chen, An edge computing platform of guide-dog robot for visually impaired, in: 2019 IEEE 14th International Symposium on Autonomous Decentralized System (ISADS), 2019, pp. 1–7.
[29] V. K. Sarker, J. Peña Queralta, T. N. Gia, H. Tenhunen, T. Westerlund, Offloading slam for indoor mobile robots with edge-fog-cloud computing, in: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019, pp. 1–6.
[30] S. Dey, A. Mukherjee, Robotic slam: A review from fog computing and mobile edge computing perspective, in: Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services, ACM, 2016, p. 153–158.
[31] O. Arsene, C. Postelnicu, W. Wang, E. Simona Lohan, D. Iulian Nastac, An architecture for indoor location-aided services based on collaborative industrial robotic platforms, in: 2019 International Conference on Localization and GNSS (ICL-GNSS), 2019, pp. 1–6.
[32] V. Petrov, M. A. Lema, M. Gapeyenko, K. Antonakoglou, D. Moltchanov, F. Sardis, A. Samuylov, S. Andreev, Y. Koucheryavy, M. Dohler, Achieving end-to-end reliability of mission-critical traffic in softwarized 5g networks, IEEE Journal on Selected Areas in Communications 36 (3) (2018) 485–501.
[33] D. Rapone, R. Quasso, S. B. Chundrigar, S. T. Talat, L. Cominardi, A. De la Oliva, P.-H. Kuo, A. Mourad, A. Colazzo, G. Parmeggiani, A. Z. Orive, C. Lu, C.-Y. Li, An integrated, virtualized joint edge and fog computing system with multi-rat convergence, in: 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2018, pp. 1–5.
[34] Q. Li, Y. Tian, Q. Wu, Q. Cao, H. Shen, H. Long, A cloud-fog-edge closed-loop feedback security risk prediction method, IEEE Access 8 (2020) 29004–29020.
[35] D. He, Y. Qiao, S. Chan, N. Guizani, Flight security and safety of drones in airborne fog computing systems, IEEE Communications Magazine 56 (5) (2018) 66–71.
[36] S. Wang, Cloud-dew architecture, International Journal of Cloud Computing 4 (3) (2015) 199–210.
[37] M. De Donno, K. Tange, N. Dragoni, Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog, IEEE Access 7 (2019) 150936–150948.
[38] N. Jangid, Real time cloud computing, in: 1st National Conference on Data Managment & Security (DaMS), 2011.
[39] A. Botta, L. Gallo, G. Ventre, Cloud, fog, and dew robotics: Architectures for next generation applications, in: 2019 7th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2019, pp. 16–23.
[40] K. Ren, C. Wang, Q. Wang, Security challenges for the public cloud, IEEE Internet Computing 16 (1) (2012) 69–73.
[41] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, D. Sabella, On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration, IEEE Communications Surveys Tutorials 19 (3) (2017) 1657–1681.
[42] O. C. A. W. Group, Ieee approved draft standard for adoption of openfog reference architecture for fog computing, IEEE P1934/D2.0 (2018) 1–175.
[43] I. I. C. W. Committees, The industrial internet of things volume g1: Reference architecture, `https://www.iiconsortium.org/IIRA.htm`, accessed: 2022-4-20 (2019).
[44] M. N. Tehrani, M. Uysal, H. Yanikomeroglu, Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions, IEEE Communications Magazine 52 (5) (2014) 86–92.
[45] B. Bangerter, S. Talwar, R. Arefi, K. Stewart, Networks and devices for the 5g era, IEEE Communications Magazine 52 (2) (2014) 90–96.
[46] ETSI, Developing Software for Multi-Access Edge Computing, Group Specification (GS) 2nd edition, European Telecommunications Standards Institute (ETSI) (2 2019).
[47] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklós, Z. Turányi, Design aspects of network assisted device-to-device communications, IEEE Communications Magazine 50 (3) (2012) 170–177.
[48] U. Hunkeler, H. L. Truong, A. Stanford-Clark, Mqtt-s — a publish/subscribe protocol for wireless sensor networks, in: 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08), 2008, pp. 791–798.
[49] R. Fielding, Architectural styles and the design of network-based software architectures, Ph.D. thesis, University of California, Irvine (01 2000).
[50] G. Pardo-Castellote, Omg data-distribution service: architectural overview, in: 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings., 2003, pp. 200–206.
[51] T. S. Andjamba, G. L. Zodi, D. S. Jat, Interference analysis of ieee 802.11 wireless networks: A case study of namibia university of science and technology, in: 2016 International Conference on ICT in Business Industry Government (ICTBIG), 2016, pp. 1–5.
[52] S. Bangolae, C. Bell, E. Qi, Performance study of fast bss transition using ieee 802.11r, in: Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing, ACM, New York, NY, USA, 2006, p. 737–742.
[53] A. Corsaro, G. Baldoni, fogØ5: Unifying the computing, networking and storage fabrics end-to-end, in: 2018 3rd Cloudification of the Internet of Things (CIoT), 2018, pp. 1–8.
[54] S. Kim, M. M. Rashid, S. Deo, J. Perez-Ramirez, M. Galeev, G. Venkatesan, S. Dey, W. Li, D. Cavalcanti, Demo/poster abstract: Enabling time-critical applications over next-generation 802.11 networks, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2018, pp. 1–2.
[55] S. Bush, G. Mantelet, Industrial Wireless Time-Sensitive Networking: RFC on the Path Forward, White paper Version 1.0.3, Avnu Alliance (1 2018).
[56] C. Liang, F. R. Yu, Wireless network virtualization: A survey, some research issues and challenges, IEEE Communications Surveys Tutorials 17 (1) (2015) 358–380.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: