Edge Robotics Experimentation over Next Generation IIoT Testbeds

Dimitrios Dechouniotis*, Dimitrios Spatharakis*, Symeon Papavassiliou*
*School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece
{ddechou, dspatharakis}@netmode.ntua.gr, papavass@mail.ntua.gr

***Abstract*—The emergence of Industrial Internet of Things (IIoT) requires the interconnection between robots, sensors, and the underlying network and computing infrastructure. Edge Robotics has emerged as a flexible paradigm that enables resource-constrained mobile robots to offload computationally intensive tasks of time/mission-critical applications. In this context, Edge Computing is essential for providing additional resources towards confronting the stringent performance specifications. This article presents the architectural concepts and capabilities of the NETMODE testbed, member of the Fed4FIRE+ federation, for the state-of-the-art experimentation with robotic applications. An evaluation of the proposed architecture is conducted using a SLAM algorithm which is a compute-intensive application.**

***Index Terms*—Edge Robotics, Cyber-Physical Systems, Edge Computing**

I. Introduction

This article focuses on the case of Edge Robotics [1], that are widely used in 5G industrial verticals [6]. In this context, robot manipulators or mobile robots are equipped with various sensors and their operation relies on computationally intensive data processing and algorithms, the local execution of which usually leads to performance violations. To this end, Cloud Computing can provide the essential computing resources to execute these applications as network services, while respecting the existing performance requirements [2]. Furthermore, following the current trend in service delivery, Fog and Edge Robotics leverage the computing capabilities of Edge Computing to achieve low-latency communication [3]. In this case, an offloading mechanism is responsible for transmitting the sensing data to an edge server for further processing. However, offloading could lead to service degradation if the transmission rate is unreliable or the edge server runs out of available computing resources.

For IIoT applications, the collaboration between robots and the underlying network and computing infrastructure is necessary to investigate the trade-off between the remote and local execution of complex algorithms. To this extent, next-generation internet technologies and infrastructure are required for experimenting with such scenarios. Being the largest federation of next generation internet (NGI) testbeds, Fed4FIRE+ project [1] provides cloud, wired, and wireless, OpenFlow, and IoT resources, connected with a high-speed optical fiber network, which provides open, accessible, and reliable facilities that support a wide variety of different research and innovation communities and initiatives in Europe [7] .

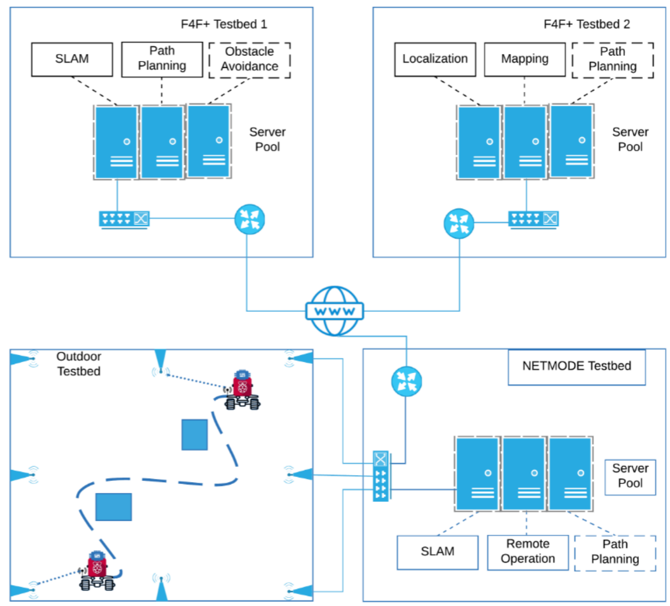

Fig. 1. Industrial Robotics Experimentation over Federated Testbeds

A. NETMODE Testbed

The NETMODE testbed[2] of the National Technical University of Athens, is an Edge Computing-enabled testbed, which provides computing, wireless, and IIoT infrastructure components and enables the deployment and evaluation of modern smart applications and Industry 4.0 use cases. NETMODE testbed provides the experimenters with mobile robots, equipped with various sensors, for designing and executing smart manufacturing experiments in both indoor and outdoor environments. Therefore, the experimenters are able to reserve a group of heterogeneous resources, orchestrate a cyber-physical system (CPS), and experiment on smart manufacturing and Industry 4.0 scenarios. This article presents how the NETMODE testbed can be utilized to orchestrate heterogeneous computing, network resources, and mobile robots

[1] https://www.fed4fire.eu/testbeds/

[2] https://www.fed4fire.eu/testbeds/netmode/

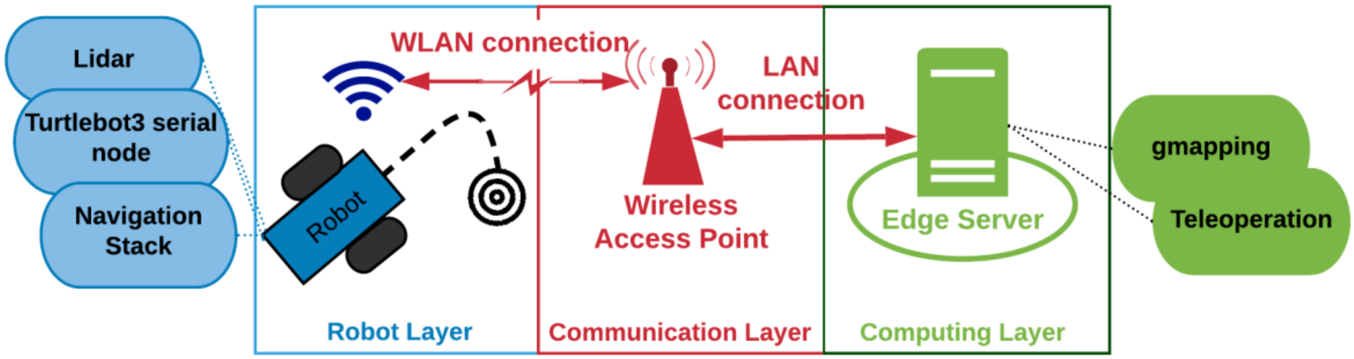

Fig. 2. The ROS packages for the autonomous navigation and SLAM scenario.

to experiment on complex smart manufacturing scenarios. In Figure 1, an example architecture of the Fed4FIRE+ testbeds is illustrated.

In particular, the available mobile robots;, TurtleBot3 Burgers[3], are controlled by Raspberry Pis 3 Model B+ (4vCPUs) and equipped with high-definition camera and LiDAR modules along with motion sensors (encoders). Although this setting is able to support most of the modern robotic applications, the quality of service (QoS) of complex applications is far from being satisfied because of the limited processing capabilities of the Raspberry. Such applications refer to (and are not limited to) localization algorithms, motion planning, and Simultaneous Localization and Mapping (SLAM). To incorporate the execution of these complex algorithms, the NETMODE testbed, also offers the remote resources of the edge computing infrastructure, which include servers and Intel NUC-based processing nodes. Through cutting-edge WiFi nodes (supporting 802.11ac protocol), mobile robots are able to offload these compute-intensive applications to be executed on powerful machines.

Therefore a complex CPS is formulated. Nevertheless, offloading could lead to service degradation, if the transmission rate is unreliable or the edge server runs out of available computing resources [4]. Under this complex scenario, the fundamental trade-off between performance and consumed resources can be investigated. For instance, in time-critical IIoT applications, a trade-off between navigation accuracy and mission duration must be investigated. Moreover, the system performance can be expressed through various indicators such as energy consumption, network delay, or mission accuracy. Under this setting, it is evident that a decision mechanism is necessary for the robot to determine whether compute-intensive tasks should be executed remotely on the edge server-side. This decision must also consider the network conditions and the available resources of the servers. Hence, one can develop switching techniques [5] that incorporate the offloading decision. This two-layer system can assist robotic applications, reduce the mission completion time and minimize the resource utilization of the resource-constrained TurtleBots. For example, a human operator could remotely control the robot by employing live camera feeds or data from various sensors to assist in an IIoT application (e.g., warehouse robotics).

II. Edge-Assisted SLAM

Leveraging the proposed architecture, an edge-assisted SLAM application is deployed and evaluated. For controlling the robot's motion and developing the robotic applications, we select the Robotic Operating System (ROS)[4] , which allows the easy deployment of off-the-shelf algorithms for robotic applications. Figure 2 shows various ROS packages that can be executed on both the robot and the edge server. The compute-intensive application, i.e., *gmapping* [5] which is used for SLAM, is deployed on the server side. Specifically, *gmapping* application creates a 2-D occupancy grid map, building floorplan, from the LiDAR sensor and the pose information collected by the odometry sensors of the mobile robot.

On the other hand, the necessary libraries to control the motion of the ROS-enabled TurtleBot (i.e., serial node, navigation stack) and employ the LiDAR sensor are executed on the robot. Hence, the autonomous navigation (ROS navigation stack) is consolidated to the robot, while the LiDAR transfers raw data through a wireless connection to the edge server. We used an Intel NUC kit (barebone) with Intel Core i7-10710U (12M Cache, 4.70 GHz) and 8GB RAM DDR4, to act as edge server in this scenario. A sample output of the SLAM algorithm in the outdoor testbed on NETMODE is presented in Figure 3. Finally, in the available teleoperation mode, an operator is able to move remotely the robot in the outdoor testbed of NETMODE in real-time, until a map is created.

Under this setting, we experimented with a parameter of the SLAM algorithm, i.e., the interval (in seconds) for map update. By lowering this value the occupancy grid is updated

[3]https://www.roscomponents.com/en/mobile-robots/214-turtlebot3-burger.html#/courses-no/turtlebot_3_burger_model-burger_intl_

[4]https://www.ros.org/

[5]http://wiki.ros.org/gmapping

Execution	Map update Interval(s)	Average CPU Utilization(%)
Local	0.5	62%
	1	45%
	2	33%
	5	24%
Remote	0.2	30%
	0.5	27%
	1	22%
	2	17%
	5	12%

TABLE I
EXPERIMENT EVALUATION.

more often, at the expense of a larger computational load. Hence, for various values of this parameter, we evaluated the CPU utilization of the *gmapping* algorithm, when executed on the robot (locally) or the edge server (remotely). The CPU utilization is measured in process level with htop tool. Table I presents the CPU utilization of the SLAM algorithm for both local and remote execution. As it is shown, the shorter mapping interval is, the higher utilization occurs. Furthermore, it is worth-notable the significant increase of the CPU utilization for small values (i.e., 0.5sec and 1sec), which is evident due to the constrained processing capabilities of the Raspberry Pi. On the other hand, the CPU utilization of the edge server remains low, allowing the parallel execution of other compute-intensive applications. Additionally, for large operating grounds, saving the constructed map (i.e., the occupancy grid) on the Raspberry device is not feasible due to limited storage capabilities.

To sum up, the proposed architecture could assist ROS-enabled mobile robots in saving valuable resources to execute complex time/mission-critical tasks, e.g., object recognition and SLAM.

III. CONCLUSIONS AND FUTURE WORK

This demo article presented a two-layer architecture for the deployment of ROS-enabled applications in the context of edge robotics. Based on a SLAM application, the experimental results show significant performance acceleration of the compute-intensive tasks when edge infrastructure is utilized. In our future plans, we aim at integrating ROS along with lightweight versions of Kubernetes resource orchestration platform in order to design and deploy efficient resource scaling mechanisms for the containerized robotic applications.

ACKNOWLEDGMENT

This work was supported in part by H2020 Fed4FIREplus Project under Grant Agreement No. 732638.

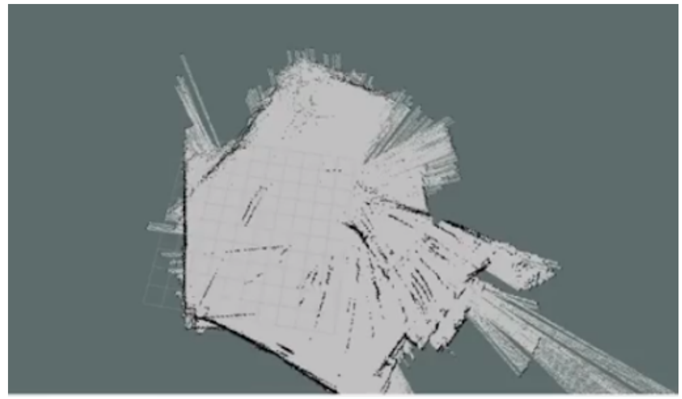

Fig. 3. An example of the map constructed in the outdoor testbed of NETMODE.

REFERENCES

[1] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg. "A survey of research on cloud robotics and automation," IEEE Transactions on automation science and engineering, vol. 12, no. 2, pp. 398-409, 2015.

[2] M. Afrin, J. Jin, A. Rahman, A. Rahman, J. Wan, and E. Hossain. "Resource Allocation and Service Provisioning in Multi-Agent Cloud Robotics: A Comprehensive Survey," IEEE Communications Surveys & Tutorials, vol. 23, is. 2, pp. 842-870, 2021.

[3] D. Spatharakis, M. Avgeris, N. Athanasopoulos, D. Dechouniotis and S. Papavassiliou, "Resource-aware Estimation and Control for Edge Robotics: a Set-based Approach," IEEE Internet of Things Journal, Early Access, 2022.

[4] S. Chinchali, A. Sharma, J. Harrison, A. Elhafsi, D. Kang, E. Pergament, E. Cidon, S. Katti, and M. Pavone. "Network offloading policies for cloud robotics: a learning-based approach", Springer Autonomous Robots, vol. 45, no. 7, pp. 997–1012, 2021.

[5] D. Spatharakis, M. Avgeris, N. Athanasopoulos, D. Dechouniotis and S. Papavassiliou. "A Switching Offloading Mechanism for Path Planning and Localization in Robotic Applications," 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), pp. 77-84, 2020.

[6] I. Dimolitsas, M. Avgeris, D. Spatharakis, D. Dechouniotis, S. Papavassiliou. "Enabling Industrial Network Slicing Orchestration: A Collaborative Edge Robotics Use Case," 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), pp.215–220,2021.

[7] D. Dechouniotis, I. Dimolitsas, K. Papadakis-Vlachopapadopoulos, S. Papavassiliou. "Fuzzy multi-criteria based trust management in heterogeneous federated future internet testbeds," Future Internet, vol. 10, is. 7, no. 58, 2018.