International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.9 (2016), pp.221-230
http://dx.doi.org/10.14257/ijmue.2016.11.9.23

A Design on the Communication Protocol for Data Collection of the IOT Based on Cold Chain System

Dae-Sik Ko

Mokwon University, Korea
kds@mokwon.ac.kr

Abstract

This paper designed and analyzed the communication protocol for the data communication between the IOT device and the middleware server of the IOT based Cold Chain Monitoring System. In this regard, we have referenced the CoAP Protocol Model of IETF CoRE WG, and analyzed and designed the process status flow, data flow and frame format of the IOT device (IOTD) and the middleware server. In order to test the effectiveness of the proposed Cold Chain Communication Protocol, we have produced the Zigbee, Bluetooth and wifi module mounted gateway and the temperature, humidity sensor. After sending the temperature data from the IOTD to the middleware server, we analyzed the received data from the middleware server. As the result, we were able to interpretate the data sent from the IOTD, and identify that in using these data the cold chain data such as the temperature, humidity, and battery remains data can be collected in real time.

***Keywords**: cold chain, protocol design, temperature management, IOT, data frame design*

1. Introduction

With the recent development of the signal processing technology, miniaturization of various sensors, and increase of needs for consumer-driven data services, IOT services that can be monitored anytime, anywhere are expanding. However, even with the communication with various sensors, there are many difficulties in data sharing due to the unstandardized sensor data and communication protocol [1-2].

The cold chain monitoring system refers to a system that controls the state of control by collecting the real time data of agriculture and fishery products and medicines [1-2] In order to provide IOT cold chain framework based services, continuous communication between the control server and IOT device is necessary. In servers and clients that use the cold chain framework, in other words, the communication protocol model between the middleware server and the IOT sensor device, reliability is also important due to the characteristics of sensor signals in IOT environments, but it has the characteristic of requiring the collection of continuously changing measurements from numerous sensor nodes. Therefore, this paper has referenced the CoAP protocol model of IETF CoRE WG, which is low in protocol processing load, and excellent in the compatibility of accommodating various sensors [3].

The CoAP protocol model of IETF CoRE WG needs to satisfy several requirements for the interoperability guarantee. In the CoAP protocol, we attempted to secure the simplicity of the protocol by dividing the message transmission processes of messages that require and do not require reliability in the message transmitting and receiving process. In cases that require reliability, data is sent from the Client to the Server, and a confirmation process on the data reception was placed in the server. In cases that do not require reliability, data is sent by the client from the server, and do not require the server to perform a confirmation process on the data reception [3-4].

ISSN: 1975-0080 IJMUE

This paper designed and analyzed the communication protocol for the data communication between the IOT device and the middleware server of the IOT based Cold Chain Monitoring System. In this regard, we have referenced the CoAP Protocol Model of IETF CoRE WG, and analyzed and designed the process status flow, data flow and frame format of the IOTD and the middleware server.

Lastly, In order to test the effectiveness of the proposed Cold Chain Communication Protocol, we have produced the gateway and the temperature, humidity sensor. After sending the temperature data from the IOTD to the middleware server, we analyzed the received data from the middleware server.

2. Design of the IOT Based Cold Chain Communication Protocol

2.1. Communication Protocol between the IOTD and the Middleware Server

A. Connection Type Design

The communication connection type between the cold chain IOTD and the middleware server can be divided into a 1-step connection type (2 Tier Type) where the IOTD is directly connected to the middleware server, and a 2-step connection type (3 Tier Type) where a router, gateway, or a coordinator acts as a intermediate between the IOTD and the middleware server. As such, even when the connection types are categorized differently, a communication model should only define the interface standard between the IOTD and the middleware server. Also, a router, gateway, or a coordinating device should deliver data transparently without playing any role between the IOTD and the middleware server, and indicate the distinct ID of the specific device [5-7].

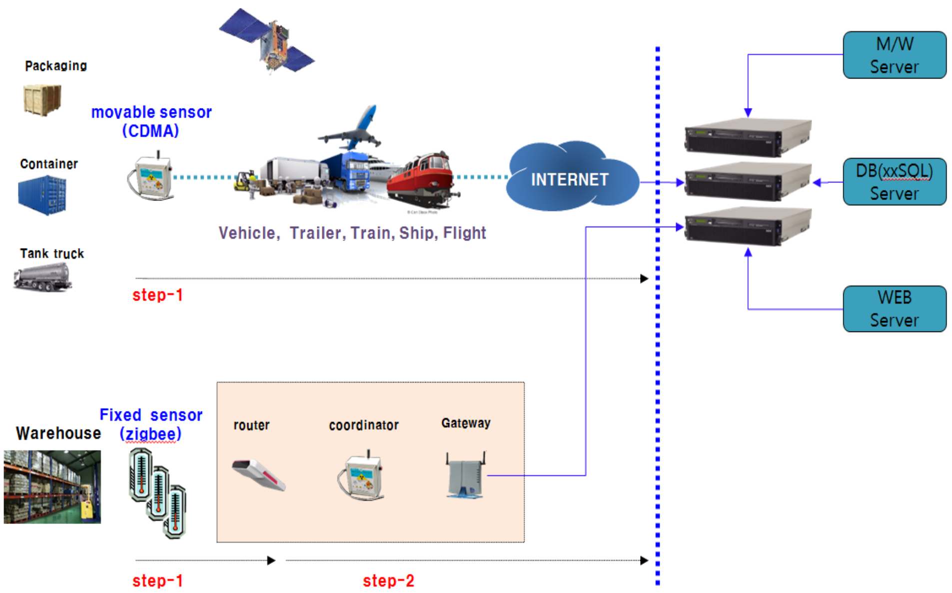

Figure 1. Connection types for Communication between IOTD and Middleware Server (Upper: 2-Tier Type), (Lower: 3-Tier Type)

B. IOTC and Process Status Flow of Middleware Server

The communication between the IOTD and the middleware server operates as a client/server model. At this time the middleware server acts a server, the IOTD as a client, and as one part of the IOTD, a gateway in a connection model is simply regarded as a part of the IOTD. IOTD and the middleware server uses the TCP Socket Communication of TCP/IP for communication. On the protocol process, middleware server multiplexes the process to process the data sent by multiple IOTDs without loss, and these processes performs their function using the inter process communication. Although IOTD uses a simple TCP/IP protocol, it has a n:1 communication structure with the communication model(CM) of MWS because of the large number of sensor types and numbers [8-9].

C. IOTD and MWS Communication Protocol Design

The sensor data flow between the IOT device (IOTD) and the middleware server (MWS) is as represented in Figure 2.

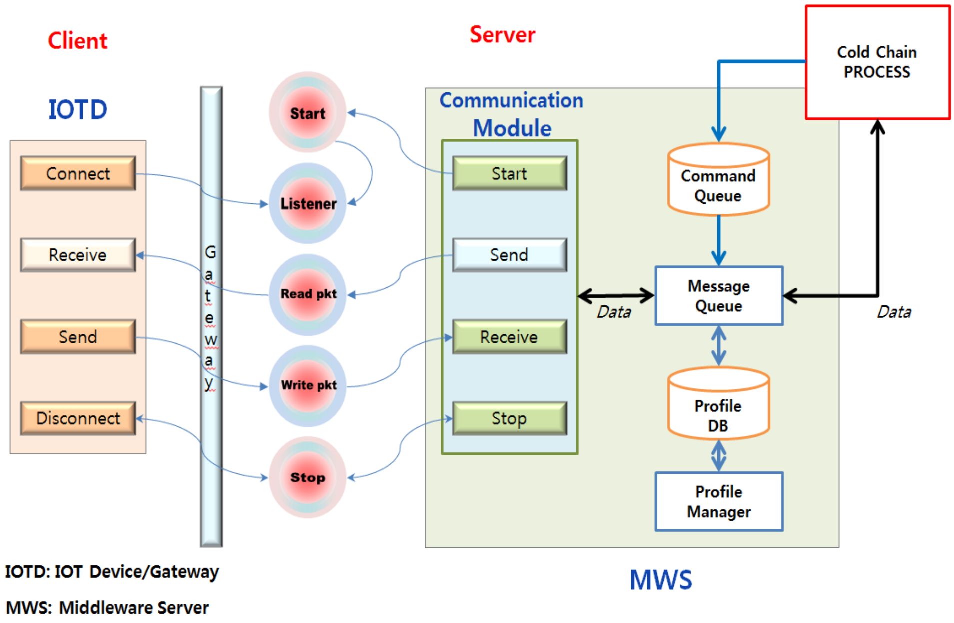

Figure 2. Protocol between IOT Device and Middleware Server

Figure 2 is the IOTD and middleware server process status in the Client/Server format. Communication protocol uses the TCP Socket which is much used in most IOT communication methods. Communication protocol assigns some of the unknown ports. The basic protocol, applied by designing from the cold chain framework, is expressed as Figure 3 following the ASN.1 format. Applying the standard CoAP protocol, it was designed to enable reliability and unreliability message transmissions also in the protocol model of the cold chain framework [3-4].

a. Reliability Message Transmission b. Unreliability Message Transmission

Figure 3. Basic Protocol Format of Cold Chain Framework

IOTD & MW Server Communication Protocol Diagram

Field			Byte	Range	Description	Char
STX			1	0x02	Start of Packet	
LI			1	3 ~ 255	Length Indicator(STX를 포함한 총 Packet 길이)	
HEADER [M]	PGI Type		2	0x30~0x39	ASCII 2 Bytes	
	PID		20	0x30~0x39	PersonalID : Sensor ID	
	or		n	String	"	
		US	1	0x1F	Unit Seperator	
	IOT_ID		12	0x30~0x39	Phone Number : IOT Serial Number	
	or		n	String	"	
		US	1	0x1F	Unit Seperator	
	GW_ID		12	0x30~0x39	GateWay Serial Number	
	or		n	String	"	
		US	1	0x1F	Unit Seperator	
	FIRM		8	0x30~0x39	Sensor Version (Firmware version) : YYMMDDsss	
	or		n	String	"	
		US	1	0x1F	Unit Seperator	
	VER		8	0x30~0x39	Server Version (Alarm version) : YYMMDDsss	
	or		n	String	"	
		US	1	0x1F	Unit Seperator	
DELIMITER			1	0x7C	'|' (Shift + ₩)	|
BODY	DATETIME [M]	Descriptor	n	String	DATETIME	
		US	1	0x1F	Unit Seperator	
		YY	4	0x30~0x39	Measuring year	
		MM	2	0x30~0x39	Measuring month	
		DD	2	0x30~0x39	Measuring day	
		hh	2	0x30~0x39	Measuring hour	
		mm	2	0x30~0x39	Measuring minute	
		ss	2	0x30~0x39	Measuring sec	
	DELIMITER		1	0x1E	Record Seperator	
	TEMP [O]	Type	2	0x30~0x39	Type of sensor data / Index : 0x30x31	01
		Descriptor	n	String	Temperature	
		US	1	0x1F	Unit Seperator	
		Signed	1	0x2B, 0x2D		+/-
		Upper	2	0x30~0x39	측정온도 정수부	
		Point	1	0x2E		.
		Lower	1	0x30~0x39	측정온도 소수부	
	DELIMITER		1	0x1E	Record Seperator	
	Humidity [O]	Type	2	0x30~0x39	Type of sensor data (34/35/36/37)	14
		Descriptor	n	String	Humidity	
		US	1	0x1F	Unit Seperator	
		Value	3	0x30~0x39	정수부	
	DELIMITER		1	0x1E	Record Seperator	
	BAT [O]	Type	2	0x30~0x39	Type of sensor data	08
		Descriptor	n	String	Battery	
		US	1	0x1F	Unit Seperator	
		Value	3	0x30~0x39	정수부	
ETX			1	0x03	End of Packet	

범례

센서 및 GPS Data 정의	
?????	장치 오류, 측정 값 오류
XXXXX	처음부터 값이 없을 경우
ZZZZZ	최초 측정 후 값이 측정이 안될 경우
null	null value
0	value '0'

Parameter 설명	
length n	가변길이, variable length
US	Unit Seperator, 0x1F
RS	Record Seperator, 0x1E
M	필수항목, Mendatory
O	선택항목, Optionary

Figure 4. Frame Structure between IOTD and Middleware Server

The case of the loss of outgoing and incoming data influencing the service was defined as in Figure 4, to transmit and receive CON/ACK signals between the client and server at all times, and through this secured reliability. In cases of relatively low data reliability, it was designed to only transmit sensor data between the client and server as in b of Figure 3 and enabled high speed processing through this. But since it is fundamentally based on the reliability of TCP/IP, it is judged to have rarely any data loss.

D. Server (MWS)/Client(IOTD) Protocol Packet Structure of cold chain framework

The overall data frame structure between the IOTD and MWS is as shown in Figure 4.

Since in Figure 4, each primitive uses fixed and variable intermixedly on the protocol frame structure, the overall frame length cannot be fixed and the DateTime primitive of the header and body is always transmitted basically. The minimal length of the proposed cold chain protocol frame is 90 bytes, not exceeding the maximum length of 512 bytes, and it does not perform any separate data error tests.

PGI of Figure 4, as the Primary Group Index categorizing by protocol type, has the same results as Table 1, and has expandability for the addition of sensor connection network types.

Table 1. PGI Definition

PGI Name	Set number	Description	Note
BT Classic	“10“	BT Classic	‘0x31 0x30‘
BLe	“11“	BLe	‘0x31 0x31‘
WiFi	“12“	WiFi	‘0x31 0x32‘
ZIGBEE	“13“	ZIGBEE	‘0x31 0x33‘
UWB	“14“	UWB	‘0x31 0x34‘
IrDA	“15“	IrDA	‘0x31 0x35‘
NFC	“16“	NFC	‘0x31 0x36‘
RFiD	“17“	RFiD	‘0x31 0x37‘
CDMA(LTEx)	“18“	CDMA(LTEx)	‘0x31 0x38‘
LPWA	“19“	LPWA	‘0x31 0x39‘
WiBro	“20“	WiBro	‘0x32 0x30‘
WiMax	“21“	WiMax	‘0x32 0x31‘
Skinplex	“22“	Skinplex	‘0x32 0x32‘
			// Reserved for Future ~ “29”
RS232C	“30“	RS232C	‘0x33 0x30‘
RS422	“31“	RS422	‘0x33 0x31‘
RS485	“32“	RS485	‘0x33 0x32‘
USB	“33“	USB	‘0x33 0x33‘
FIREWIRE	“34“	FIREWIRE	‘0x33 0x34‘
PLC	“35“	PLC	‘0x33 0x35‘
			// Reserved for Future ~ “39”

3. Experimental Results

In order to test the effectiveness of the proposed protocol, this paper has collected temperature data in real time by implementing the sensor, gateway hardware, and software. Figure 5 is a block diagram of the test system.

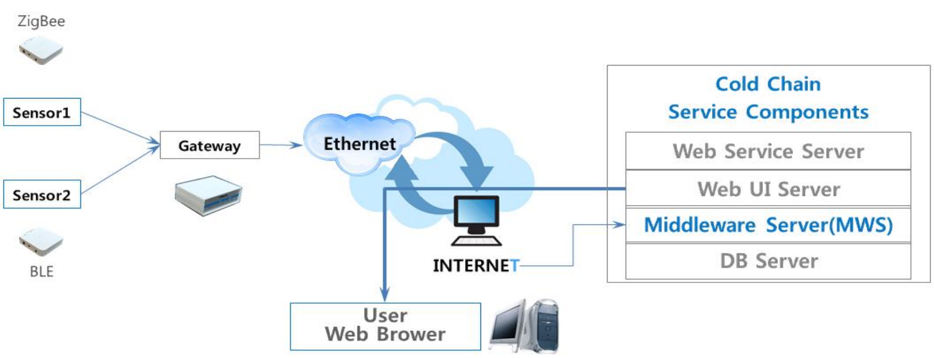

Figure 5. Block Diagram of Test System

To perform tests, Sensor1 used ZigBee and Sensor2 was produced to communicate in the Beacon mode of Ble. Gateway accessed the internet(Public IP Network) through a hub that uses a private IP, and cold chain components were constituted using public IP. MS Windows 2012 for server, MS-SQL 2014 for DB, and MS Visual Studio 2015 for developing tools were used. Figure 6 is the received data from the BLE Sensor IOTD and Table 2 is its interpretation.

??11Coldchain-
000000000174:F0:7D:C9:89:AA00.08.DC.01.02.032016041620160416|DATETIME2016051419
515101Temperature+24.314Humidity04208Battery082

02 98 31 31 43 6F 6C 64 63 68 61 69 6E 2D 30 30 30 30 30 30 30 30 30 31 1F 37 34 3A 46 30
3A 37 44 3A 43 39 3A 38 39 3A 41 41 1F 30 30 2E 30 38 2E 44 43 2E 30 31 2E 30 32 2E 30 33
1F 32 30 31 36 30 34 31 36 1F 32 30 31 36 30 34 31 36 1F 7C 44 41 54 45 54 49 4D 45 1F 32 30
31 36 30 35 31 34 31 39 35 34 32 32 1E 30 31 54 65 6D 70 65 72 61 74 75 72 65 1F 2B 32 34 2E
32 1E 31 34 48 75 6D 69 64 69 74 79 1F 30 33 36 1E 30 38 42 61 74 74 65 72 79 1F 30 38 33 03

Figure 6. Received BLE Sensor Data

Table 2. Interpretation of the Received BLF Data

Received Data	Length	Interpretation
02	STX(1)	Start of Text
98	LI(1)	
31 31	PGI(2)	Uses Blu Network
43 6F 6C 64 63 68 61 69 6E 2D 30 30 30 30 30 30 30 30 30 31	PID(20 or n)	Coldchain-0000000001
1F	US	
37 34 3A 46 30 3A 37 44 3A 43 39 3A 38 39 3A 41 41	IOT_ID(12 or n)	74:F0:7D:C9:89:AA
1F	US	
30 30 2E 30 38 2E 44 43 2E 30 31 2E 30 32 2E 30 33	GW_ID(12 or n)	00.08.DC.01.02.032
1F	US	
32 30 31 36 30 34 31 36	FIRM(8 or n)	20160416
1F	US	
32 30 31 36 30 34 31 36	VER(8 or n)	20160416
1F	US	
7C	DELIMITER(1)	
44 41 54 45 54 49 4D 45	DATETIME(8)	Descriptor
32 30 31 36 30 35 31 34 31 39 35 34 32 32	YYYYMMDDhhmmss (12)	20160514195422
1E	RS	
30 31	TYPE(2)	Temperature
54 65 6D 70 65 72 61 74 75 72 65	Descriptor	Temperature
1F	US	
2B 32 34 2E 32	Value	+24.2

1E	RS	
31 34	TYPE(2)	Humidity
48 75 6D 69 64 69 74 79	Descriptor	Humidity
1F	US	
30 33 36	Value	036
1E	RS	
30 38	TYPE(2)	Battery
42 61 74 74 65 72 79	Descriptor	Battery
1F	US	
30 38 33	Value	083
03	ETX	End of Text

Meanwhile, Figure 7 is the received data from the ZigBee Sensor IOTD and Table 3 is its interpretation.

??13Coldchain-000000000100155100006B8BDA00.08.DC.01.02.0320160416201604
16|DATETIME2016051419595401Temperature+24.914Humidity03508Battery082

02 98 31 33 43 6F 6C 64 63 68 61 69 6E 2D 30 30 30 30 30 30 30 30 30 31 1F 30 30 31 35 35 31 30
30 30 30 30 36 42 38 42 44 41 1F 30 30 2E 30 38 2E 44 43 2E 30 31 2E 30 32 2E 30 33 1F 32 30 31
36 30 34 31 36 1F 32 30 31 36 30 34 31 36 1F 7C 44 41 54 45 54 49 4D 45 1F 32 30 31 36 30 35 31
34 31 39 35 33 35 32 1E 30 31 54 65 6D 70 65 72 61 74 75 72 65 1F 2B 32 34 2E 34 1E 31 34 48 75
6D 69 64 69 74 79 1F 30 33 37 1E 30 38 42 61 74 74 65 72 79 1F 30 38 32 03

Figure 7. Received Zigbee Sensor

4. Conclusion

This paper designed and analyzed the communication protocol for the data communication between the IOT device and the middleware server of the IOT based Cold Chain System. Cold chain IOT devices include sensors, and in order to collect data from these sensors, there are two existing types. One is the type which directly connects between the sensor and the middleware server of the cold chain framework, and the other type is going through a gateway in between. In order to interface various IOT modules, the connection type through a gateway is generally used. For cold chain communication modules, communication modules such as Zigbee, Bluetooth, and active RFID are used, and communication modules to the server can be designed with Ethernet, Wifi, LTE, CDMA modules. communication between the IOTD and middleware server basically utilized the client/server model, and refered to the CoAP protocol model of IETF CoRE WG. In this paper, we have designed the connection setting and data transmission between the IOTD and middleware server, and the protocol and frame structure including disconnection, with detailed specifications on them.

And, we have produced a Zigbee, bluetooth and wifi module mounted gateway and the temperature, humidity sensor in order to test the effectiveness of the proposed Cold Chain Communication Protocol, After sending the data from the IOTD to the middleware server, we analyzed the received data from the middleware server. As the result, we were able to interpretate the data sent from the IOTD as in Figure 6,7 and Table 2,3, and were able to identify that in using this cold chain data such as the temperature, humidity, and battery remains data can be collected in real time.

Table 3. Interpretation of the Received Zigbee Data

Received Data	Length	Interpretation
02	STX(1)	Start of Text
98	LI(1)	
31 33	PGI(2)	ZigBee
43 6F 6C 64 63 68 61 69 6E 2D 30 30 30 30 30 30 30 30 30 31	PID(20 or n)	Coldchain-0000000001
1F	US	
30 30 31 35 35 31 30 30 30 30 30 36 42 38 42 44 41	IOT_ID(12 or n)	001551000006B8BDA
1F	US	
30 30 2E 30 38 2E 44 43 2E 30 31 2E 30 32 2E 30 33	GW_ID(12 or n)	00.08.DC.01.02.03
1F	US	
32 30 31 36 30 34 31 36	FIRM(8 or n)	20160416
1F	US	
32 30 31 36 30 34 31 36	VER(8 or n)	20160416
1F	US	
7C	DELIMITER(1)	
44 41 54 45 54 49 4D 45	DATETIME(8)	Descriptor
32 30 31 36 30 35 31 34 31 39 35 39 35 34	YYYYMMDDhhmm ss(12)	20160514195954
1E	RS	
30 31	TYPE(2)	Temperature
54 65 6D 70 65 72 61 74 75 72 65	Descriptor	Temperature
1F	US	
2B 32 34 2E 39	Value	+24.9
1E	RS	
31 34	TYPE(2)	Humidity
48 75 6D 69 64 69 74 79	Descriptor	Humidity
1F	US	
30 33 35	Value	035
1E	RS	
30 38	TYPE(2)	Battery
42 61 74 74 65 72 79	Descriptor	Battery
1F	US	
30 38 32	Value	082
03	ETX	End of Text

Acknowledgments

This research was supported by a grant (14TLRP-C 090585-01-000000) from Transportation & Logistics Research Program (TLRP) funded by Ministry of Land, Infrastructure and Transport of Korean government

References

[1] D. Ko, "A Study on Design of the Cold Chain Framework Based on Cloud", Journal of KIIT, vol. 13, no. 11, **(2015)**, pp. 79-87.
[2] L. Jiang, L. D. Xu, H. Cai, Z. Jiang, F. Bu and B. Xu, "An IoT- Oriented Data Storage Framework in Cloud Computing Platform", IEEE Transactions on Industrial Informatics, vol. 10, no. 2, **(2014)**, pp. 1443- 1451.
[3] Z. Shelby, Sensinode and K. Hartke, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18. [2013-06--28] http://tools.ietf.org/html/draft-ietf-core-coap-18
[4] IETF 79th meeting CoRE WG, http://www.ietf.org/proceedings/79/slides/core-0.pdf
[5] H. Park, "A Study on Analysis of Internet of Things Hardware for Real Time Monitoring of Cold Chain Logistics", Journal of KIIT, vol. 14, no. 3, **(2016)**, pp. 227-235.
[6] WHO, "Study protocol for temperature monitoring in the vaccine cold chain", the Quality, Safety and Standards (QSS), of the Department of Immunization, Vaccines and Biologicals.
[7] V. Karagiannis, "A Survey on Application Layer Protocols for the Internet of Things", Transaction on IoT and Cloud Computing, **(2015)**, pp. 1-10
[8] D. Ko and S. Song, "Design of Smart Cold Chain Application Framework Based on Hadoop and Spark", International Journal of Software Engineering and Its Applications, http://dx.doi.org/10.14257/ijseia, vol. 9, no. 12, **(2015)**, pp. 99-106.
[9] E. Ferro and F. Potori, "Bluetooth and Wi-Fi wireless protocols: a survey and a comparison", IEEE Wireless Communications, vol. 12, no. 1, **(2005)**, pp. 12-26.

Author

Dae-Sik Ko, 1978 Mar. ~1982 Feb.: Kyunghee university (B.S), 1998 Aug. ~ 1991 Feb.: Kyunghee university (M.S and PhD), 1994 Mar. ~ 1995 Feb.: UCSB Post-Doc, 2011 Jan. ~ 2012 Dec.: President of KIIT, 2002, 2013: Mokwon University Director Library and Computer Center, Dean of College of Engineering, 1989 ~ present: Professor of the Mokwon University Interested in Multimedia communication, IT convergence, Cloud Computing.