2021 IEEE 11th International Conference on System Engineering and Technology (ICSET 2021), 6 November 2021, Shah Alam, Malaysia

Low Latency Peer to Peer Robot Wireless Communication with Edge Computing

C. Asavasirikulkij*, C. Mathong*, T.Sinthumongkolchai**, R. Chancharoen*, W. Asdornwised**
* Department of Mechanical Engineeering, Chulalongkorn University
** Department of Electrical Engineeering, Chulalongkorn University

Abstract — **A new peer-to-peer wireless communication platform is the technology developed for a mobile robot that works in a smart factory. It features high bandwidth and low latency communications such that the mobile robot can communicate with an edge computer in real-time application. In our demonstration, a mobile robot is loaded with sensors and connected to an edge computer and a serial link manipulator. Successfully implement the algorithm on the edge computer using Lidar scan from the mobile robot during operations, the positions of the mobile robot are precisely determined. The Lidar scan, which is 2873 bytes, can be real-time updated at 150 milliseconds with a latency of 1.5 milliseconds. Our proposed wireless network can connect all the machines in an automation cell, i.e., a mobile robot, a manipulator, a warehouse, and an edge computer. Particularly, on our edge computer, the edge computer collects all the data from all the machines in-house to generate an up-to-date map used to control the automation cell during the robot's collaboration work. Here, a use case is where the mobile robot picks up a package from the manipulator and then delivers it to the warehouse. In the meantime, both the mobile robot and the manipulator are in motion during the package transfer.**

Keywords — Cyber-Physical Systems (CPS); Smart Factory; Edge Computing; Mobile Robot Navigation

I. INTRODUCTION

As we are heading towards the fourth industrial revolution, production is enhanced so that a manufacturing plant, called a smart factory, is highly adaptive and flexible. One of the key elements in the next generation plant is robots, especially a mobile form. The robots will either work autonomously or collaborate with other machines to effectively complete their jobs. Currently robots come with high processing power and a precision motion that can function autonomously in their working environment and thus play an important role in the factory. One of the key technologies behind the fourth industrial revolution is wireless network performance. This paper proposes a novel and comprehensive wireless network for a next-generation factory for a mobile robot to communicate with another machine and real-time interact with them to complete a task. The mobile robot can work in a considerably larger space compared to the machine with a wire attached. In such a scenario, the big concern is its position in the working area that requires advanced sensors and high processing computation.

Many of the SLAM [1] methodologies for a mobile robot have been extensively researched and developed. These include range scanner [2], RFID [3], odometry [1], and machine vision [4][5]. Each method has its benefits and drawbacks. Thus, in our work, two methods, including (1) odometry and (2) range scanner, are implemented. Then, we addressed how peer-to-peer wireless communication could effectively benefit from these two methods.

Briefly, odometry [1][2] estimates the position of a mobile robot from the rotational position of its wheel, sensed by encoders, with the knowledge of its kinematics and dynamics. However, the typical two-wheel robot is a non-holonomic system [6][7], and the resulting estimation provides relative position with an incremental error along the walk of the robot. In addition, the robot using the odometry method cannot perceive its movement error since it believes that encoders perform ideally and its trajectory is correct.

Range scanner, the second method in our study, uses Lidar sensor to scan the environment. It is similar to machine vision except that it uses its laser as a light source, and the resulting image is only one-dimensional. In this way, the result is more reliable and easier to process to determine the robot's position. The SLAM method provides an absolute position that is applied to the robot to solve the odometry issues. The current position of a robot is important but also the time for the data to be ready. The current position controls the mobile robot in real-time and informs the collaborated machine during the collaboration work. The latency of the data is very important in this scenario. Our proposed wireless network connected to the edge computer is the key infrastructure for a smart factory that can effectively handle a huge computation and data latency on the floor, especially during a critical task.

Recently, wireless technology based on wireless protocol IEEE 802.11 has been developed and easily deployed at various places, i.e., airports, hotels, offices, malls, and home residences, etc. The sixth generation of wireless protocol 802.11ax has been launched and provides greater network capacity, higher efficiency, better performance, and reduced latency than the previous fifth-generation wireless protocol 802.11ac [8]. The requirements of low latency and extremely high reliability are essential for some industries (e.g., industrial automation, autonomous driving, etc.).

In addition, according to the record of Cisco, machine-to-machine communication is calculated to grow from 780 million in 2016 to 3.3 billion by 2021, which makes the core network suffers from the extremely huge number of connections (massive Internet of Things; mIoT). This leads to meet the challenges brought by the traffic explosion and diverse requirements [9]. Edge computing [10] substitutes cloud computing to reduce the limitation of bandwidth and latency. With edge computing [11], data processing at the edge of the network reduced collaboration latency between the mobile robot and the collaboration machine.

The scenario of a working automation cell in our work consists of a mobile robot, a robot manipulator, and a warehouse. Section II is about our concerned technologies, noted that the edge computer handles the positions of the mobile robot and the manipulator. The edge computer can access both onboard sensors on the mobile and sensors on the manipulator. This way, our edge computer can efficiently handle the huge computations that are required. In Section III, the experimental demonstration is to validate the effectiveness of the proposed technique.

978-1-6654-3765-3©2021 IEEE

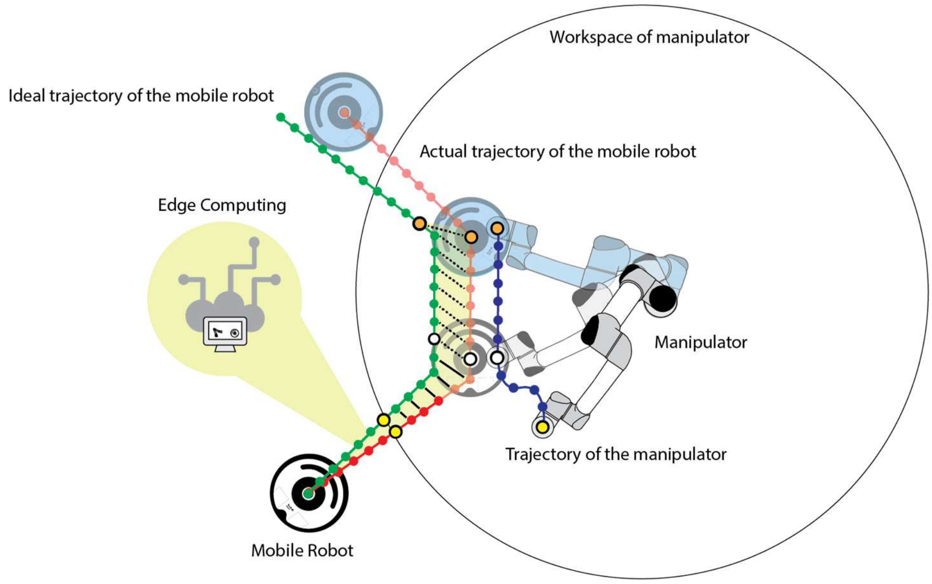


Fig. 1. A mobile-industrial robot collaboration in working cell.

II. Methodology

A. Robot Interaction

The collaboration between a mobile robot and a manipulator can be achieved mainly with precise positions of all machines at a current time. However, the position estimation that is very accurate and real-time is still a challenge with the current methodology and hardware. In Fig. 1, a mobile robot interacts with a manipulator to receive a package. At the same time, both the mobile robot and the manipulator are moving. In this case, both robots must follow the same trajectory at the same time during the drop-off. The trajectory of the manipulator, shown in blue, is assumed precisely controlled and can be changed on the fly. The mobile robot is then programmed to follow a planned path, shown in green. However, the mobile robot must know its current position during its movement along the trajectory and the performance controller to control the robot trajectory with unknown and disturbance. As mentioned earlier, the estimate position comes with an error and a delay. The actual trajectory of the mobile robot is thus a different path, shown in red. The proposed network architecture supports the mobile robot such that it can send all the onboard data to compute and determine the pose of the robot at the edge. The trajectory is then sent to the manipulator robot to correct its path in real-time. This technique is possible only with a wireless network with high data bandwidth and low latency.

However, the mobile robot communicates with the edge computer via a wireless network. The data transmission might cause a delay when data is transmitted to one shared medium (edge computer) from various resources. If there is data latency, all the machines will know only the past positions of the others and estimations are required since the interaction is at present, not the past. The best estimation is at the edge computer that acts as a centralized supervisor that commands the mobile robot and manipulator during its interaction. The edge computer can issue a corrective action to both the mobile robot and the manipulator.

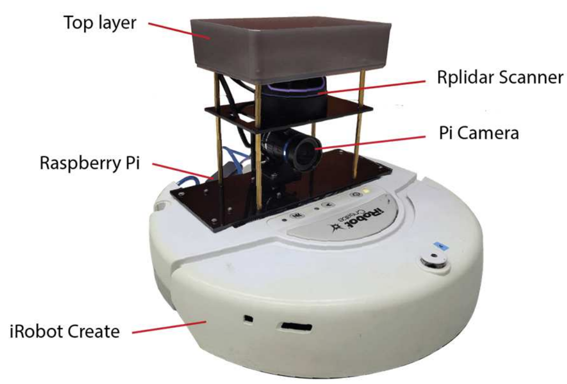


Fig. 2. A connected mobile robot with Lidar and camera.

B. Mobile Robot Localization using Edge Computing

The mobile robot in our study is designed with iRobot Create as a mobile base shown in Fig. 2. The Raspberry Pi, HQ camera, and Lidar are installed on the top of the robot. The onboard Raspberry Pi wirelessly connects to the edge computer. Our robot comes with internal sensors including an encoder which is used for odometry. In this proposed architecture, the robot processes simple tasks onboard and sends large data (bandwidth) to compute for its position on the edge computer. The path generation is also computed on the edge to generate the trajectory for the mobile robot in real-time. The MATLAB's Probabilistic Roadmap generates the robot path in this study. This way, the robot operation is highly flexible and supports the operation in unstructured environments with other machines. Notes that the obstacle avoidance and cliff are under control onboard.

In detail, the onboard processor controls the robot trajectory using the position estimated by odometry. Although the trajectory errors accumulate along the walk, SLAM can correct them. The raw data of Lidar scans is delivered to process at the edge computer where the environment database is dynamically collected. The processing power requires for this method. The higher the

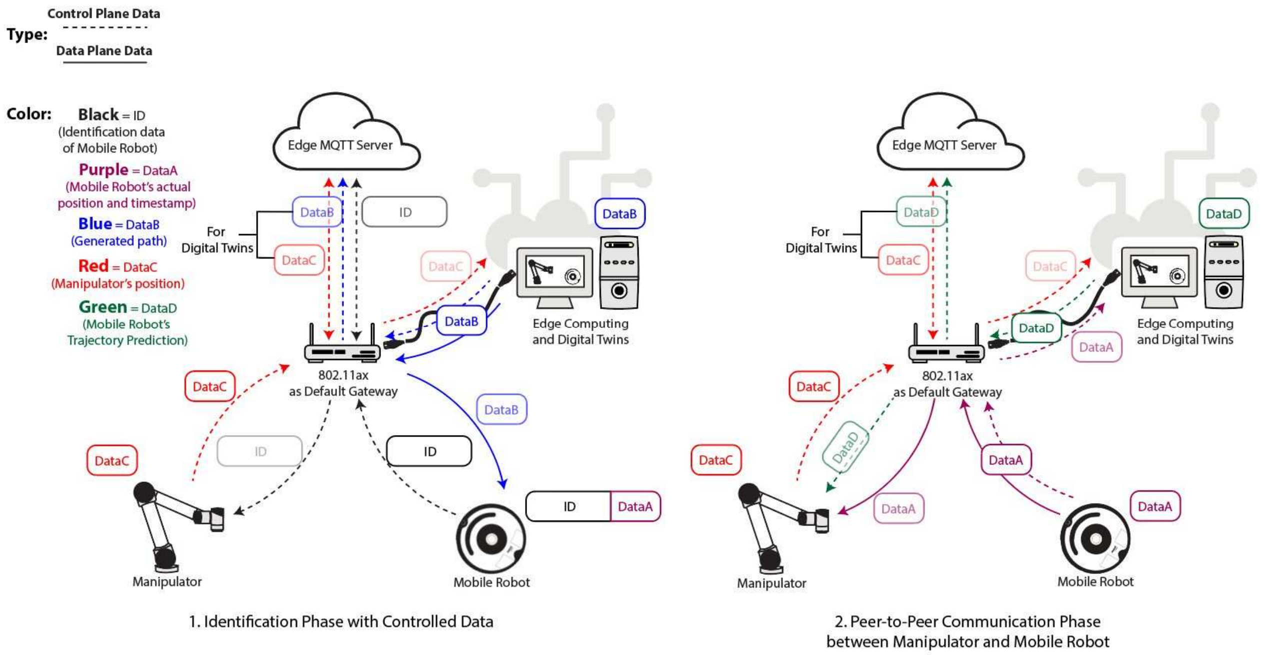


Fig. 3. A network architecture for an automation cell.

update rate, the lesser the accumulative error in the odometry estimation.

The other benefit of this architecture is that the mobile robot will be lighter since onboard processing power and power usage can significantly reduce. The soft computation is handled on the edge and cloud, not on the mobile robot.

C. *Network Communication Architecture*

Considering our network architecture, for handling massive Machine Type Communication (mMTC) in the various zones, all the data will passthrough the W-LAN 802.11ax protocol. This protocol features *Orthogonal Frequency Division Multiple Access (OFDMA) which can* accommodate more users on the same channel, greater network capacity, higher efficiency, and better performance, and reduce latency for machine-to-machine communication [8].

As shown in Fig. 3, we propose a Software-Defined Wireless Networking (SDWN) approach, in which the control plane and data plane are physically separated [12]. The control plane (dash line) component is handled by an MQTT broker, while the data plane (solid line) consists of data from devices that need to respond to each other. In this case, the data plane is the data interaction between the robot manipulator and the mobile robot. Moreover, there is one device in between these planes which is edge computing. In the data plane function, edge computing generates the path (DataB) and sends it to the mobile robot. Thus, the mobile robot will travel along with this trajectory data. Meanwhile, in the control plane function, the MQTT broker acts as a centralized controller collecting the devices' localization data (DataB and DataC) and broadcasting it into the Cyber-Physical Systems (Edge computers) on the in/out-zone edge computing.

For our research scenario, the mobile robot receives the path (DataB) and moves into the manipulator's workspace for the robot interaction. The manipulator should know the mobile robot's trajectory and time when it gets there, so the manipulator can shift along with the mobile robot and send cargo to it. However, there will be position and time errors due to the non-ideal physical environments and the latency of wireless communication, so it is impossible to sync their movement without predicting and optimizing these errors.

The mobile robot sends its actual position from the Lidar sensor with timestamp (DataA) to the edge computing in the control plane function to deal with these problems. The edge computing predicts the future position/timestamp from the generates mobile robot's path (DataB) and the actual position/timestamp of the mobile robot (DataA). Then, the prediction data (DataD) will be sent to the MQTT broker for edge computers and the manipulator. Additionally, the mobile robot also sends its actual position with a timestamp to the manipulator. Hence, the manipulator can compare this data with the prediction data and optimizes its trajectory during the interaction with the mobile robot.

The mobile robot and the manipulator must communicate with low latency and high data rates for a better manipulator's trajectory optimization. The key point of our architecture is to increase the speed and performance of the network by minimizing the packet size of the data created by the devices communicating with TCP/UDP protocol in the data plane. Hence, the mobile robot's data divides into two parts: Identification Data (ID) and Actual data (DataA). In the first phase (identification phase), the Identification Data is sent via the control plane, where the MQTT broker is a centralized controller. The MQTT broker acts as an IP Address management [13], in which the mobile robot publishes its Dynamic IP Address, given by the router, to the MQTT broker topic. Then, the manipulator will know the mobile robot's IP by subscribing to this topic, creating reliable peer-to-peer communication in the data plane at the second phase (P2P communication phase). Finally, the Actual data part left is for streaming using TCP/IP or UDP/IP protocol in the data plane. The packet will be more lightweight, so the transmission of data rates improves, and the manipulator's trajectory adjustment will be more accurate.

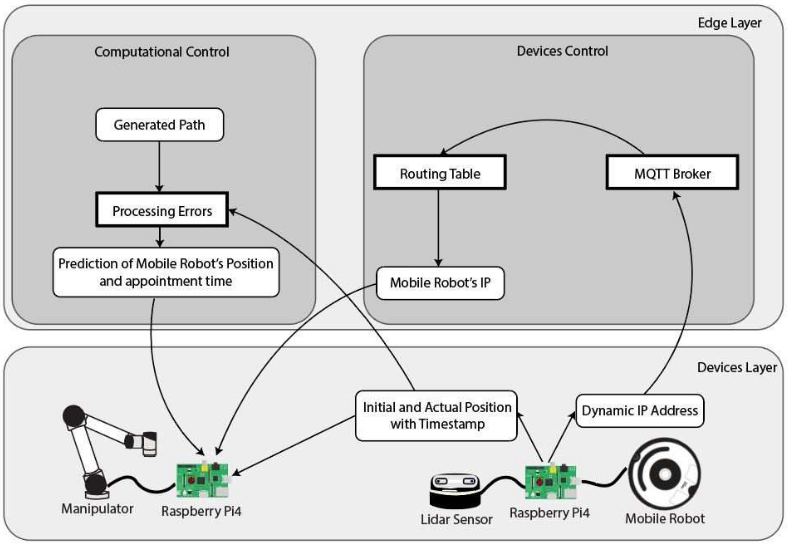


Fig. 4. Data management of the control function (Generated Path to robot)

D. Algorithm of Data Management and Analysis

When sending a package to the warehouse, the robot manipulator will request a mobile robot from the edge computer. Edge computer will assign a vacant mobile robot in the workspace to connect to the robot manipulator. Then, data of where to pick up the package (endpoint coordinate) and what type of package (where to deliver the package) are sent to the mobile robot. First, the mobile robot should initialize itself using Lidar for its current position (x-coordinate, y-coordinate) and orientation (theta). Then, the path was generated in the edge computer from the initialized point coordinate to the endpoint coordinate. These waypoints were sent to the mobile robot and edge computer.

Lidar collects and sends its current position with the timestamp to the edge computer along the path, as shown in Fig. 4. With the edge computer simulation running in parallel with the physical world, absolute position from the Lidar will be used to compare with the position in the edge computer.

Since the error of mobile robot increases with time, therefore we use the weighted moving average calculation which gives more weight to recent data than the past data. Errors will be calculated in real-time with the past dataset and new dataset to ensure correctness. Having calculated the errors in the edge computer, the edge computer will know the mobile robot path from the start till the end from the configurated simulation in which the position at different timestamp is known from calibrated odometry formula. Having known the position error and latest position, we can map position to the future position of the real physical mobile robot in the workspace.

E. Time Delay Measurement

To measure the delay of the wireless network, the One-Way Delay metric (OWD) is the most suitable one. This metric calculates the difference in the time interval between the timestamp of the sent package and the received package. Thus, two different computer clocks are used with the additional process to synchronize these clocks [14].

As shown in Fig. 5, edge computing is used to synchronize a client's clock with a given time reference. All the devices' programming interfaces connect to the edge computing server and optimize their time compared to the reference. Then, their clocks will synchronize with optimal precision. The sender's device sends a packet with its timestamp to the receiver side. While receiving the packet, the receiver's device also collects its timestamp. Hence, the receiver will measure the time delay by calculating the difference between its timestamp and the sender's timestamp based on the same time reference.

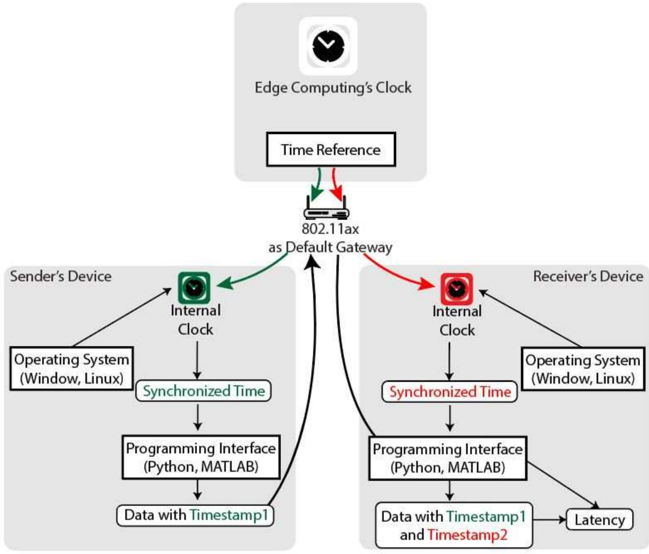


Fig. 5. Latency measurement method using time synchronization from the Global Clock.

III. Experimental Results

In this experimental demonstration, the system consisted of one unit of edge computing, one mobile robot and one manipulator. Dell Inspiron 157000, with intel core i7, Window 10, Nvidia GeForce GTX 1060 and RAM 8 GB, was used as the edge computer. For the collaboration between the mobile robot and the manipulator, both machines must share the same map where everything in the automation cell is up to date. The mobile robot, loaded with sensors, can free roam in the workspace and peer-to-peer wireless communication with the edge computer. In the experiment, the mobile robot successfully sent its onboard Lidar scan to the edge where SLAM algorithm was executed. Now, the edge computer saw the current position of the mobile robot as well as part of the

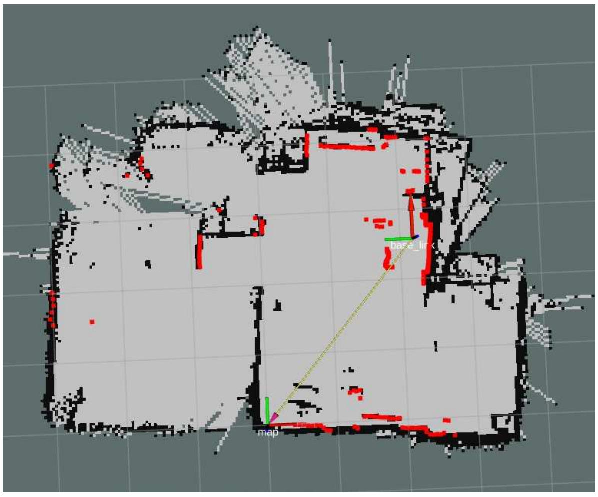


Fig. 6. Mobile robot navigation in ROS with SLAM method

environment, as seen from the Lidar scan. The digital map was then updated, shown in Fig. 6. Peer-to-peer wireless communication is the vital key to implement this technique.

In the collaboration work, the robot still needed a safe path where it could walk into the workspace of the manipulator. In our proposed architecture, the edge computer held the updated map and organized the machines during the collaboration task. The edge computer generated and selected the recommended path for the mobile robot, shown in Fig. 7. In this experiment, Probabilistic Roadmap (PRM) was executed on the powerful edge computer. The path was then sent to the mobile robot. In the experiment, the digital map was built with ROS package as a static grid map at 3.75x3.75 meters with 0.01 resolution.

In the experimental scenario, the mobile robot to pick up a package from the manipulator and deliver it to the warehouse. The robot did not have prior knowledge about the map and where the manipulator and warehouse are. In the experimental result, shown in Fig. 8, the mobile robot started at (130.0000, 100.0000) where the manipulator was at (210.0000, 250.0000). The generated path was a piecewise linear with via points as A(130.0000, 100.0000), B(147.3610, 95.2234), C(177.3497, 130.3414), D(194.8776, 163.3247), E(207.0222, 228.3489), and F(210.0000, 250.0000) consecutively. Now, the mobile robot must be able to walk autonomously along the path. The mobile robot used its wheel's encoders as feedback to stay on the path using odometry. However, the robot cannot rely only on the odometry since precision accuracy was not guaranteed as shown from the actual trajectory (A', B', C', D', E', F').

Therefore, Lidar needed to acquire the absolute position (actual trajectory) despite its processing time from the computation. In this case, odometry and SLAM were used while odometry was computed onboard and SLAM was computed on the edge computer with different update rates. The low latency peer-to-peer communication was the vital key we can shift the heavy computation to the edge.

The edge computer determined the position of the mobile robot as the best guess with computational power. However, since a latency appeared in peer-to-peer communication, the best guess position was updated with the communication delay. The delay significantly degraded the performance of the trajectory controller. To deal with a delay, the timestamp

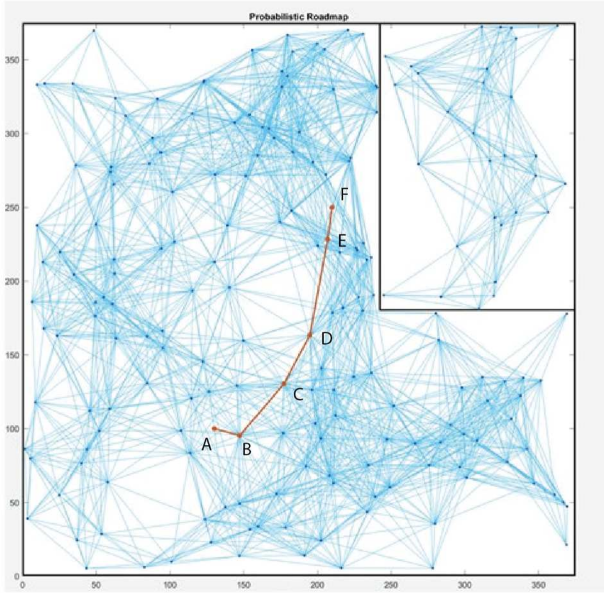


Fig. 7. Generated path from edge computer

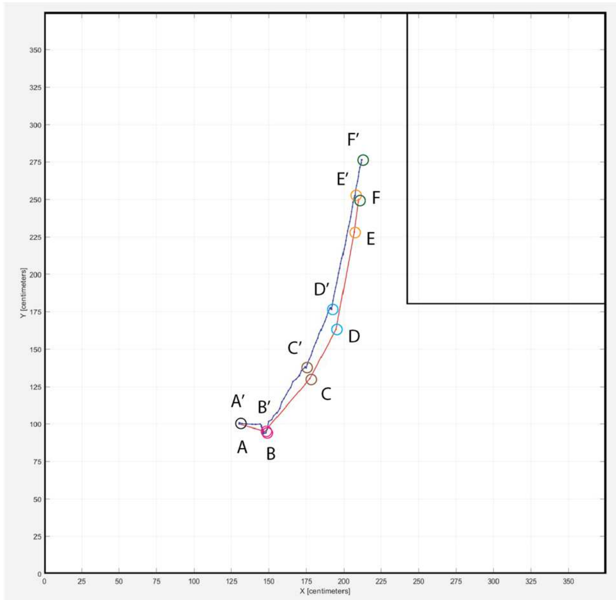
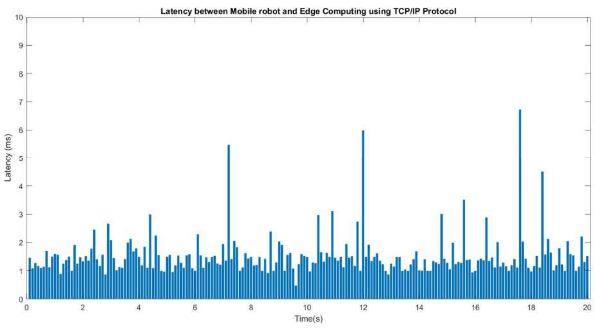


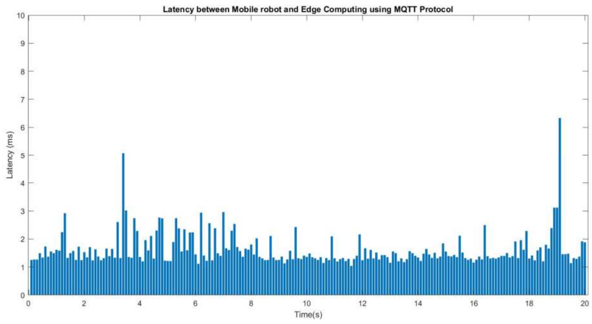
Fig. 8. Comparison between planned path and the mobile robot real path

was sent with the data and extrapolation can be used to estimate the position at the current time. The lower the delay, the better the control performance.

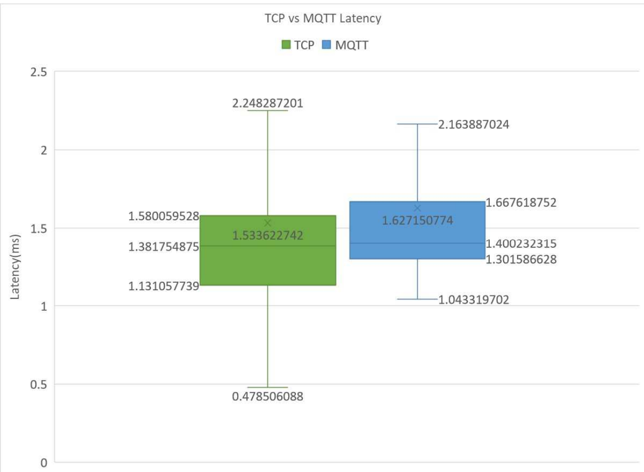
For measuring the delay, our setup of a wireless network would base on Wi-Fi 802.11ax. And the coverage investigation communication protocols were TCP/IP and MQTT. As shown in Fig. 9, the communication latency was sensed for data transmission every 100 milliseconds for 20 second period during the roaming of the mobile robot. It seemed that our communication architecture achieved low latency with an average of 1.5336 and 1.6272 milliseconds for the TCP and MQTT protocol, respectively. However, the latencies could increase to a maximum of 6.7217 and 6.3338 milliseconds due to the switching of the communication channel. Moreover, it was noted that the typical joint controller expects the feedback signal at 1 millisecond.



a) Latency of TCP/IP Protocol



b) Latency of MQTT Protocol



c) Comparative latency between TCP/IP and MQTT

Fig. 9. The latency between mobile robot and edge computing.

Considered the previous research, which used the combination of infrastructure and ad hoc mesh networks, they reached an average latency of 3.8 milliseconds, and their maximum latency was 11.3 milliseconds [15]. Compared to our system, we got an average latency of ~1.5 and ~1.6 milliseconds, and our maximum latencies were ~6.7 and ~6.3 milliseconds for the use of TCP and MQTT, respectively.

IV. CONCLUSION

We successfully set up a wireless network for an automation cell such that a mobile robot can communicate with other machines and the edge computer during collaboration work. The high bandwidth and low latency wireless network allow the edge computer to perform high computation, instead of using an onboard processor, i.e., SLAM, path generation, image processing in our demonstration case. Furthermore, all the machine shares the same up-to-date map on edge computer. During the demonstration case where the mobile robot is to pick up a package from a manipulator, the mobile robot sent the raw Lidar scan (2873 bytes) to the edge computer to process SLAM algorithm at the 150 milliseconds update rate. The resulting position is then used to correct the position of the mobile robot. The latency in that communication is determined as 1.5 milliseconds. As a result, our proposed peer-to-peer communciation has some prospects for the next generation smart factory.

ACKNOWLEDGMENT

This research is funded by TSRI Fund (CU_FRB640001_01_21_8)

REFERENCES

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, “Introduction to Autonomous Mobile Robots,” 2nd Edition, The MIT Press, London, 2011.

[2] S. T. Pfister and S. Thomas, “Algorithms for Mobile Robot Localization and Mapping, Incorporating Detailed Noise Modeling and Multi-Scale Feature Extraction,” Dissertation(Ph.D.), California Institute of Technology Pasadena, California, April 14, 2006.

[3] T. Tsukiyama, “RFID Based Navigation System for Indoor Mobile Robots,” *IFAC Proceedings Volumes*, vol. 44, issue 1, pp. 1084-1089, 2011.

[4] S. Se, D. Lowe, and J. Little, “Vision-based Mobile Robot Localization And Mapping using Scale-Invariant Features,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2051-2058, 2011.

[5] N. Gryaznov and A. Lopota. “Computer Vision for Mobile On-Ground Robotics,” *Procedia Engineering*, vol. 100, pp. 1376-1380, 2015.

[6] H. Cen and B. K. Singh, “Nonholonomic Wheeled Mobile Robot Trajectory Tracking Control Based on Improved Sliding Mode Variable Structure,” *Wireless Communications and Mobile Computing*, pp. 1-9, 2021.

[7] T. Zhang and G. Liu, “Remote tracking of nonholonomic mobile robots via networked predictive control,” in *2016 IEEE International Conference on Mechatronics and Automation*, pp. 84-89, 2016.

[8] A. F. Rochim, B. Harijadi, Y. P. Purbanugraha, S. Fuad, and K. A. Nugroho, “Performance comparision of wireless protocol IEEE 802.11ax vs 802.11ac,” in *International Conference on Smart Technology and Applications (ICoSTA)*, pp. 1-5, 2020.

[9] L. Ma, X. Wen, L. Wang, Z. Lu, and R. Knopp, "An SDN/NFV based framework for management and deployment of service based 5G core network," *China Communications*, vol. 15, no. 10, pp. 86-98, Oct. 2018.

[10] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing:Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[11] Z. Fan, W. Chen, G. Zhu, Y. You, F. Deng, Y. Hou, W. Liang, R. Fu, J Xin, J. Chen, and H. Wang, “Collaborative Robot Transport System Based on Edge Computing,” in *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 1320-1326, 2019.

[12] S. Kamamura, T. Miyamura, and K. Shiomoto, “IP Fast Reroute Control using Centralized Control Plane Architecture,” in *International Conference on Network and Service Management –* CNSM, pp. 274-277, 2010.

[13] J. Ha and N. Park, “Unified Control Architecture for 5G Convergence Network,” in *International Conference on Circuits, Devices and Systems (ICCDS)*, pp. 226-230, 2017.

[14] K. Stangherlin, R. C. Filho, W. lautenschlager, V. Guadagnin, L. Balbinot, R. Balbinot, and V. Roesler, "One-way delay measurement in wired and wireless mobile full-mesh networks," in *2011 IEEE Wireless Communications and Networking Conference*, pp. 1044-1049, 2011.

[15] J. Haxhibeqiri, E. A. Jarchlo, I. Moerman, and J. Hoebeke, “Flexible Wi-Fi Communication among Mobile Robots in Indoor Industrial Environments,” *Mobile Information Systems*, 2018.