# SDN based QoS Control Methods with the Classifications of IoT Data Transmissions for Disaster Information Systems

Noriki Uchida[1]*, Kanji Nakano[1], Tomoyuki Ishida[1], and Yoshitaka Shibata[2]
[1]Fukuoka Institute of Technology, 3-30-1 Wajiro-higashi, Higashi-ku, Fukuoka, 811-0295 Japan
n-uchida@fit.ac.jp, mgm19105@bene.fit.ac.jp, t-ishida@fit.ac.jp
[2]Iwate Prefectural University, 152-52 Sugo, Takizawa, Iwate 020-0693, Japan
shibata@iwate-pu.ac.jp

**Abstract**

With the recent IoT (Internet of Things) technology developments, new functions are expected to the future DIS (Disaster Information System) with IoT devices. However, it is considered that the increased IoT data would cause congestions for Internet services, and it is necessary to consider the QoS (Quality of Service) controls for the system. Therefore, this paper proposes the QoS control methods using the SDN (Software Defined Network) by the classifications of IoT data transmissions. In detail, first of all, the routing functions with the Extend Dijkstra's algorithm is introduced. In the routing, IoT data is classified into four categories, and the proper route is selected by the Dijkstra's algorithm with the QoS priority values of the classifications. Also, the switching functions are introduced to avoid network congestion in the selected route. In the switching, if the IoT traffic load reaches more than 80 percent of the maximum throughput of the selected route, the IoT data is distributed by the additional route. The paper then reports the implementations of the prototype system by using the SDN, and the experiments show the effectiveness of the proposed methods.

**Keywords**: SDN, IoT, Disaster Information System

## 1 Introduction

It is considered that the DIS (Disaster Information System) technology have rapidly developed after the Great East Japan Earthquake in the March of 2011, and the new technologies such as IoT (Internet of Things) and 5G network are also expected to realize the new kind of applications for the DIS. Especially, IoT technology have spread with the various areas for the sensor networks and the home networking service in recent, and the new functions with these IoT devices are also expect to the future DIS technology [21][20].

However, the previous paper [16] reports IoT devices will be increased by over 20 billion around 2020, and it is afraid that the network in the ISP (Internet Service Provider) might become the higher packet delay or the packet loss. Then, it might affect the QoS controls for the whole Internet system [19] . After the large-scale disasters, it is necessary to consider the ISP network limited bandwidth because of the damages or the data explosions.

Therefore, this paper proposed the QoS control methods by using the SDN (Software Defined Network) with the classifications of IoT data transmissions for the core network of DIS. In details, the IoT data transmissions are firstly classified into the four categories by the traffic patterns [17] and the classifications are used for the extended QCI (QoS Class Identifier) [1] [15] in order to maintain the latency, the PER (Packet Loss Rate), or the throughput in the core network of DIS. Secondly, the proper routes

*Research Briefs on Information & Communication Technology Evolution (ReBICTE)*, Vol. 7, Article No. 2 (June 15, 2021)
DOI:*10.22667/ReBiCTE.2021.06.15.002*
*Corresponding author: Fukuoka Institute of Technology, 3-30-1 Wajiro-higashi, Higashi-ku, Fukuoka, 811-0295 Japan

in the core network are selected by the extended Dijkstra's algorithm [14] based on the QoS policies in the extended QCI. Next, the switching functions are confirmed to distribute the traffic loads if there are possible network congestions.

Then, the prototype system is implemented using OpenFlow [8] and the cloud services. The experiments are confirmed for the effectiveness of the proposed routing and switching functions by the system.

In the following, section 2 explained the assumed DIS network structures and the general functions. Then, section 3 discusses the proposed methods. In detail, this section deals with the classifications of IoT data transmission and the extended QCI, the proposed routing controls by the extended Dijkstra's algorithm, and the proposed switching controls for the traffic distributions. Then, the prototype system is explained using the SDN in section 4, and section 5 reports the experiments with the implemented prototype system. Finally, section 6 discussed the conclusion and the future studies in the proposed methods.

## 2 Proposed Network Architecture

This paper assumes the heterogeneous DIS network consisted of the core network, LAN (Local Area Networks), DTN (Delay Tolerant Networks) [12], and IoT wireless network as shown in Figure 1.

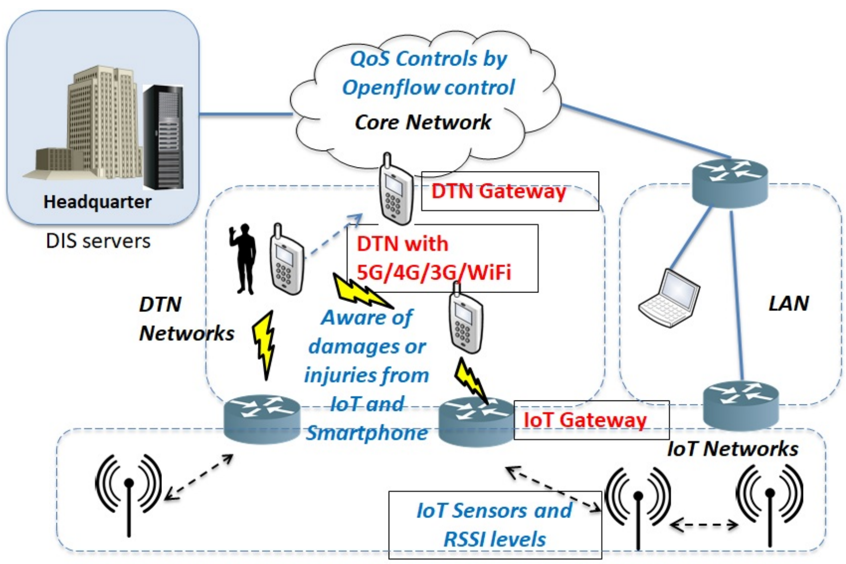


Figure 1: Proposed DIS Network

In the Figure, the core network is assumed as ISP networks, and this paper mainly discusses the proposed QoS controls by using OpenFlow in this network. Then, the core networks connect to LAN and DTN, and the DTN mainly provides the 5G/4G/3G/WiFi networking service after the disasters to aware of the damages or injuries from the sensor data of IoT devices and smartphones [21]. The DTN routing is the store-and-carry typed protocol, and the original approach was created by the interplanetary networks [9].

D2D (Device-To-Device) transmissions duplicate the data in the routing, so the DTN is also widely studied in disaster-related researches because of its robustness. Finally, the IoT network mainly consists of sensor devices and home network devices such as the temperature data, the room lighting, and streaming services by the Philips Hue [10] and the Chrome Cast [2]. Bluetooth, LoRa, and Zigbee connect the

IoT devices, and the data is transmitted to the cloud service in the core networks through the gateway such as Google Home [7] or Nature Remo [5] in the assumed networks.

However, since this paper especially focuses on the proposed QoS control methods from the IoT networks, Figure 2 is the assumed network architecture in this paper.

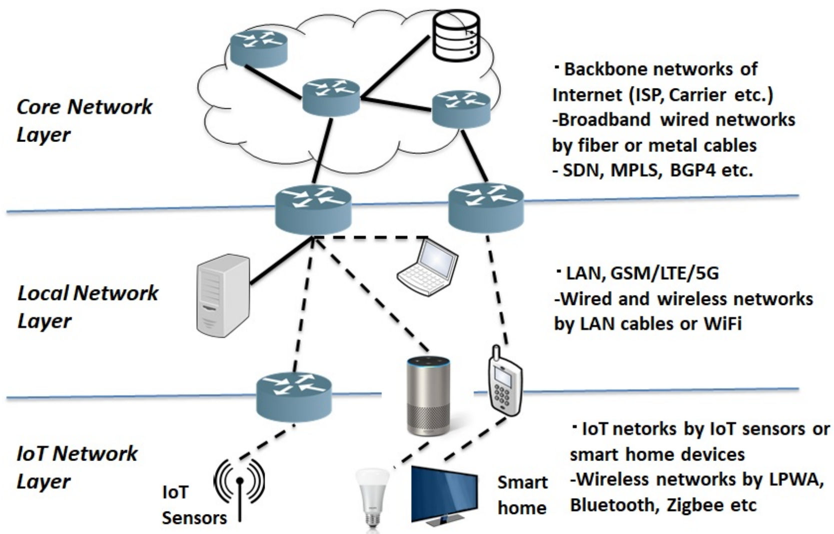


Figure 2: Assumed Network Architecture in This Paper

The assumed network consists of the core network layer, local network layer, and IoT network layer as shown in Figure 2. Then, the core network layer deals with the backbone of the Internet, such as ISP networks and the mobile carrier networks, and it is assumed that the SDN protocol is used inside the ISP or carrier network by broadband cables. Also, MPLS or BGP4 is assumed as the network protocol between the networks in the layer. Next, the local network layer is the LAN and smartphone accesses, and the network services are usually provided by the LAN, WiFi, and 3G/4G/5G with wired and wireless networks. Finally, the IoT network layer is the wireless networks such as the IoT sensors or the smart home devices. The devices are mainly connected to the IoT gateway by LPWA (Low Power Wide Area) such as LoRa, Bluetooth, and Zigbee in the layer.

Then, in the proposed methods, the IoT network layer's data transmissions are classified into four categories according to the traffic patterns, and they are the event-driven type, the streaming type, the sensor type, and the keep-alive type. The proposed routing and switching functions in the core network layer are then confirmed by using the extended QCI in the following sections.

# 3 Proposed Methods

## 3.1 Classifications of IoT Data Transmissions

In the proposed classifications of IoT data transmissions, the four categories are introduced as the event driven type, the streaming type, the sensor type, and the keep-alive type in the proposed methods. Here, the four categories of IoT data transmissions are the event driven type, the streaming type, the sensor type, and the keep-alive type. Then, the event driven type is the type of data when someone switches or configures the IoT devices. For example, when someone orders "OK, Google, switch the red light on!" to the Google Home, the smart light such as the Philips Hue turns on the light and configure the red light. Here, the voice command changes to the packet data from the smart speaker, and the data goes to the

GCP (Google Cloud Platform) Service. Therefore, this type can be distinguished by observing the GCP in the proposed methods [3][4][11]. Next, the streaming type is the video or audio streaming service using the Chromecast or the podcast. This type also goes through the cloud service such as the GCP when the services make actions. Therefore, the proposed methods can observe this type of IoT service.

Moreover, the sensor type is the data transmissions from IoT sensors such as the temperature or humidity. Currently, since this type of IoT sensors usually uploads to the cloud based database services such as the Firebase [3], this research implemented the observing functions in the cloud database service. Finally, the Keep-alive type is the data transmissions that regularly occur from IoT devices to keep the IP connections or condition monitoring. For example, Google Home or Chromecast regularly sends small packets to the servers to keep IP connections and the services. The proposed methods distinguish this type from the characteristic of the traffic pattern in the prototype system.

Then, this paper proposes the extended QCI by using these classifications for the user policy-based routing and switching controls. The original 3GPP QCI is the QoS class identifier for allocating the appropriate LTE communication service, and the order of priority, packet delay budget, and packet error loss rate are presented by type of QCI [15]. Therefore, this paper proposed the extended QCI including IoT classifications as shown in Table 1.

Table 1: Extended QCI Table

| QCI | IoT Type | Protocol | QoS Priority | Delay Budget | PER Budget |
|---|---|---|---|---|---|
| 1 | Event-driven Type | TCP | Delay | 60[ms] | $10^{-6}$ |
| 2 | Streaming Type | UCP | Throughput(Video)<br>/Jitter(Audio) | 100[ms] | $10^{-3}$ |
| 3 | Sensor Type | TCP | NA | NA | $10^{-6}$ |
| 4 | Keep-alive Type | TCP | PER | 300[ms] | $10^{-6}$ |

For example, the QCI 1 in the table is the Event-driven typed IoT data transmission, and TCP transmits the IoT data. Also, the QoS priority of this type should be the delay, and the affordable delay and PER are 60 ms and $10^{-6}$ according to the table.

Then, the proposed extend QCI is used for the extended Dijkstra's algorithm as the following section, and the optimal routing configurations are confirmed by the QoS policies in each IoT data transmissions.

### 3.2 Routing Functions with Extended Dijkstra's algorithm

Then, the optimal route in the core network is selected by the proposed extend Dijkstra's algorithm, and the OpenFlow confirms the network configurations in the proposed methods. The Dijkstra's algorithm is one of the shortest path tree algorism in the graph theory, and an optimal route is selected by the minimal cost of possible vertexes from a node at every step of the algorithm [14]. In the original algorithm, the path cost ($C_p$) of each vertex is previously configured in the network graph, and the sum of $C_p$ is calculated in every possible routes are compared. Then, the route of the minimum cost is selected as the shortest path.

In the proposed extend Dijkstra's algorithm, the $C_p$ of every vertex is calculated as the following

Formula (1) with the consideration of the QoS priority in the extend QCI.

$$C_p = \begin{cases} \dfrac{MaxBandwidth}{PathBandwidth}(WhenThroughputpriority) \\ 1 - \dfrac{Pathdelay}{Maxdelay}(WhenDelaypriority) \\ 1 - \dfrac{PathJitter}{MaxJitter}(WhenJitterpriority) \\ 1 - \dfrac{PathPER}{MaxPER}(WhenPERpriority) \end{cases} \tag{1}$$

Here, $C_p$ is the path cost of each vertex in the assumed network, and the minimal cost of possible vertexes is selected at each step in the proposed methods. For example, in the event-driven type of IoT data, the delay is focused on the QoS priority from the extended QCI. Then, $C_p$ is calculated at every vertex by using 60ms as the max delay and the observed value as the path delay, and the optimal route for the Event-driven typed IoT data is selected as the usual Dijkstra's algorithm methods. Also, in the calculations of $C_p$, if the observed delay of the vertex is more than 60ms in the extended QCI, the vertex is neglect for the route selecting processes in the proposed method.

Moreover, in the case of the video streaming typed IoT data, the throughput is set as QoS priority as the extend Dijkstra's algorithm. Here, the max throughput is previously configured by the network specifications of NIC (Network Interface Card) such as 100Gbps or 10Gbps, and the path throughput is used by the observed values from the network monitoring tools such as Zabbix. The max jitter and delay are then configured as 50ms from the previous field experiments in this paper.

### 3.3 Switching Functions for Network Congestion

This paper also proposes the switching functions for the distributions of the traffic load. If the IoT data is increased in the single selected route, the proposed switching function prepare the additional possible route. Then, the traffic is distributed to the multiple route by the switching controls in the proposed methods. Therefore, the network traffic is periodically observed by the cloud-based network monitoring services such as Zabbix [11]. The additional route from other QoS routes is used for the traffic distribution as the following steps.

1. When the observed values in a configured route satisfy the condition of formula (2), the flow entry from the OpenFlow controller is added to use an additional route of the other QoS policy for the exceeded packets.

2. When the observed values in the additional route satisfy the condition of formula (3), the added flow entry is deleted to return it to its original state.

3. Even if there are exceeded traffic using an additional route, more routes are automatically created by using the OpenFlow controller in ISP.

$$Th(max) \geq 80\% \tag{2}$$

$$Th(max) \leq 50\% \tag{3}$$

Here, Th(max) means the maximum throughput in the formula.

# 4 Prototype System

The prototype system is implemented for the evaluations of the proposed methods as shown in Figure 3.

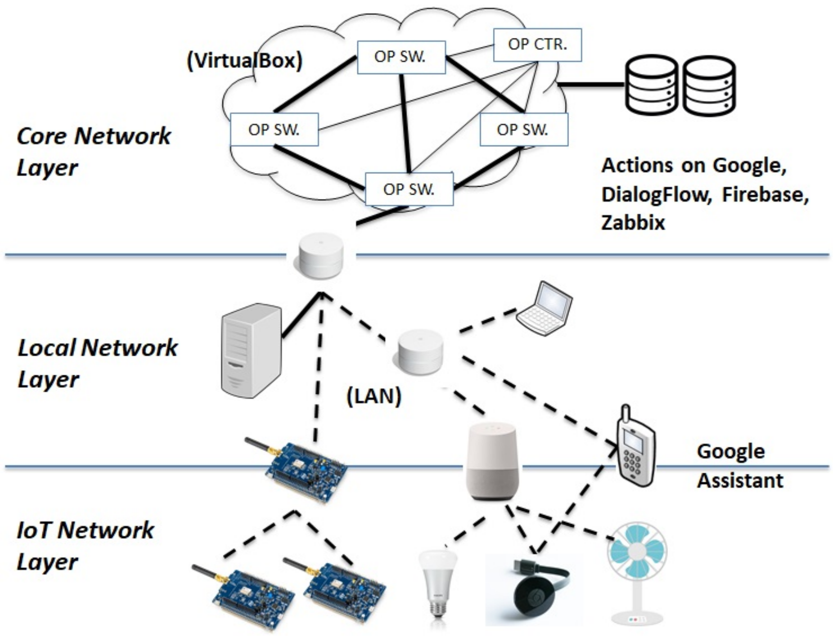


Figure 3: Prototype System of the Proposed Method

In the prototype system, the virtual ISP network of the Core Network layer is implemented by VirtualBox on the PC, and the specifications are as follows; CPU : Core i9-9900K, RAM : 32GB, GPU : NVIDIA GeForce GTX 1060 6GB, NIC1 : Intel I219-V, NIC2 : ASIX AX88179. Then, the virtual network is connected to the cloud services such as Actions on Google, Dialog Flow, Firebase, and Zabbix as shown in the figure. In the could services, the information of IoT data firstly send to the Actions on Google from the Google Assistant of the IoT gateway. Then, the proposed functions such as the classifications of IoT data, storing IoT data to Firebase, monitoring the network conditions by Zabbix and configuring the OpenFlow controller are mainly implemented in Actions on Google.

Next, the Local Network layer is implemented in the Uchida Lab. in Fukuoka Institute of Technology, Japan by Google WiFi [10][6] for the WMN and LAN.

Then, in the IoT Network layer, the prototype system uses the Philips Hue as the Event-driven typed IoT devices [9], the Google Chromecast as the streaming type [2], the Raspberry Pi 3B with Lora as the sensor type [18][13], and the Google Home as the keep-alive type [7].

Moreover, the gateway devices between the Local Network and the IoT Network Layers are the Raspberry PI 3B, the Google Home, and the smartphone (Google Pixel 3a) in the prototype system, and they are connected to the Actions on Google in the cloud services by installing the Google Assistant.

Therefore, the classifications of the proposed methods can be distinguished by these GCP services. In addition, the OpenFlow network is configured in the prototype system as shown in Figure 4 for the following experiments.

Three OpenFlow switches (OP SW.) are simply connected to each other in the network to evaluate the proposed routing and switching functions, and the OpenFlow controller is connected to these OpenFlow switches in the following experiments. Three OpenFlow switches (OP SW.) are simply connected to each other in the network to evaluate the proposed routing and switching functions, and the OpenFlow controller is connected to these OpenFlow switches in the following experiments.

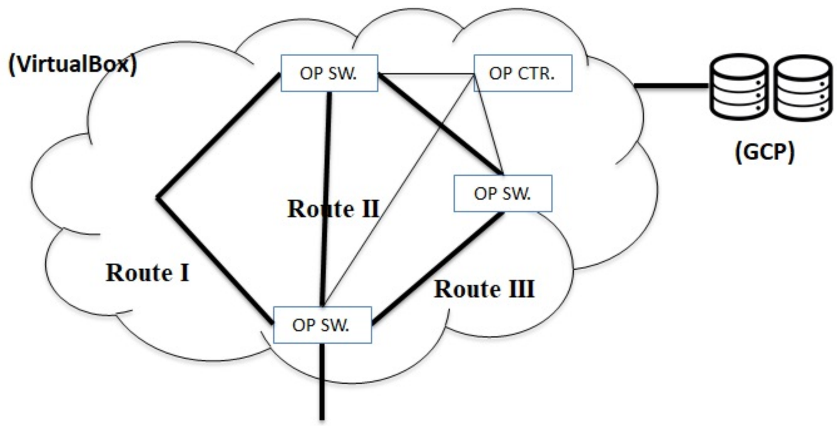


Figure 4: OpenFlow Network in the Prototype System

## 5 Experiments

In the experiments, the proposed routing functions and switching functions are evaluated by the implemented prototype system. Therefore, each route the virtual networks was previously configured the network conditions using the Meter functions in OpenFlow 1.3. The parameters are; route I is 1Gbps, route II is 100Mbps, and route III is 10 Mbps in the following experiments.

Then, first of all, the proposed routing functions have experimented with the following actions simultaneously.

1. Sensor typed IoT data (temperatures) is continuously sent to the database (Firebase) by Raspberry PI 3B. 2. Event-driven typed IoT data is sent by turning on the room light (Philips Hue) 3. Streaming typed IoT data is downloading.

That is, Google Home activates YouTube videos of the Chromecast.

Thus, Figure 5 is the captured monitoring windows of the prototype system.

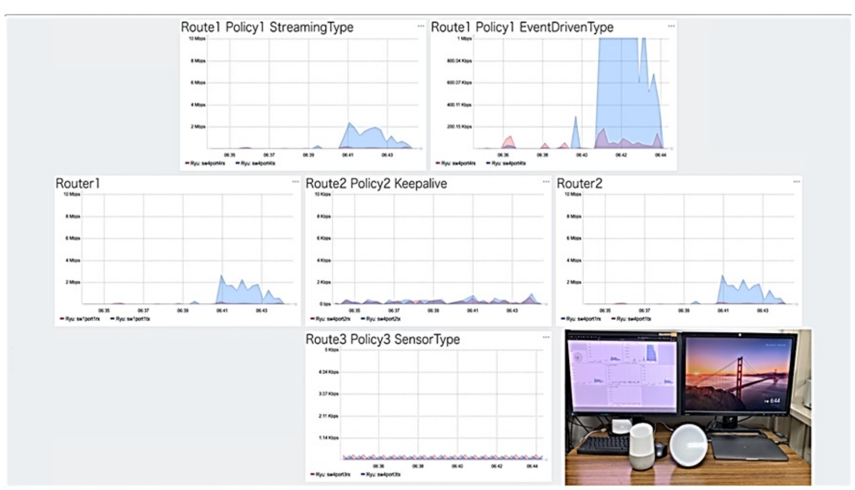


Figure 5: Captured Monitoring Windows of the Prototype System

As the results, route I is used for the Event-driven type and the Streaming type of IoT data by the proposed methods. The delay of route I is less than 1ms because of one hop, so the proposed methods selected for the delay priority data in the prototype system.

Additionally, the observed peak throughput of video streaming (Star Wars adverting video on Youtube) was 2.39 Mbps out of 1Gpbs as the maximum throughput in route I, it is supposed that the proposed methods properly selected the throughput priority route in the prototype system.

Secondly, Keep-Alive type IoT data selected route II in the experiments since it is the PER priority in the extended QCI. This results also show the effectiveness of the proposed methods because route II was the lowest PER in the prototype system. Finally, Sensor type IoT data also selected the rest of the possible routes correctly, and the data went through route III in the experiments.

Therefore, it is considered that the proposed methods properly classified the IoT data, and the OpenFlow in the prototype system controlled these data.

Next, the proposed switching selections were experimented by the prototype system. In the experiment, 1Gbps data streaming in route I was produced by iperf, and the network conditions of route I and II were observed by Zabbix. Figure 6 shows the transitions of each route in the experiment.


Figure 6: Results of Switching Functions

According to the results, it is considered that the data distribution to route II was started around 06:13. That is because the throughput of route I was reached 80% of the maximum throughput around 06:11. Then, the distribution ended at 06:15 because the throughput in route I decreased less than 50% of the maximum throughput.

Therefore, although there is a certain degree of delay by OpenFlow controls, it is considered that the proposed switching functions reduce the possibility of the network conditions when IoT data is increased.

## 6 Conclusion and Future Studies

It is considered that the rapid developments of IoT technology expect us to create new applications in various areas including the DIS. However, such rapid growth of IoT devices might cause network conditions, especially for the damaged networks after disasters. Therefore, this paper proposed the QoS control methods by using the SDN (Software Defined Network) with the classifications of IoT data transmissions for the core network of DIS.

IoT data transmissions are classified into four categories: the event-driven type, the streaming type, the sensor type, and the keep-alive type. Then, the proposed route selections are confirmed by the extended Dijkstra's algorithm with the extended QCI including the IoT classifications, to realize the QoS priority in DIS. Thirdly, the switching functions are also proposed for the network congestion in the selected route. Finally, these network configurations are dynamically controlled by OpenFlow from the cloud services such as Actions of Google.

Then, the paper presents the prototype system's implementations, and the experiments by the prototype system are reported. The results show the proper routing functions of the proposed methods by the prototype system, and it is supposed that the effectivity of the switching functions for the network congestion.

Now, the additional experiments with the prototype system are planning including the effectivity of the extend Dijkstra's algorithm by the more complex network conditions, and the modifications of the prototype system are also planning for the rapid SDN controls with the cloud services. Then, the field experiments are also the future studies for the actual usages of the DIS.

## Acknowledgments

This paper is supported by JSPS KAKENHI Grant Number 19K04972.

## References

[1] 3gpp standard.:standardized qos characteristics.

[2] Google inc. : Chromecast technical specification. `https://store.google.com/product/chromecast_2015_specs`, [Online; Accessed in Mar 2021].

[3] Google inc. : Dialogflow documents. `https://dialogflow.com/docs`, [Online; Accessed in Mar 2021.

[4] Google inc. : Firebase documents. `https://firebase.google.com/docs/`, [Online; Accessed in Mar 2021.

[5] Google inc. : Google home technical specification. `https://store.google.com/product/google_home_specs`, [Online; Accessed in Mar 2021.

[6] Google inc. : Google wi-fi specification. `https://support.google.com/wifi#topic=7216602`, [Online; Accessed in Mar 2021.

[7] Nature inc. : Nature remo technical specification. `https://nature.global`, [Online; Accessed in Mar 2021.

[8] Onf (open networking foundation): Openflow 1.3. `https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf`, [Online; Accessed in Mar 2021].

[9] Philips inc. : Philips hue technical specification. `https://developers.meethue.com/`, [Online; Accessed in Mar 2021.

[10] Qualcomm inc. : Qualcomm son specifications. `https://www.qualcomm.com/solutions/networking/features/wi-fi-son`, [Online; Accessed in Mar 2021.

[11] Zabbix : Zabbix specification. `https://www.zabbix.com/`, [Online; Accessed in Mar 2021.

[12] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41(6):128–136, June 2003.

[13] M. C. P. Consortium. *Mobile System Technical Text Expert Edition 8th Edition (In Japanese)*. RIC TELECOM, 2020.

[14] E. W. DIJKSTRA. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, June 1959.

[15] H. Jang and J. Lin. Sdn based qos aware bandwidth management framework of isp for smart homes. In *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6, August 2017.

[16] S. Kashima, T. Kondo, T. Murata, and M. Ueno. A study of security requirements and function implementationfor home networks in a period of iot expansion). Technical Report 481, March 2018.

[17] T. Ohnishi and N. Takahiro. Traffic prediction method based on correlation between iot devices. Technical Report 16, September 2018.

[18] H. Q. *Internet of Things A to Z: Technologies and Applications*. Wiley-IEEE Press, 2018.

[19] T. Tachibana, T. Furuichi, and H. Mineno. Implementation and evaluation of priority control mechanism for heterogeneous iot environment. In *Multimedia, Distributed, Cooperative, and Mobile Symposium (DICOMO2016)*, volume 2016, pages 1100–1107, July 2016.

[20] N. Uchida, M. Fukumoto, and Y. Ishida, T. Shibata. Implementations on static body detections by locational sensors on mobile phone for disaster information system. In *Advances in Internet, Data and Web Technologies*, pages 189–196. Springer International Publishing, January 2020.

[21] N. Uchida, K. Nakano, C. Takegawa, T. Ishida, and Y. Shibata. Proposal on the disaster information system based on qos controls of heterogeneous radio networks and radio atmosphere. In *2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)*, pages 1–5. IEEE, 2019.

## Author Biography

**Noriki Uchida** received the B.S. degrees from University of Tennessee in 1994, M.S. degrees in Software and Information science from Iwate Prefectural University in 2003, and Ph.D. degree degrees in the same University in 2011. From 2011 to 2014, he was an associate professor in the Saitama Institute of Technology, and he is currently an professor in the Fukuoka Institute of Technology.?His research interests include Cognitive Wireless Networks, QoS, and Heterogeneous Network. He is a member of IEEE, Information Processing Society of Japan (IPSJ), and Institute of Electronic and Communication Engineering in Japan (IEICE).

**Kanji Nakano** received his Information Engineering degrees from Fukuoka Institute of Technology, Japan, in 2019, and the M.S. degrees in Graduate School of Communication and Information Networking in the same university in 2021. Currently, he is working for NTT Laboratories in Japan, as a Researcher. His research interests include the SDN, Wireless Network Protocols, and Networking QoS Technology.

**Tomoyuki Ishida** received the B.S. and M.S. degrees in Software and Information science from Iwate Prefectural University in 2004 and 2006, and Ph.D. degrees in the same University in 2010. Currently he is an associate professor in the Ibaraki University. His research interests include Web Geographic Information System for local governments, Disaster Management System, Safety Confirmation System, Regional Disaster Prevention Planning, Virtual Reality and Tele-Immersion. He is a member of IEEE, Virtual Reality Society of Japan (VRSJ), Information Processing Society of Japan (IPSJ) and Visualization Society of Japan (VSJ).

**Yoshitaka Shibata** received his Ph.D. in Computer Science from the University of California, Los Angeles (UCLA), U.S.A. in 1985. From 1985 to 1989, he was a research member in Bell Communication Research, U.S.A., where he was working in the area of high-speed information network and protocol design for multimedia information services. Since 1998, he is working for Iwate Prefectural University, Japan as an executive director of Media Center and a professor of Faculty of Software and Information Science, and Information Science and Research and Regional Cooperation Division in the same university. He is a member of IEEE, ACM, Information Processing Society of Japan (IPSJ) and Institute of Electronic and Communication Engineering in Japan (IEICE).