

Real Performance Evaluation On MQTT and COAP Protocol in Ubiquitous Network Robot Platform (UNR-PF) for Disaster Multi-robot Communication

Muhammad Ikrar Yamin¹, Son Kuswadi² and Sritrusta Sukaridhoto³

¹ Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia

² Mechatronics Study Program, Mechanical and Energy Engineering Department, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia

³ Multimedia Creative Department, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia.

ahmad.yamin@yahoo.com, {sonk ,dhoto}@pens.ac.id

Abstract

Disaster multi-robot has a significant role in a disaster area to do many tasks like detection of fire, search and rescue of victims, etc. It needs to build good communication between the operator and multi-robot and among multi-robot themselves to perform their tasks quickly and efficiently. This relates with the queue message protocol system. In this research, we implemented the queue message protocol on mesh topology and integrated it on the robot platform. Recently, development of IoT (Internet of Things) Technology causes development of communication protocol. MQTT and CoAP are among the communication protocols used for IoT needs. Both protocols performance were compared when used and implemented into disaster multi-robot. We also integrated MQTT protocol and robot platform python based (UNR-PF). The result shows that MQTT protocol is easier to be implemented on to disaster multi-robot platform (UNR-PF) on mesh topology than CoAP, and that data transfer rate of MQTT protocol has data transfer rate higher than CoAP.

Keywords: Disaster, Mesh, Robot, Communication Protocol, CoAP, MQTT, Internet of Things

1. INTRODUCTION

Indonesia is one of the most disastrous countries. During the period of 5 years between 2010 - 2014, the number of disasters in Indonesia reached 1,907, consisting of 1,124 natural disasters, 626 non-natural disasters and 157 social disasters[1]. The most difficult thing when the disaster happens is to seek and rescue survivors or find the dead victims in the disaster area. Using robot for this case brings several advantages. Robot can move quickly, find victims more accurately than humans, and work in potentially dangerous area.

One of the pioneers in the field of rescue robots for urban disaster is a team led by Dr. Robin Murphy. He led CRASAR (Search and Tracking Help Center Robot) for search and rescue operations at the New York World Trade Center after the terrorist attacks on September 11, 2001[2]. Multi-robot with different capabilities in a single coalition was needed in order to cover the whole disaster area and satisfy the search and rescue task requirements[3]. Multi-robot design with embedded system offers easiness due to its easy operation, light weight, less power, reliability, real-time, and low cost[4]. In addition, effective communication inter-robots will bring successful control and coordination of multi-robot group[5].

There are many factors to perform effective communication among disaster multi-robot. Using wifi is one solution to solve the problem with wire because of its length and inflexibility. There are many wireless technologies used to remotely control the mobile robot like bluetooth, 3G and Wi-Fi. Selection of wireless technologies in physical layer of communication depends on the type of application to be developed considering the following: range, frequency and data rate[6]. In transport layer, an appropriate communication protocol needs to consider those related to battery usage, coverage area, delay, and bandwidth, among others. Multi-agent like multi-robot disaster needs to choose the best network topology and protocol communication based on their condition and environment.

The appropriate wifi that can be used for multi-agent is multicast because using unicast communication may cause difficulties in controlling high data, large bandwidth usage, and data traffic density. Mesh is a suitable topology for multi-agent to solve loss signal in a disaster area[7].

In this paper, we discuss wifi-mesh and MQTT, CoAP as IoT protocol communication for disaster multi-robot with several experiments to compare the performance between MQTT and CoAP in Wifi-Mesh.

2. RELATED WORKS

2.1 Multi-robot For Disaster Purpose

Robot design for disaster areas considers many aspects including field, mechanisms and mobility, obstacles, missions, and more. Some research have been done to overcome these things by making robots with wheels that match the uneven terrain[8], robots that can learn to avoid obstacles with certain algorithms[9], robots that can maneuver by adaptation by changing themselves (see Figure 1)[10], remote-controlled robot equipped with navigation capabilities[11], and multi-robot cooperation to rescue disaster victims[12].

This research use multi-robot which consists of 3 robots with their specializations, that is, 1 robot as the leader and 2 robots as the followers. The multi-robot will do search and simple rescue when finding victims in the experimental field. All robots can detect fire and human based on thermal.

When one or more robots found a fire in a location, they sent signal to the system or operator, then it ordered the robot equipped with fire

extinguisher to approach the location to extinguish the fire. In addition, if they found or detect a human, the robot equipped with water tube would approach the victim and did simple rescue by giving water controlled by the operator.



Figure 1. Adaptive Morphology-based Design of Multi-Loconotion Flying and Crawling Robot: "PENS-FlyCrawl" Multi-robot[10]



Figure 2. (a) Leader; (b) Follower1(Fire Extinguisher); (c) Follower2(Water-Provider)

2.2 IoT Communication Protocol for Disaster Multi-robot

Recently, research on IoT technology is growing rapidly. This technology aims to make all devices around us connect to the Internet so that monitoring purposes and automation can be done. Associating with this technology, we thought about how to implement it on multi-robot. Communication between sensors or devices in IoT will be applied to multi-robot communication. Among the IoT communication protocols that can be used for multi-robot are MQTT and CoAP.

Research about light communication protocols such as MQTT to control the robot and GPS tracking has also been conducted by integrating it on the cloud platform. But the study applies for one robot only[13]. There are some papers that have discussed the difference between CoAP and MQTT and compared the performance between them. As discussed earlier, CoAP and MQTT use different transport layer (UDP vs TCP). A study shows that MQTT is better than CoAP in the communication delay if the loss rate of the network is 20% and CoAP is better than MQTT if the loss rate is 25%[14].

Another study shows that MQTT is faster and more power efficient than HTTP[15], and the other study shows that CoAP is more power efficient

compared to MQTT[16]. MQTT and CoAP also consume low bandwidth[17]. Other research compares communication protocol MQTT-SN and CoAP in the Internet network using LAN ethernet cable for robot. MQTT-SN is a protocol using similar transport layer with CoAP[18]

2.3 Wireless Mesh for Multidisaster Robot

Communication connection between the robots in the disaster area is needed to be maintained and the loss of connection also should be predicted for multi-robot control needs and to make data transmission able to continue. Therefore, mesh network topology management in multi-robot network is necessary. There is a study of wireless mesh network implementation between nodes (laptops) using applications[19].

A Better Approach for Mobile Adhoc Networking (B.A.T.M.A.N.) is a routing protocol for a multi-hop cellular adhoc network that is being developed by the German "Freifunk" community and is intended to replace the state link-OLSR routing protocol. The important points of B.A.T.M.A.N. is the decentralization of knowledge about the best route through the network - no single node has all the data. Kharisma Babu et al.[20] have compared between OLSR and B.A.T.M.A.N. and the result is that B.A.T.M.A.N. is more efficient than OLSR. Davinder et al. [21] have compared B.A.T.M.A.N., DSR, and OLSR protocol performances for 100 nodes based on packet delivery ratio, end to end delay, routing load, and throughput measurement; the result is that B.A.T.M.A.N. is better than the others. Daniel Satier et al. [22] have proved good performance of B.A.T.M.A.N. in a real experiment.

2.4 Ubiquitous Network Robot Platform (UNR-PF)

The revolution in the field of ubiquitous computing becomes a strong foundation for development of robotics technology, known as ubiquitous robots (ubirobots). By combining stand-alone robot like mobile robot with artificial intelligence and web technology, it will become the ubirobot. To establish autonomous and self-evolving ubirobots, it requires an adaptation to ubiquitous computing middleware, or it needs to build a completely new one, so that it is able to consider the specific nature of the robot and its possible interaction. Ubiquitous Network Robot Platform (UNR-PF) is an interesting example of a specially designed middleware for ubirobots applications. This middleware consists of a platform layer: a local platform for robotics systems in single and global regions for robotics systems with lots of areas declared with a local number.

Functionality of the UNR-PF was evaluated through demonstration experiments on six service cases: a remote listening support service, a community formation service, a healthcare service, a shopping support service, an attracting customer service, and a touring support service and a shopping mall[19]. Figure 3 shows the overview of an UNR-PF and its parts.

The UNR-PF is consisted of three layers: service application layer, UNR-PF layer, and robot component layer. Service application layer consists of

services that are provided depend on user needs. The UNR-PF layer is designed to connect between application and robot component layer by seeing databases of user, robots and environments. Local platform in UNR-PF is a platform for multi-robot in single area. Global Platform is group of local platform that covered multiple areas.

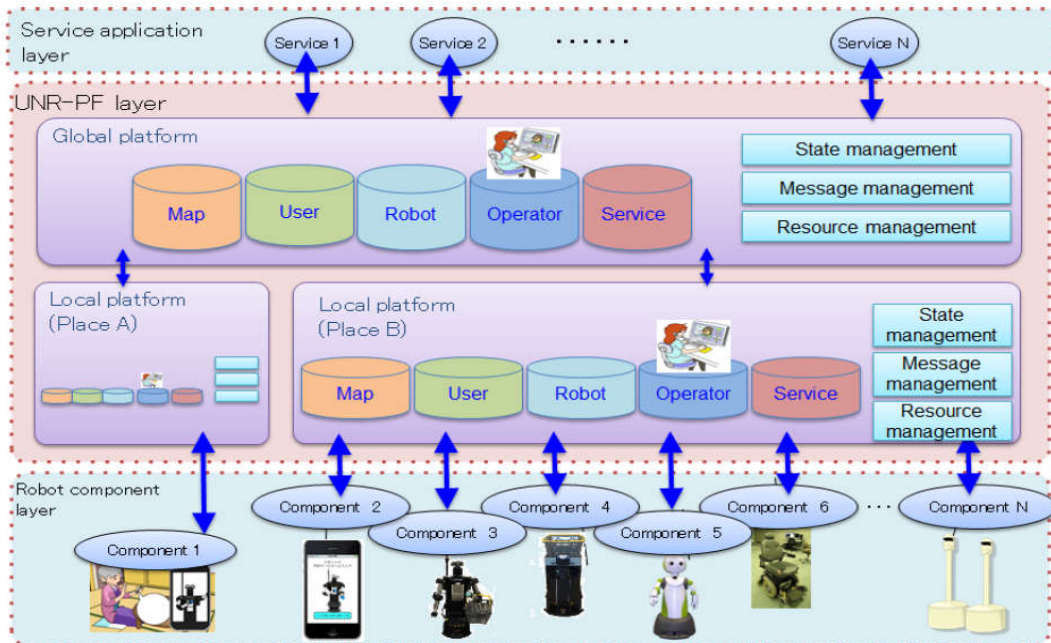


Figure 3. Ubiquitous network robot platform (UNR-PF) for elderly people in Japan[23]

In this research, we designed the robot platform (UNR-PF) that was integrated with IoT protocol. Figure 4 shows our UNR-PF's proposed in this research. There are 3 layers for the platform. The first layer is services provided by the multi-robot. These services are multi-robots tasks like building formation between multi-robot, fire detection in disaster area, human detection, etc. The second layer is the robot platform such as GUI to control and monitor the multi-robot. This platform uses a webserver with Tornado framework (Python based) so that it can be accessed anywhere-and anytime and make it can be accessed by many devices like smartphone, videotron, etc. Python's Framework is used because light and powerful to build web application.

This framework is integrated with MJPEG Streamer to get pictures from the camera in multi-robot then would be shown in the webserver. To make webserver has ability to view data and picture real time, we used websocket technology.

And the third layer of the UNR-PF is robot component layer that consists of operator or user, and the multi-robot. The operator who registered would have fully access to control the multi-robot. As we know,

disaster area is unknown field and sometimes unpredictable so user or operator still needed when multirobot face many problem in the site.

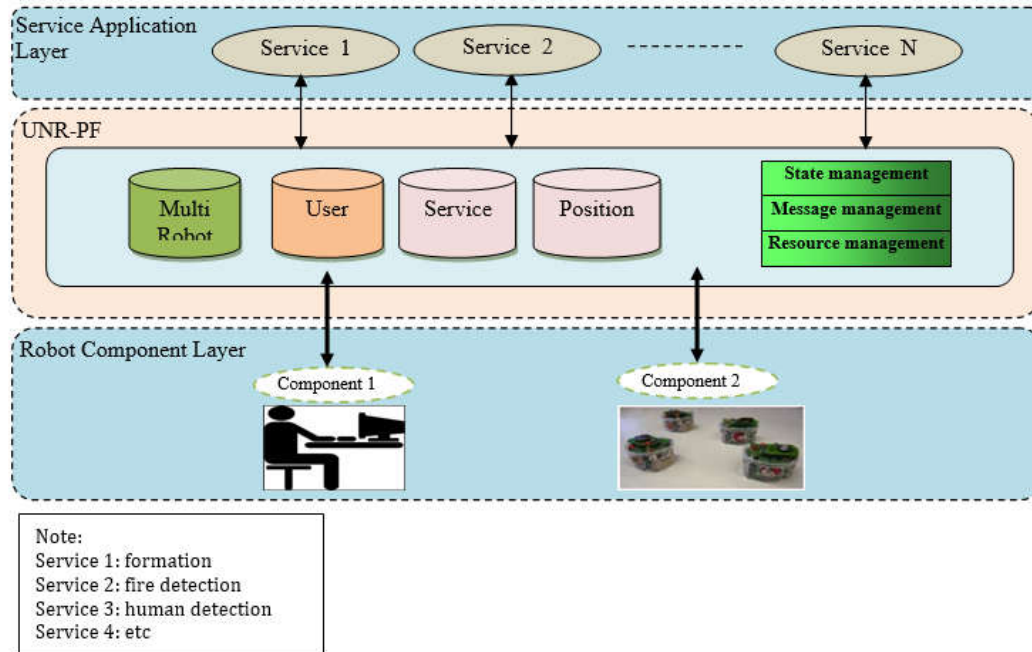


Figure 4. UNR-PF for disaster multi-robot

3. ORIGINALITY

The contribution of this research is about the implementation of IoT communication protocols with wireless mesh for robot disaster on top of robot platform. The experiment for this research was about comparing performances between MQTT and CoAP protocol for disaster robot.

The experiment used a 3m x 3m indoor space. There were obstacles about 1-10 cm placed at random and a target of fire from a candle.

4. SYSTEM DESIGN

4.1 Hardware Setup

In this research, we used multi-robot consisting of 3 robots, in which each had their own special task.

Figure 5 shows all robots used in this research based on special mechanism to complete their tasks. Every robot has arm to fulfill its special task like monitoring, searching and giving first aid to the victim. This multi-robot is semi-autonomous controller. The Multirobot has behaviour based control to do routine task and operator can control the multi-robot remotely from groundstation when it needed.

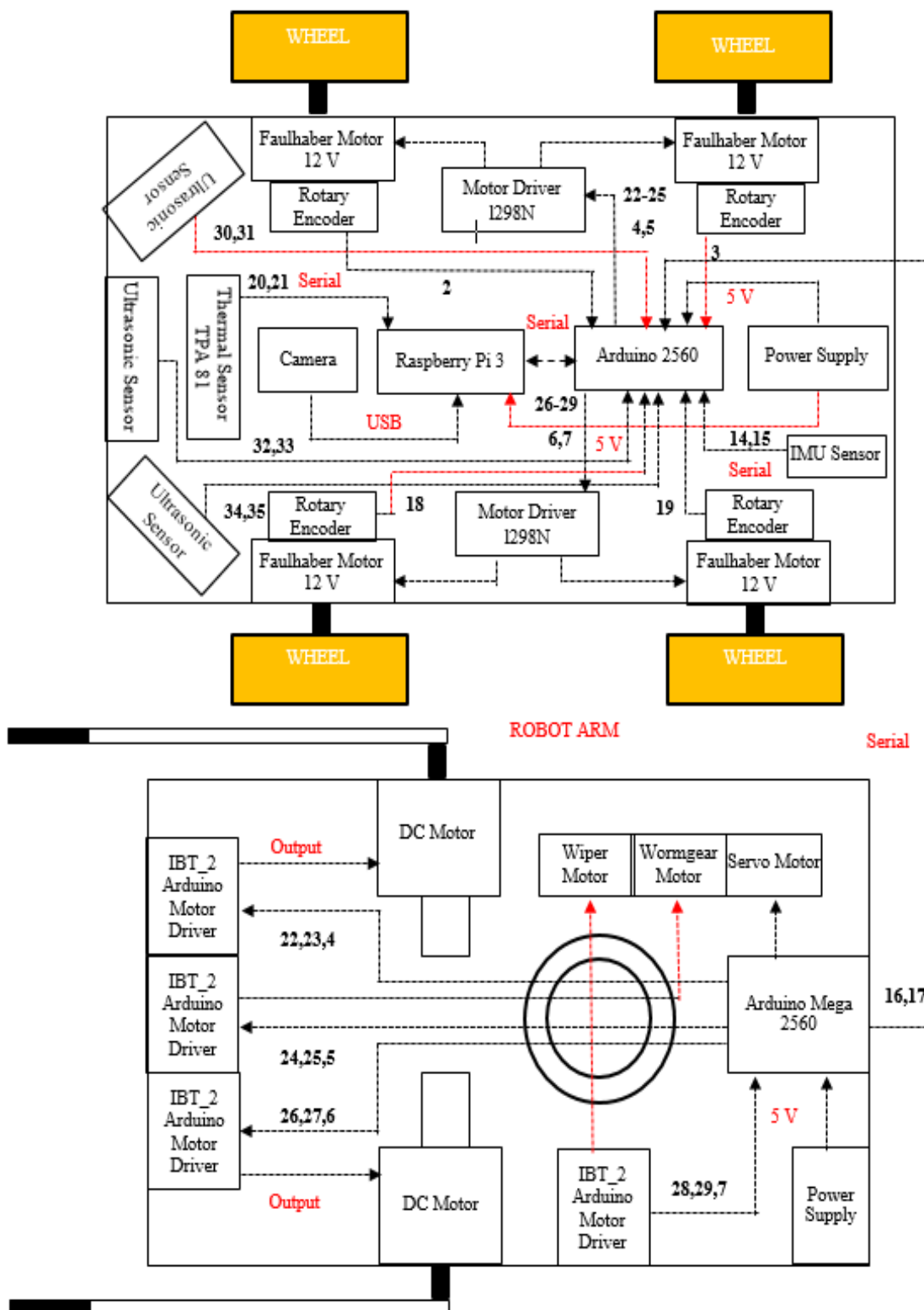


Figure 5. Robots Hardware

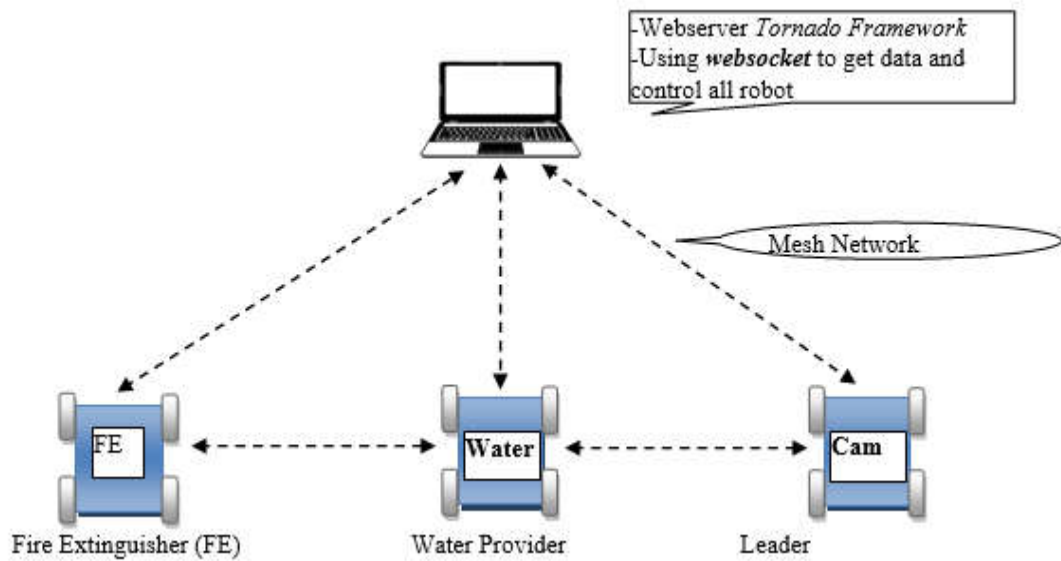


Figure 6. Communication Structure among the robots

Figure 6 shows communication structure for the disaster multirobot. The communication network was build on mesh topology. Every robot and server can send and receive data each other real time. In the server webserver was installed to build Ubiquitous Network Robot Platform (UNR-PF).

4.2 Software Setup

4.2.1 Robot's Platform

We made a disaster multi-robot platform for controlling and monitoring the area around the multi-robot. Figure 7 shows a login page for accessing the web for those who had the privileges to use the web.

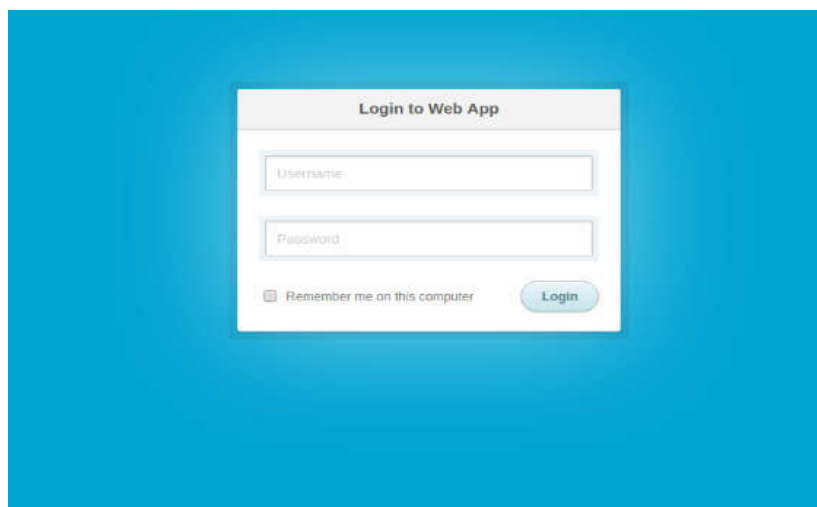


Figure 7. Web's Login Page

Figure 8 shows the main page of the web. We were able to monitor the environment around the disaster area and also datas from IMU sensor and temperature sensor installed in robot real time.

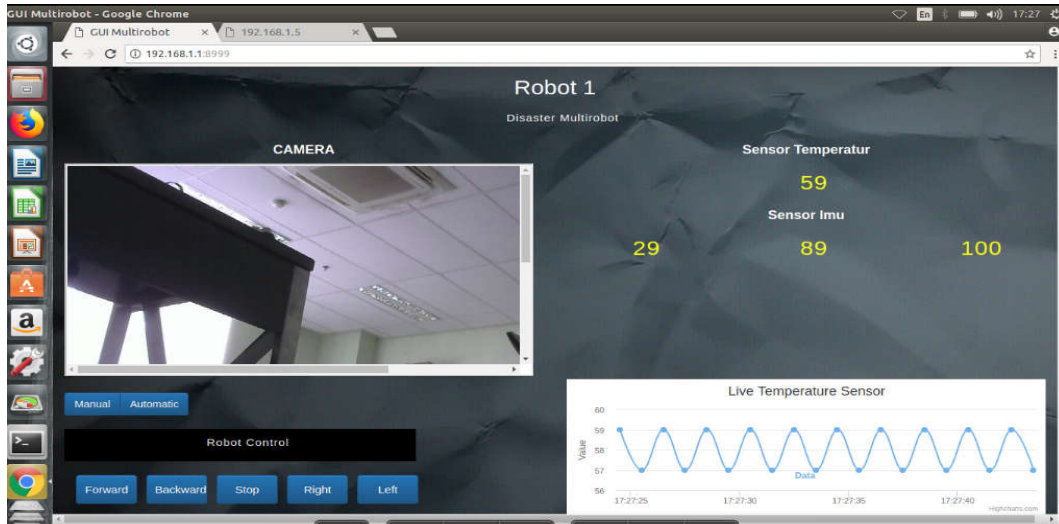


Figure 8. Python Based Web Page For monitoring and controlling disaster multi-robot

4.2.2 Communication's Platform

We provided network topology for the multi-robot needs. We used Batman-adv 2015 application to make the topology, then we designed the communication system as in Figure 9.

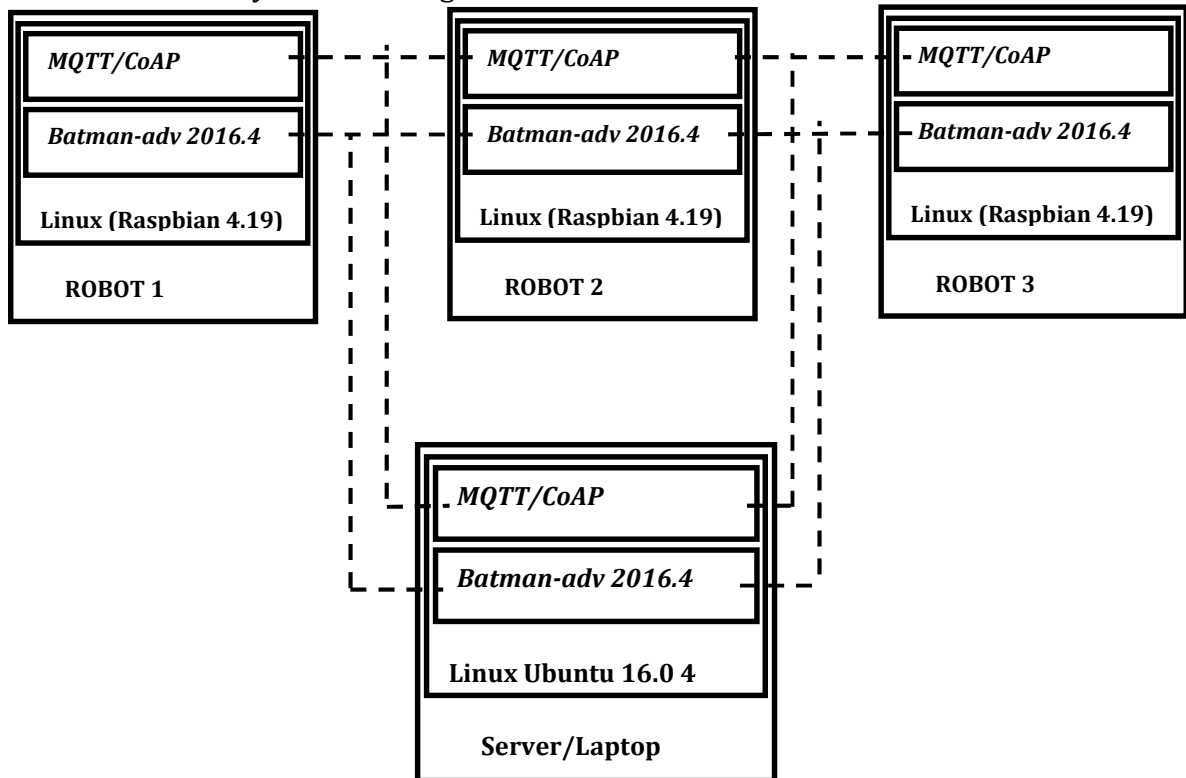


Figure 9. Software communication design for robots

Figure 9 shows that raspberries in all robots were installed linux as Operating System(OS). Batman-Adv is routing protocol linux based for building mesh network. Pahoo MQTT Python client from python library used for MQTT protocol and Coapthon from python library used for CoAP protocol. Communication protocol was needed to make coordination between robots and to control all robots by the operator. Furthermore, this communication was useful for data traffic from multi-robot that sent to the server. This system worked in real time and there was less packet lost received by the robot. In this experiment MQTT and COAP protocol were used to see the performance of both of them.

5. EXPERIMENT AND ANALYSIS

Server for this experiment was Lenovo G40-30 Laptop, which has Intel Celeron N2840 processor. Each robot was equipped with raspberry pi 3 Model B V1.2 that equipped wifi module on board to support wifi mesh network. The server or laptop would send data to all 3 robots and the performance of communications protocols could be monitored by using Wireshark. The experiment environment is summarized in Table 1.

Table 1. Experiment environment

Parameter	Value
Wireless	802.11n
Linux (Robot)	Raspbian 4.19
Linux (Server/PC)	Ubuntu 16.04 LTS
Batman-Adv	2016.4 version
Wireshark (Server/PC)	2..9.0 version
Tcpdump (Raspberry)	4.9.2
Paho-mqtt	1.3.1
CoAPthon	4.0.2

The system needed to be configured to see the performance of MQTT on Wifi-Mesh network. Figure 10-12 show the design of the system for MQTT Protocol. In this research, the laptop was used as a broker and a publisher to make the system easier. The webserver as GUI (Graphical User Interface) showed data received from the multi-robot and controlled the movement of multi-robot. All systems were set with Python-based.

The laptop sent 988 byte string data to all robots for 1 hour. This experiment was carried out in stages by sending data to 1 robot, then to 2 robots, and finally to 3 robots. All robot as subscriber that receive data from publisher. In MQTT protocol, determining a topic is needed for data communication between publisher and subscriber. In this research one topic was used because all robots received the same data. The topic in this case was "*robotmonitoring*". Tcpdump software was installed in all robot to find out data flow that received.

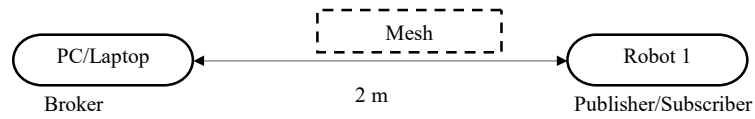


Figure 10. MQTT design for 1 robot

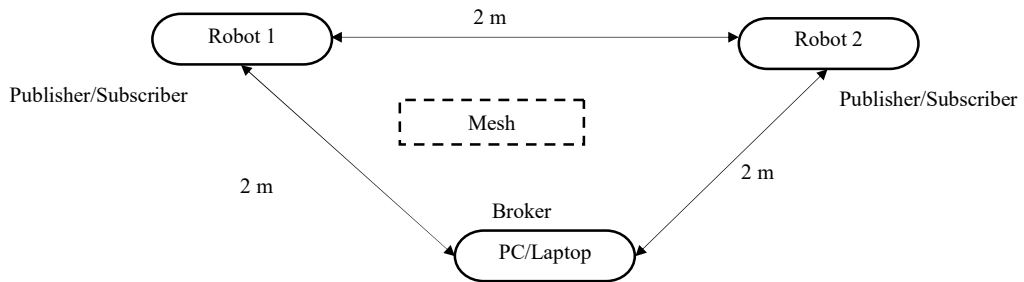


Figure 11. MQTT design for 2 robots

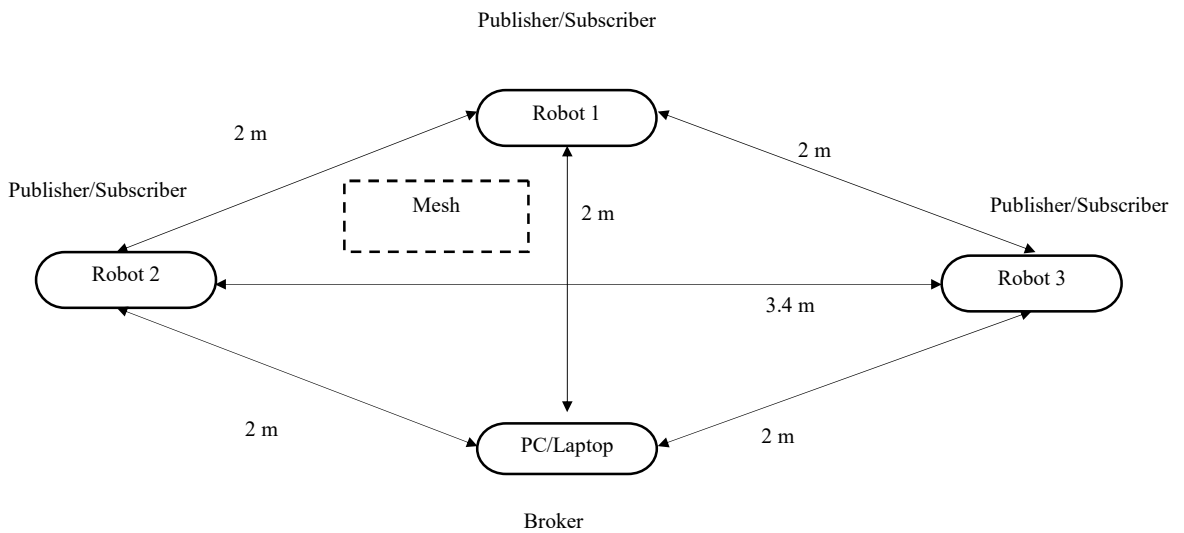


Figure 12. MQTT design for 3 robots

The combination of CoAP and mesh-wifi did not require a special configuration. Figure 13-15 show CoAP configuration as the communication protocol in disaster multi-robot.

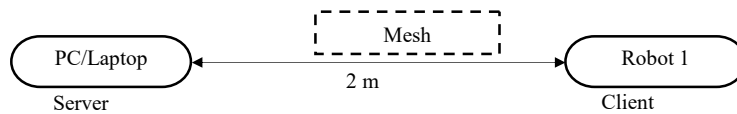


Figure 13. CoAP design for 1 robot

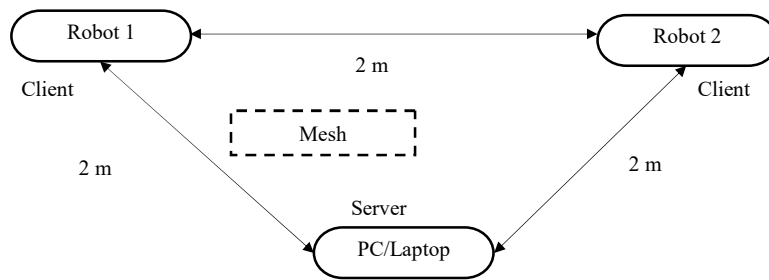


Figure 14. CoAP design for 2 robots

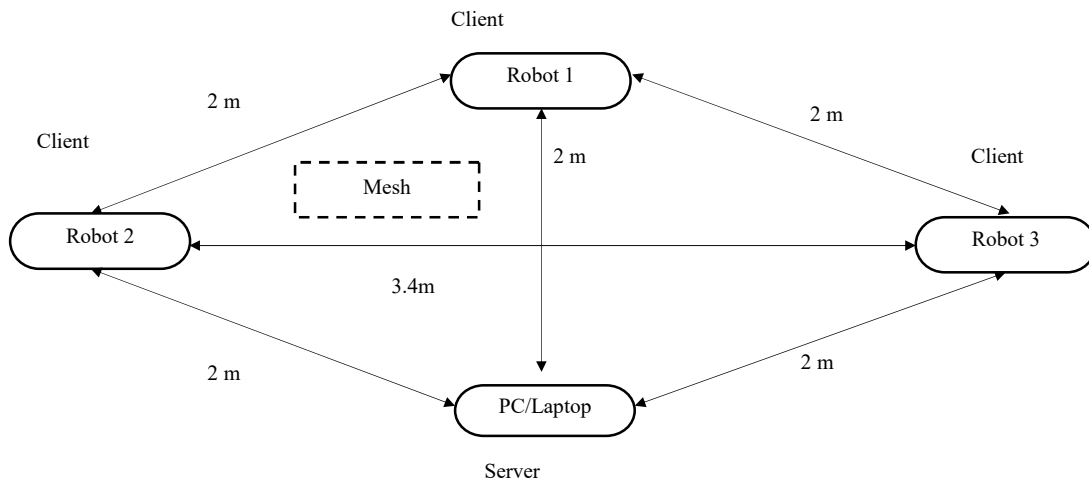


Figure 15. CoAP design for 3 robots

PC and all robots were installed with CoAPthon. PC set to be the server and all robot as clients. A laptop/PC sent 988 byte string data for all robots within 1 hour continuously which was like the previous experiment with MQTT protocol.

Then, a performance test was conducted to see the behavior generated by the system on the MQTT and CoAP communication protocols. This test aims to find out how much CPU capacity is used in the server when sending data continuously to 3 robots in a different set distance from the server controlled by the operator. This test needs an application that runs on the Wheezy Raspbian terminal called SAR (System Activity Report), an application used to know CPU's performance.

6. RESULT

The result of experiment for 1, 2 and 3 robots was plotted in one graphic for MQTT and CoAP protocol respectively. Figure 16 shows MQTT and CoAP performance in this experiment.

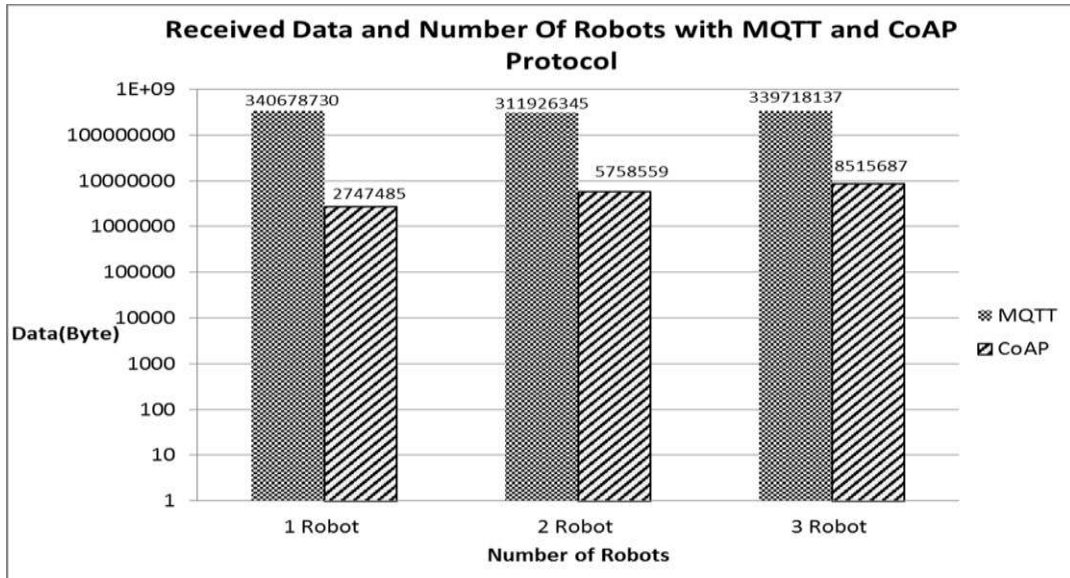


Figure 16. Received data in multi-robot with MQTT and CoAP protocol

Figure 16 shows that for MQTT protocol, with any numbers of robots, all receive nearly the same amount of data because MQTT works on the TCP /IP layer where it cannot broadcast data in parallel to all clients. For CoAP, the total received data is greater for more number of robots. However, the total data capacity received using the CoAP protocol is fewer than using MQTT. Therefore, for UNR-PF system it would be better suited to use MQTT protocol than CoAP.

Comparison on error packet data between MQTT and CoAP is shown in Figure 17. CoAP had not error data because CoAP's data transfer rate was slow and the distance still in CoAP's coverage area.

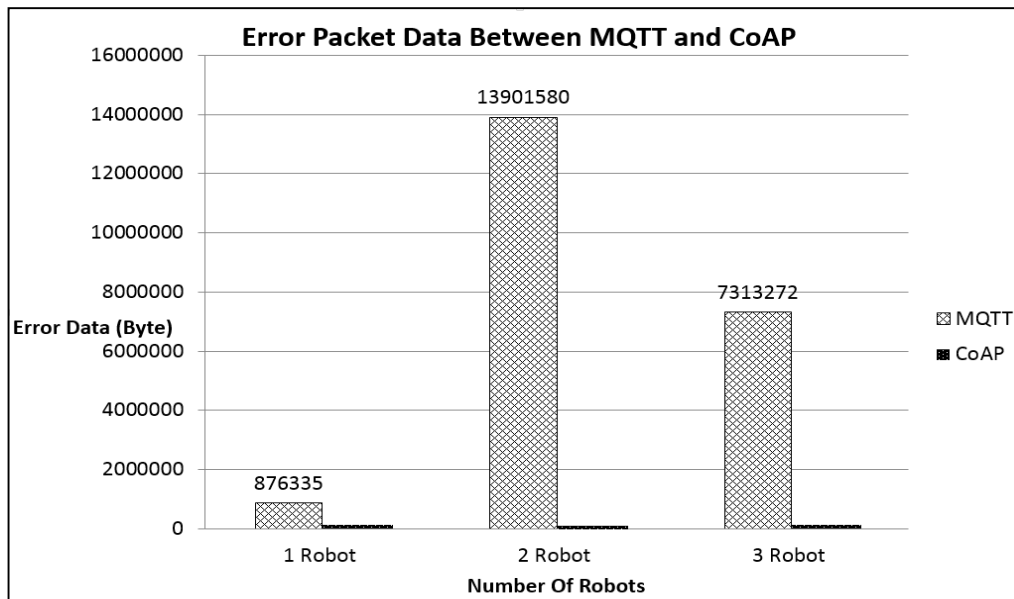


Figure 17. Error Packet Data in MQTT and CoAP Protocol

In our experiment, it is easier to use MQTT to send data to the targeted robots which can be fully controlled from the publisher or sender, not from the subscriber or receiver. MQTT uses topics to connect the sender and data receiver. So data communication lines can be arranged easily. CoAP requires an address that needs to be used in GET method on each client to obtain the desired data from server; so, the main role in the acquisition of data is the client. GET method in this CoAP like a request order to server. Then Server will serve the order by sending data to the clients. This method must be prepared before executing in a python script.

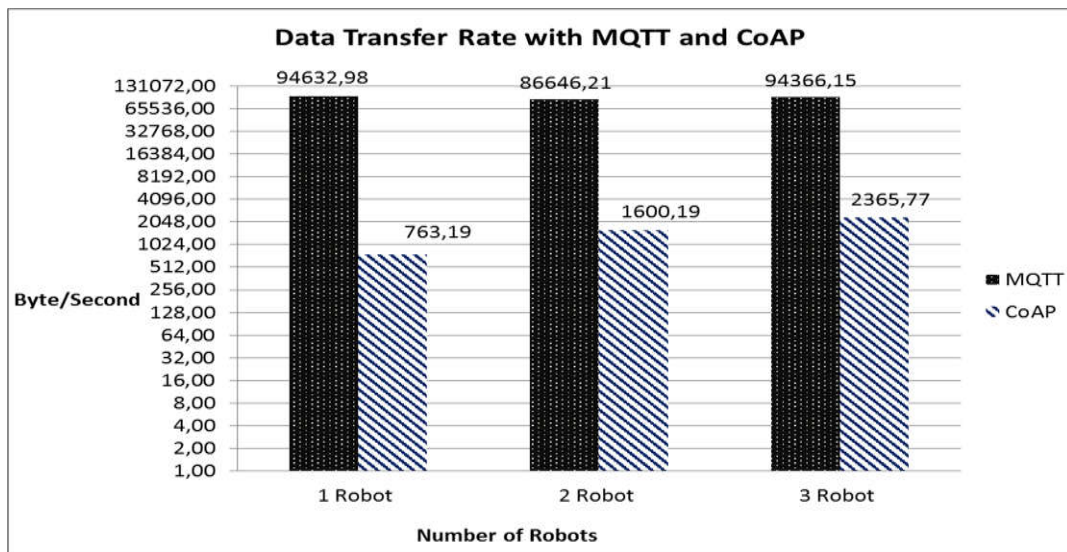


Figure 18. Comparison Data Transfer Rate between MQTT and CoAP

Data transfer rate for MQTT protocol is higher than CoAP. Therefore, multi-robot using MQTT protocol receives more data in 1 hour (based on the experiment) than multi-robot using CoAP protocol. This result leads to the conclusion that the MQTT protocol can be used for real time communication rather than CoAP.

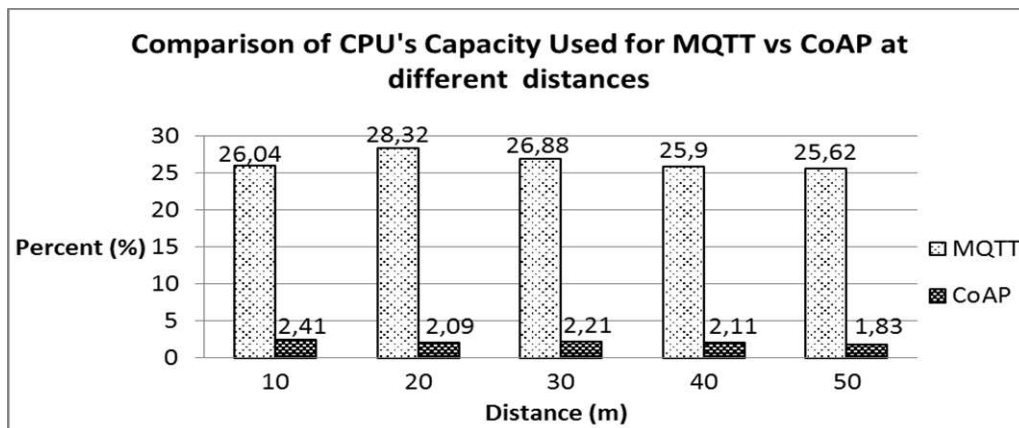


Figure 19. Comparison Of CPU's Capacity Usage in MQTT and CoAP

Figure 19 shows that CoAP protocol need CPU's Capacity Usage fewer than MQTT protocol. The computer's capacity usage can influence the heat on the hardware and the computational processes within it. This result show MQTT protocol has efficiency better than CoAP in using CPU.

6. CONCLUSION

This study has compared MQTT and CoAP communication protocols performance in disaster multi-robot. MQTT Protocol is good for real time communication because it has higher data transfer rate than CoAP in UNR-PF in sending data. Moreover, MQTT is simple to use and integrate in UNR-PF. MQTT protocol works on TCP/IP and CoAP protocol works on UDP in which error correction is not necessary. CoAP protocol has CPU's capacity usage lower than MQTT when it sent data for 1 hour. In the future, the development of communication of disaster multi-robot can be done by adding a variety of coordination to complete many tasks in the network. And using of other routing protocols may also be used as a comparison. Robots sytem can be developed by extending disaster area that consist of many spots.

REFERENCES

- [1] E. Mediastianto, "STATISTIK KEJADIAN BENCANA TAHUN 2014," 2015. [Online]. Available: <http://www.penanggulangankrisis.depkes.go.id/statistik-kejadian-bencana-tahun-2014>. [Accessed: 11-Feb-2016].
- [2] R. R. Murphy, "Trial by fire [rescue robots]," *IEEE Robotics & Automation Magazine*, vol. 11, no. 3, pp. 50–61, 2004.
- [3] H. Kitano, "RoboCup Rescue: A grand challenge for multi-agent systems," in *Proceedings - 4th International Conference on MultiAgent Systems, ICMAS 2000*, 2000, pp. 5–12.
- [4] J. Liu, H. Zhang, B. Fan, G. Wang, and J. Wu, "A Novel Economical Embedded Multi-mode Intelligent Control System for Powered Wheelchair," *2010 Int. Conf. Comput. Control Ind. Eng.*, pp. 156–159, 2010.
- [5] X. Long, J. Jiang, and K. Xiang, "Towards Multirobot Communication," *2004 IEEE Int. Conf. Robot. Biomimetics*, pp. 307–312, 2004.
- [6] S. Kahar, R. Sulaiman, A. S. Prabuwo, and N. A. Ahmad, "A Review of Wireless Technology Usage for Mobile Robot Controller," vol. 34, no. Icsem, pp. 7–12, 2012.
- [7] M. Li, M. Chen, K. Lu, S. Mao, H. Zhu, and B. Prabhakaran, "Robot swarm communication networks: Architectures, protocols, and applications," in *3rd International Conference on Communications and Networking in China, ChinaCom 2008*, 2008.
- [8] T. Sun, X. Xiang, W. Su, H. Wu, and Y. Song, "A transformable wheel-legged mobile robot: Design, analysis and experiment," *Rob. Auton. Syst.*, vol. 98, pp. 30–41, 2017.

- [9] A. R. A. Besari, R. Zamri, A. S. Prabuwo, and S. Kuswadi, "The study on optimal gait for five-legged robot with reinforcement learning," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5928 LNAI, pp. 1170–1175, 2009.
- [10] S. Kuswadi, M. N. Tamara, D. A. Sahanas, G. I. Islami, and S. Nugroho, "Adaptive morphology-based design of multi-locomotion flying and crawling robot 'PENS-FlyCrawl,'" in *2016 International Conference on Knowledge Creation and Intelligent Computing, KCIC 2016*, 2017, pp. 80–87.
- [11] B. Doroodgar, Y. Liu, and G. Nejat, "A learning-based semi-autonomous controller for robotic exploration of unknown disaster scenes while searching for victims," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2719–2732, 2014.
- [12] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," *1997 8th Int. Conf. Adv. Robot. Proceedings. ICAR'97*, pp. 193–200, 1997.
- [13] N. Aroon, "Study of using MQTT cloud platform for remotely control robot and GPS tracking," *2016 13th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol. ECTI-CON 2016*, 2016.
- [14] D. Thangavel, X. Ma, A. Valera, H. X. Tan, and C. K. Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," *IEEE ISSNIP 2014 - 2014 IEEE 9th Int. Conf. Intell. Sensors, Sens. Networks Inf. Process. Conf. Proc.*, no. April, pp. 21–24, 2014.
- [15] N. Stephen, "Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on Android," 2012. .
- [16] N. De Caro, W. Colitti, K. Steenhaut, G. Mangino, and G. Reali, "Comparison of two lightweight protocols for smartphone-based sensing," in *IEEE SCVT 2013 - Proceedings of 20th IEEE Symposium on Communications and Vehicular Technology in the BeNeLux*, 2013.
- [17] S. Bandyopadhyay and A. Bhattacharyya, "Lightweight Internet protocols for web enablement of sensors using constrained gateway devices," in *2013 International Conference on Computing, Networking and Communications, ICNC 2013*, 2013, pp. 334–340.
- [18] M. H. Amaran, N. A. M. Noh, M. S. Rohmad, and H. Hashim, "A Comparison of Lightweight Communication Protocols in Robotic Applications," *Procedia Comput. Sci.*, vol. 76, no. Iris, pp. 400–405, 2015.
- [19] H. Yuliandoko, S. Sukaridhoto, U. H. Al Rasyid, and N. Funabiki, "Performance of Implementation IBR-DTN and Batman-Adv Routing Protocol in Wireless mesh Networks," *Emit. Int. J. Eng. Technol.*, vol. 3, no. 1, 2015.
- [20] L. M. Cortés-peña, "Wireless Mesh Network Implementation," *Comput. Eng.*, 2007.
- [21] Davinder Singh Sandhu and Sukesha Sharma, "Performance Evaluation of BATMAN, DSR, OLSR Routing Protocols - A Review," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 1, pp. 184–188, 2012.

- [22] D. Seither, A. König, and M. Hollick, "Routing performance of wireless mesh networks: A practical evaluation of BATMAN advanced," in *Proceedings - Conference on Local Computer Networks, LCN, 2011*, pp. 897–904.
- [23] M. Shiomi, K. Kamei, T. Kondo, T. Miyashita, and N. Hagita, "Robotic service coordination for elderly people and caregivers with ubiquitous network robot platform," *Proc. IEEE Work. Adv. Robot. its Soc. Impacts, ARSO*, pp. 57–62, 2013.