An Adaptation of IoT to Improve Parcel Delivery System

Ha Yoon Song
Department of Computer Engineering,
Hongik Unversity, Seoul, Republic of Korea
Email: hayoon@hongik.ac.kr

Hyochang Han
Department of Electronic and Electronical Engineering,
Hongik Unversity, Seoul, Republic of Korea
Email: hhcmiso1@gmail.com

Abstract—Recently, IoT technology has been applied in various field. One of the possible fields of an application is logistics system. In current system, a delivery must go through the designated logistics hub, which doesn't provide shortest distance. Such system costs time and inefficient expenses. In this paper, we propose an enhanced parcel delivery system based on IoT technology for reducing total delivery distance and seeking for much economy. First, we designed a sort of IoT devices which can be attached to parcels. This device has various functionalities including the ability to figure out current delivery route. Second, we addressed some difficulties such as : (i) issues linking IoT device into its platform; (ii) issues for designing IoT devices functionalities. Third, we propose ways to improve the efficiency of IoT based parcel delivery system. From these considerations, our system may improve total economics of parcel delivery system.

I. Introduction

Thanks to the 4th Industrial Revolution, Internet of Things (IoT) technology is expanding and prospering in many industrial sectors. Accordingly, a rising number of domestic and foreign corporate are rushing for their own IoT platform in order to launch new services. In current IoT platform, computing and saving process of data is mostly done in central cloud. However, the centralized IoT system is causing many issues. Centralized IoT system requires a giant central server which processes and saves data that are received from a number of devices. This demands big administrative expense. It is difficult to increase connection of IoT devices continuously, since central server has its limit. To resolve this problem, we should expand the central server, but expanding the central server is very inefficient and will not be recommended. In addition, IoT devices require efficient management of data and central server is very important due to its real time data processing. Therefore, if there is a problem with the central server, every IoT device belongs to the platform will become useless. It is predicted that distributed networking of IoT will take place in the future to solve these problems [1].

In this case, the central server's role will be reduced, and the portion of the work handled by the terminal device will increase. In the future era of IoT, firmware-based IoT devices, which only deal with simple tasks, will be raised to the level of the RTOS (Real Time Operation System) to secure safety and connectivity. Amazon acquired FreeRTOS, the most commonly used RTOS in embedded systems in 2017 [2]. In the same year, Google announced Tensorflow-Lite, a deep learning framework for IoT devices, to compensate for problems with cloud-based AI devices [3]. Based on existing experimental IoT platform provide by SK Telecom (SKT) of Republic of Korea, our parcel delivery enhancement can be implemented. The name of IoT platform is ThingPlug which also provides LoRaWAN as dedicated global network for IoT.

In this paper, we propose a distributed networked IoT delivery system instead of a centralized IoT platform as described in section II. Section III introduces functionalities of delivery dedicated IoT devices. Section IV introduces a process that connects IoT devices to distributed network IoT platform. Section V discusses issues in the process of interworking commercial networks. Section VI describes dedicated IoT devices we implemented, and Section VII concludes and introduces direction of research.

II. IoT Platform

Various factors such as Network, Device, and Application Server are needed to implement in IoT service. IoT platform refers to a service that allows various components to meet and easily combine and helps increasing utility value by connecting each element. As a representative, Qualcomm's AllJoyn, Microsoft's Azure IoT Suite and SKT's ThingPlug provide centralized services [4][5]. In this paper, we use SKT's ThingPlug as an IoT platform. Designed terminal device collects sensor data according to purpose and sends it to ThingPlug network server through gateway. Application Server can import data stored in ThingPlug's Network server and send control commands to the terminal device. These sequences are well expressed in Fig. 1
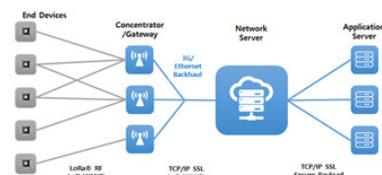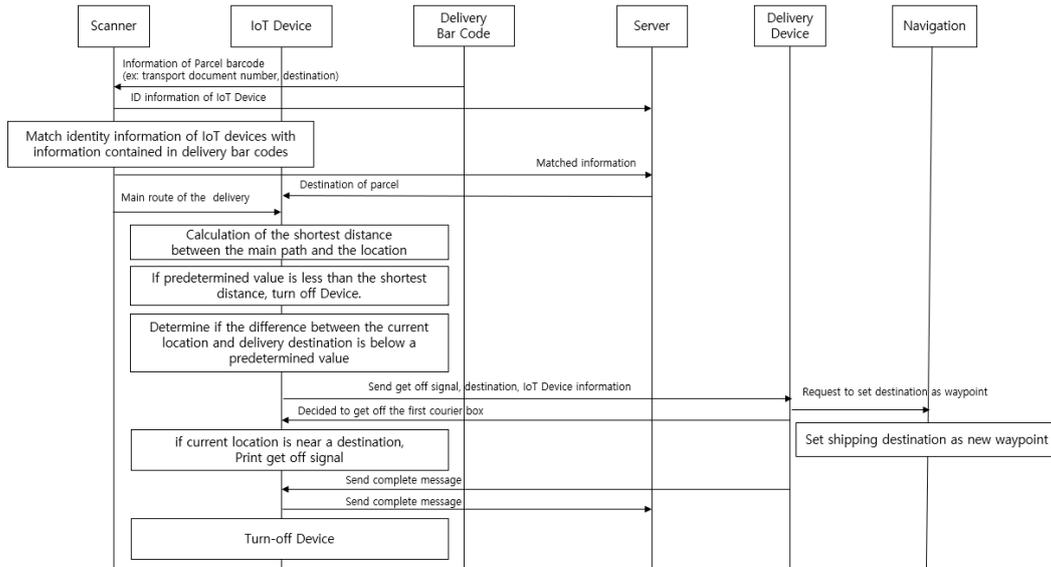

Fig. 1. IoT Platform

Fig. 2. Sequence Diagram of Improved Parcel Delivery

There are three ways to utilize ThingPlug Network Server. First case is to use one terminal device. In this case, we can communicate with ThingPlug Network Server using API provided or with ThingPlug via LoRa module. Second case is to use the terminal devices near the fixed gateway. In this case, we can connect the terminal device and the gateway in a wireless start topology method. Then the gateway collects the sensor from the terminal devices and send it to the ThingPlug server. At this time, the wireless communication between the terminal device and the Gateway can use LoRa RF, and the communication between the gateway and ThingPlug can use Ethernet. The third case is the terminal device that moves a wide range of areas. In this case, it is not possible for a single gateway to cover all the terminal devices, which is moving extensively. Therefore, we use Low Power Wide Area Network (LPWAN) that can communicate with ThingPlug server anywhere. LPWANs include LoRaWAN and Sigfox [6][7][8].

III. IoT DEVICE FUNCTIONALITIES

Operation methods of IoT devices attached to delivery boxes were divided into several stages. There are four phases: the collection phase, GPS phase, the moving phase and the completion phase. We presented Sequence Diagram at Fig. 2

A. Parcel Collection Stage

ID information will be printed through the display module once pick-up of parcels starts with IoT device attached delivery box and IoT device is turned on. After a while LoRa will be connected with server. Then, the scanner will collect the data from the barcode attached on the delivery box. This data will be transmitted to the server and saved in a table form by matching the information, invoice number, and delivery destination of the IoT device. And IoT device get Collecting

path and calculate shortest path with collecting path and parcel destination. If that distance is less than threshold value, then start Parcel in move stage.

B. GPS Data Collection Stage

Once data are received from the geopositioning module of the IoT device, the data are stored for the current location of the IoT device. Usually, GPS is a representative one of geopositioning system. In case the data cannot be received from the GPS module, following steps are necessary. First, you need to get to nearby access point (AP) with Wi-Fi module. Then, obtain the current location information from the MAC address and API of the AP [9]. If Wi-Fi AP is not accessible, receive the current location information from Bluetooth module attached to the IoT device in the carrier's smartphone. If didn't get positioning data, you should be waiting for a certain delay and start from the beginning again.

C. Shortest Path Check Stage

In this stage, we need to know how to check the shortest path in order to add a new destination. At the destination, other parcels can just stop over and move on to next logistics hub. The process is as follows. Once data have been collected through stage B., then begin stage C. and use the data to check the distance between the current location and the destination. In case the distance is shorter than the threshold value, then send the parcel data to ThingPlug server via Lora and send ID info of IoT device to transmitter's smartphone via Bluetooth. Then, add new destination info from IoT device then update logistics path. Lastly, finish the moving step and move onto stage D. In case the distance is longer than the threshold value, loop will be started again after waiting for a certain delay of time.

D. Parcel Deliver Completion Stage

After the stage C. on every fixed time, then check the distance between current location and destination. In case the distance value is very small, IoT device will recognize that the parcel has arrived the destination. IoT device will emit signal by LED or display module. Then the courier can recognize that the box should be unloaded. Moreover, IoT device sends the completion signal to ThingPlug server and the signal will be updated. Then the server will send completion message to the IoT device. IoT device will emit return signal and will be turned off. Afterwards, the original shipper will collect IoT devices.

IV. IoT Platform Interlocking Process

This paper adopted SK Telecom's IoT platform, ThingPlug, which utilizes LoRaWAN installed in South Korea, considering the size of the data and the size of the commercial network transmitted by IoT device. SK Telecom provides commercial network linkage through its officially certified LoRa module. In this paper, LoRa module of Wisol was used [10]. The LoRa module of the Wisol is connected to the Micro Controller Unit (MCU) and the Universal Asynchronous Receiver/Transmitter (UART) to send and receive messages through serial communication. The message sent from MCU to LoRa module follows the Command-Line Interface (CLI) command format defined in the user manual. ThingPlug's LoRa commercial network interworking is accomplished through Open Test Bed (OTB) certification and Quality Assurance (QA) testing.

Looking at the test items required by OTB certification, the first thing to do is to identify the debug message of the LoRa module from the MCU and design and implement it so that CLI command can be sent. During OTB authentication, all debug messages sent by LoRa module must be printed out because the debug message is verified through the customer's Host PC. Second, if you receive the Reset Downlink control command, we must reset the module after five seconds of delay. If IoT device that is on commercial network shows abnormal symptoms of operation, ThingPlug server sends an order to reset IoT device. MCU parses DEBUG message sent from LoRa module and performs Device Reset through CLI command. Since the reset command received from the server is a Confirmed message type, the reset must be performed after waiting about five seconds for LoRa module to receive the command and send the ACK. In addition, we should implement MCU command for perform the following actions: (1) Data Send 65 Bytes. (2)Data Send 66 Bytes. (3) Link Check Request. (4) Device Time Request.

For example, max payload is 65 bytes, so sending 65 bytes as shown in (1) can proceed without error but sending 66 bytes as shown in (2) must be able to check the ERROR debug message. Also, we need to check message type. There are two kind of message confirmed and unconfirmed. Confirmed message need to check packet was received. If the terminal device or Server sends a Confirmed message, they should send unconfirmed message with the ACK to make sure that message was processed. If there are no ACK, they send confirmed message again. if retransmissions happened 8 times, return error and process will end. In addition, codes should be implemented that allow remote modification of the number of retransmissions. Because it is an additional remote-controlled IoT device, functions had to be implemented so that device can perform all tasks when variety of commands were issued from server. To do this, you need a code to verify that various functions work properly.

It was necessary to verify that Frame Count messages, which are larger than or smaller than the existing values, were received from the device in ThingPlug server by correctly parsing the corresponding messages, and that the frame Count was set to be larger than the existing values and then processed the message normally. It was also necessary to check if the message is handled normally when received from the terminal after setting up the Mgmt cmd. In addition, if the same message was received during processing after receiving the message, a code was required to confirm whether the message was dropped, and the uplink message was retransmitted and that the ACK was sent to the server.

There was a condition that the firmware of the LoRa module should be kept up to date to stabilize the IoT device, and that the PCB produced by itself should be used instead of the PCB provided with the IoT device. When communication with ThingPlug is required during the implementation of OTB certification items, open the device to the test network to conduct the experiment. The opening of the test network is completed by submitting the class with Device EUI to SK Telecom's ThingPlug manager and entering the information into LoRa module. You can register the device in your account in ThingPlug portal and send various downlink messages to the device through Open API Test. After OTB certification, SK Telecom is assigned a manager according to IoT service field and provided a test number for commercial network, which can be tested on commercial network in South Korea for three months after subscribing to CCBS. It also carries out commercialization and QA test through the manager's guide according to IoT service that it wants to provide. After passing QA test, connection of IoT device's commercial network will be completed.

V. Issues in the Interworking Process with Commercial Network

In this paper, an esp32-based MCU board with built-in Wi-Fi function and Bluetooth function was designed and implemented. On top of this, GPS module and Wisol LoRa module for GPS tracking are connected by UART. However, there is only one pair of RX, TX pin for UART communication on pin diagrams. Therefore, the software serial replaces the deficient UART pins. Since esp32 does not officially provide the software serial, use the open source of third party. In addition, an additional hardware series is specified in the specification, which is not specified in the pin diagram. The esp32 offers up to three pairs of hardware serial ports.

During OTB test, the staff of the test shall be able to check all debug messages. Generally, the hardware serial buffer in

esp32 can be stored up to 256 bytes. If the LoRa module receives debug messages exceeding 256 bytes, the message is lost. Therefore, the size of the buffer should be increased to 1024 bytes through a member function of the Hardware Serial object to receive and output a complete debug message.

Since IoT devices are attached to delivery boxes, IoT devices must all be operational until delivery is completed after receiving delivery. To do so, battery management of IoT device is essential. For this purpose, the LoRa module OTB test-to-test Class A format was used. Class A format is not always on but is a very efficient method for battery management by storing data in the buffer when ACK comes from the server or when the value is entered from the sensor and performing operation every certain period of time. A typical method cannot store all data, but the same method of increasing the size of the hardware serial buffer in esp32 can store 1 kb of message, and the buffer of that capacity can store enough data, and the ACK of the server is sufficient. Energy management and performance are very important elements of IoT device, both aspects of IoT device performance and energy management can be grasped through this method.

VI. IoT Device

We implemented IoT Device for parcel delivery system as shown in Fig. 3. For basic control functions, MCU board with built-in esp32-based Wi-Fi and Bluetooth function, GPS module for GPS tracking function, and battery and battery charging are attached. The purpose of the design was to make the device as small as possible when manufacturing the device, and to prepare for the impact between delivery and delivery, the impact absorbing rubber was attached to the upper and lower parts of the IoT device, and acrylic plates and devices were firmly fixed using brass supports. To prevent impact damage to the acrylic or PCB plates, washers were added to each joint to enhance stability. In addition, the antenna was attached for desired Lora communication. The battery, which is at greater risk of short-circuit due to the weight during delivery, was fixed on acrylic plates. To secure GPS data, and the GPS Sensor was secured on acrylic plates to enable stable sensor reception. In order to manage the battery of the IoT device when not in use, the module can be switched on and off using the On - OFF switch. After checking whether the battery functions properly during delivery, we confirmed that 80 hours of continuous operation is practically possible. Although it has been suggested that the protruding part of the
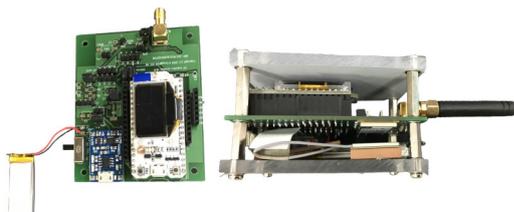

Fig. 3.   Device Image

antenna will be a problem when using the IoT device seal, it is actually attached directly to the parcel when collecting data and can be used reliably when attaching it toward the sky when attached

VII. Conclusion

In this paper, we proposed an enhanced parcel delivery system with IoT technology for reducing total delivery distance and for much economy. We showed the developing process and implementation to experimental IoT platform. Previously, central cloud did the most of control processing and computation. However, it is possible to divide up the processing to IoT device by using Bluetooth and Wi-Fi features. Then it became possible to process the data without going through server due to various functionalities of IoT device. In addition, the corresponding module can be deemed to be highly useful, such as by changing it to the delivery system or the tracking system. By using location information generated from these systems, it can be used as an active test data set in hub-related papers to prove the efficiency of the Hub and spoke method currently used and could check other additional logistics delivery methods [11][12]. Additionally, it would also devise and utilize a method for the security and integrity of the logistics system by combining blockchain with IoT platform and our IoT device functionality [13]. More functionalities can be considered; we can also add functions to identify the weight, type, size, and delivery area of the product entered before delivery and to show how to increase space efficiency and speed of delivery during loading parcels.

References

[1] S. Tomovic, K. Yoshigoe, I. Maljevic, and I. Radusinovic, "Software-defined fog network architecture for iot," *Wireless Personal Communications*, vol. 92, no. 1, Jan 2017. doi: 10.1007/s11277-016-3845-0

[2] R. Barry, "Freertos," *https://www.freertos.org/*, 2017.

[3] Google, "Tensorflow-lite," *https://www.tensorflow.org/lite/guide*, 2018.

[4] Microsoft, "Azure iot suite," *https://azure.microsoft.com*, 2018.

[5] SKTelecom, "Thingplug," *https://sandbox.sktiot.com/*, 2018.

[6] L. Alliance, "Lorawan spec 1.0.2," *https://lora-alliance.org/resource-hub/lorawanr-specification-v102*, 2016.

[7] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of lora: Long range and amp; low power networks for the internet of things," *Sensors*, vol. 16, no. 9, 2016. doi: 10.3390/s16091466

[8] T. M. W. 1.0, "A technical overview of lora® and lorawan™," *https://www.everythingrf.com/whitepapers/details/2682-a-technical-overview-of-lora-and-lorawan*, 2015.

[9] Sheng-Cheng Yeh, Wu-Hsiao Hsu, Ming-Yang Su, Ching-Hui Chen, and Ko-Hung Liu, "A study on outdoor positioning technology using gps and wifi networks," in *2009 International Conference on Networking, Sensing and Control*, March 2009. doi: 10.1109/ICNSC.2009.4919345

[10] Wisol, "Lom102a user manual," *http://lora-support.wisol.co.kr/*, 2017.

[11] G. F. George Deltas, Klaus Desmet, "Hub-and-spoke free trade areas: theory and evidence from israel," *Canadian Journal of Economics C.R.D.E., Universite de Montreal P.O. Box 6128, Station Centre-Ville Montreal, Quebec, H3C 3J7 Canada*, 13 August 2012. doi: https://doi.org/10.1111/j.1540-5982.2012.01722.x

[12] G. G. Das and S. Andriamananjara, "Hub-and-spokes free trade agreements in the presence of technology spillovers: An application to the western hemisphere," *Review of World Economics*, Apr 2006. doi: https://doi.org/10.1007/s10290-006-0056-x

[13] S. Huh, S. Cho, and S. Kim, "Managing iot devices using blockchain platform," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, Feb 2017. doi: https://doi.org/10.23919/icact.2017.7890132