Real-Time Awareness Scheduling for Multimedia Big Data Oriented In-Memory Computing

Real-Time Awareness Scheduling for Multimedia Big Data Oriented In-Memory Computing

Jianwen Xu, *Student Member, IEEE,* Kaoru Ota, *Member, IEEE,* and Mianxiong Dong, *Member, IEEE*

***Abstract*—As one of the most striking research hotspots in both academia and industry, Internet of Things (IoT) has been constantly changing our daily life by joining together nearly all we can imagine. From home furnishings and vehicles to urban facilities, all these smart things need powerful managing and processing capabilities to deal with mass multimedia data in different content forms such as images, audios, videos. Nowadays, since *Moore's Law* is no longer applicable, conventional thinking may not be adequate in facing the explosive growing amount of data. Hence, in this paper we adopt the idea of in-memory processing to solve the problem of real-time multimedia big data computing in IoT. We apply closed-loop feedback in the scheduling method design to integrate in-memory storages of all devices within a 3-tier network structure. In addition, we consider the respective conditions of different real-time required levels and content forms. The analysis results show that our scheduling method can achieve better workload allocation with less latency in comparison of existing methods.**

***Index Terms*—Internet of Things, Multimedia Big Data, Scheduling Method, In-Memory Processing.**

I. Introduction

SINCE the first Internet-connected device, a Coke machine at Carnegie Mellon University came into our view in 1982, the concept of a network made up of smart devices, Internet of Things (IoT) has gone through 35 years of development. Today we are already enjoying the comforts and conveniences brought by smart things from wearable devices, home furnishings to urban facilities. And in the future, even the Earth may experience the smartness that understanding the health conditions of our planet is just like the daily check on an Apple Watch 3.

To achieve interconnection from city scale to global and obtain the overall quantification and assessment such as traffic conditions, air quality, population distribution and ecological status, massive multimedia data collected by countless sensors and actuators in different content forms (text, images, audio and video) with strict real-time requirements put enormous pressure on transmission and processing. That is to say, the explosively growing multimedia big data needs IoT architectures to handle the surge of devices and be scalable enough to guarantee stable running.

Regarded as a recommended network standard of IoT, 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) defined by Internet Research Task Force (IRTF) provides many technical standards aiming to apply the Internet Protocol into all wireless devices [1]. Even the smallest and low-power ones can participate in network exchange without obstacles, which can well benefit IoT networking. Derived from cloud computing, fog computing uses collaborative multitudes of end-user clients and edge devices to implement large-scale distributed processing and management, which also support the development of IoT [2]. Together with other frontier technologies in fighting the challenge of brought by data amount, we are still looking for more possibilities outside of conventional thinking since there exists no guarantee that what we have always can meet what we want.

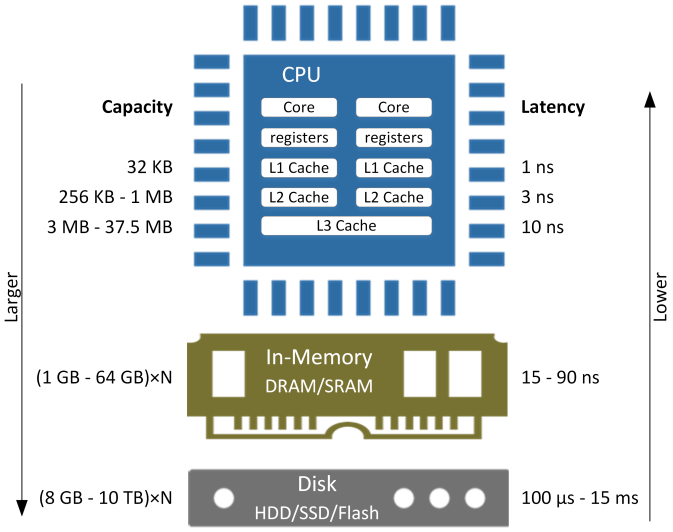


Fig. 1. Memory Hierarchy

In-memory, generally the RAM (Random Access Memory) inside a computer, stands for high-speed operation capability compared with NVM (Non-Volatile Memory). Used to be limited by volumes, sensitivities, fault-tolerance and consistency, in-memory system has to rely on storage and management capacities provided by hard disk. Generally computers input data from peripheral devices and store in hard disks, then read into RAM and wait for operations from processor. In-memory here plays the role of assisting upper caches and processors in quick addressing rather than mass storage taken by hard disks, which means high latency can be generated from the speed gaps between different levels in memory hierarchy as shown in Fig. 1.

Jianwen Xu, Kaoru Ota and Mianxiong Dong are with the Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan.

Manuscript received xx xx, 2017; revised xx xx, 2018.

In-memory processing is still a developing technology for processing data in in-memory storage or database. The concept first appears in the field of commercial database systems such as *Oracle*, *IBM DB2* and *Microsoft SQL Server* which aims to meet business intelligence (BI) needs including real-time analyzing, faster reporting and business decision making. The last decade has witnessed the rapid development of multi-core processors & larger and larger amounts of main memory with decreasing cost, making it available to build an entire system on in-memory. In a word, data processing based on in-memory rather than hard disk can be achieved in the near future.

According to the demand of multimedia big data computing in IoT, the real-time needs can just be satisfied by faster in-memory processing. Its low volume and consistency issues also are able to be alleviated through distributed processing from edge devices and pointed scheduling method.

Inspired by the advantages and applicable conditions of in-memory processing, our work focuses on how to maximize the utilization of finite in-memory processing resources and reduce latency to reach the real-time target as close as possible. We propose an scheduling method based on closed-loop feedback in workload allocation to integrate in-memory storages of all devices in the network structure. The main contributions are as follows.

- Formulate a mathematical model based on a 3-tier IoT network structure including multimedia wireless sensors, edge devices and central servers;
- Propose a closed-loop feedback scheduling method for integrating in-memory storages of all edge devices and servers and make rational workload allocation in different sensor numbers and simulation time;
- Choose total latency and workload allocation proportion as 2 metrics for evaluating the performances of our scheduling method and two existing methods, one that transmitting all raw data collected by wireless sensors to central servers and the other one depending on edge devices;
- Consider packets in different real-time levels and content forms to analyze the actual time cost from latency results.

This paper is divided by six sections to systematically elaborate our thoughts on in-memory scheduling. Section 2 introduces related works in interdisciplinary field of multimedia big data, IoT and in-memory processing. Section 3 formulates the mathematical model of a 3-tier IoT network structure and describes the problems to solve. Section 4 proposes the scheduling method making use of in-memory storages of both central and edge devices. Section 5 carries out simulation experiments and analyzes the algorithm performance of in-memory method and two existing methods. Section 6 summarizes all the work.

II. Related Work

Till now, the concept of Internet of Things already exceeds the scope of connecting machines and devices. A convergence of multiple technologies from embedded system, big data to artificial intelligence, real-time analytics have cooperated with IoT that leads to tremendous changes in industrial production and our daily life.

As a cornerstone of IoT, multimedia big data combines the advantages of relatively mature distributed processing technology and rich multimedia analysis theories and information resources, which attracts worldwide attention in scientific research fields. Financially sponsored by IEEE Computer Society, the 1st International Conference on Multimedia Big Data (BigMM) took place in Beijing, China. Since then multimedia big data has gradually evolved and become a new independent research direction, with more and more researches focusing on it [4].

Multimedia big data includes researches from basic theories and models for multimedia computing [5], energy efficient transmission [6], multimedia content analysis [7] to security and privacy considerations [8]. There are also many interdisciplinary studies such as health care and data visualization combining technologies from multimedia big data. Zhang *et al.* proposed a health cyber-physical system based on cloud computing and big data [9]. Ota *et al.* summed up deep leaning researches in mobile multimedia computing [10]. Ahmad *et al.* illustrated a scale free network visualization in smartphone based multimedia environment [11]. Sun *et al.* proposed a trust-based framework for fault tolerant wireless multimedia sensor networks [12] [13]. Li *et al.* adopted GPU-accelerated cloud computing in multimedia processing [14] [15]. Lee *et al.* looked into the prospect of service innovation and smart analytics in multimedia big data under the background of the 4th Generation Industrial Revolution (Industry 4.0) [16].

On the contrary of out-memory like Hard Disk Drive (HDD), flash memory and Solid-State Drive (SSD), in-memory mainly refers to volatile Random Access Memory (RAM) with almost the same amount of time for data reading/writing regardless of physical location. Although still limited by fault-tolerance and consistent power supply, the last decade has witnessed rapidly decreasing cost of main memory and growing demand of high-speed computing which makes it possible for in-memory processing to take place, turning RAM into the new disk [3] [17].

Inspired by many interdisciplinary studies, we adopt new idea in this paper of letting in-memory take on the workload of disk storage to differ from the aforementioned works. Researches of edge computing in the past mostly focused on how to efficiently utilize the computing resources among edge devices while we try to make use of the faster I/O speed but smaller volume in-memory processing and storage to further design a lightweight scheduling strategy to satisfy demands of IoT big data. That is, cut the whole computing task into more pieces and hand over to suitable devices which are idle or about to finish. Moreover, we also consider to minimize the time cost on waiting in order caused by small volume.

III. Problem Formulation

In this section, we formulate the mathematical model of a 3-tier IoT network system aiming at processing the mass data collected by multimedia wireless sensors, make necessary assumptions about conditions of the current scenario and explain the chosen performance metrics.

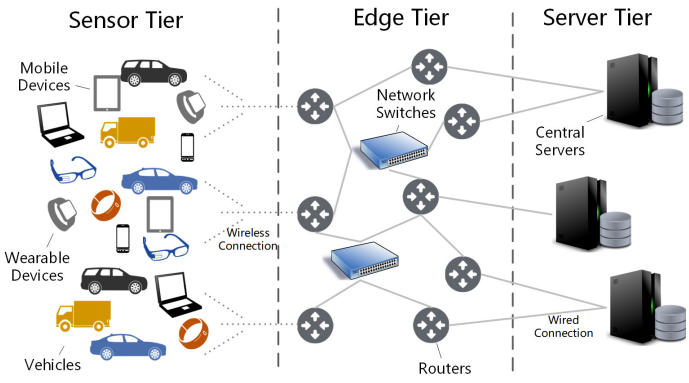


Fig. 2. A 3-tier IoT network structure

A. System Outline

As shown in Fig. 2, from outside to inside a 3-tier network model can be described as follows.

- *Sensor Tier:* a tier including all-propose wireless generalized multimedia wireless sensors (cameras, thermometers, motion sensors, air quality sensors, GPS compasses, etc) for collecting raw data for processing;
- *Edge Tier:* devices in this tier include routers, switches, integrated access devices (IADs) and etc. which all can be used in edge processing besides their original functions;
- *Server Tier:* in conventional method prior to the advent of edge computing, collected data are all delivered to this tier, and here we define it as the tier of data storage & management or data center.

In this model design, both edge devices and servers have the capacities to process images, audios and videos received from Sensor Tier. So as to deal with the multimedia big data with high demand on real-time, a high performance scheduling method on finite computing resource can help solving this problem to a great extent.

The two existing ideas respectively puts workload on servers and edge devices, the central method and edge method. In contrast with conventional central method, transferring the processing task to edge devices near the sensors can significantly cut down workload on central server and improve overall efficiency especially in large-scale distributed network infrastructure. Moreover, mass data that were originally centralized in servers are now being assigned to a great deal of edge nodes which may lessen communication bandwidths between in-tier devices (the devices of the same tier) and decrease the total data transmission [18]. However, edge computing still have to face the problem of actual deployment since available devices are very limited and there still exists room for further improvement. After learning the advantages of in-memory processing, we decide to apply it in reducing the total latency to approach the target of real-time multimedia big data analysis in Fig. 2.

B. Model Setup

For the purpose of achieving a low latency scheduling method based on in-memory processing in the designed 3-tier network system, some condition assumptions and model setups are needed to standardize and simplify the actual supply and demand of the current scenario.

Wireless sensors are already a familiar part of our daily life. For instance, inside almost omnipotent mobile phones that still are being updated rapidly there are many embedded sensors like accelerometers, digital compasses, gyroscopes, etc [19]. It is the same with sensors, wearable devices and vehicles, which are moving in different directions and speeds. To consider more actual details and fit all situations, we apply the Random Waypoint (RWP) model into setup of node distribution of mobile sensors [20].

In a typical random waypoint model design, we have to set intervals for moving speed V ($[v_{mov,min}, v_{mov,max}]$), moving time T ($[t_{mov,min}, t_{mov,max}]$), pause time ($[t_{pau,min}, t_{pau,max}]$) and moving direction Θ ($[\theta_{min}, \theta_{max}]$) and then obtain the consecutive movements. Distance and position modification of a simple example yield

$$\begin{aligned} d_{StD} &= |(x_S, y_S), (x_D, y_D)| = v_{mov} t_{mov} \\ x_{StD} &= |x_S - x_D| = d_{StD} / \sin\theta \\ y_{StD} &= |y_S - y_D| = d_{StD} / \cos\theta \\ v_{mov} &\in V, t_{mov} \in T, \theta \in \Theta \end{aligned} \tag{1}$$

We divide all sensors into 4 kinds according to different content forms: image, audio, video and interactive content. Each kind of sensors may own its packet size limit and priority level of real-time. Interactive content packets are always level 1 which will be forwarded first, packets of the other three kinds may be marked as level 2 or 3 and take turns. Positions of devices in Edge Tier and Server Tier are fixed.

C. Performance Metrics

To test the performance of our proposed scheduling method using in-memory processing and meet the demand of real-time as much as possible, we choose total latency and resource allocation ratio as 2 metrics.

1) Total Latency: There are many latencies/delays in different kinds of network systems. In our model the total latency ($L_{tot}(t)$) means the response time between data collection and processing which include 2 parts, end-to-end latency ($L_{end}(t)$), processing latency ($L_{proc}(t)$). t stands for the time-slot of the current number of packets in calculation [21].

$$L_{tot}(t) = L_{end}(t) + L_{proc}(t) \tag{2}$$

The end-to-end latency stands for the time taken for packets to travel across a network from source to destination. Here we use $L_{end}(t)$ to denote the across-tier (data transmission between different tiers) time cost which is made up of 2 parts, transmission delay (D_{tran}) and propagation delay (D_{prop}).

$$\begin{aligned} L_{end}(t) &= \sum_{i}^{n} (D_t^{tran}(i) + D_t^{prop}(i)) \\ &= \sum_{i}^{n} (Z_{pkt}(i)/R_{bit} + l_{end}(i)/v_{prop}) \end{aligned} \tag{3}$$

As shown in Equation (3), there are n packets generated in t with size of Z_{pkt} in $bytes$ and physical distances l_{end}.

R_{bit} and v_{prop} mean the bit rate and wave propagation speed, respectively. As a result, D_{tran} only cares about the packet length and has nothing to do with how far it will be transmitted. On the contrary, D_{prop} depends on travel distance and communication medium such as air (wireless, speed of light, c), copper wire (generally ranges from $0.59c$ to $0.77c$). From Sensor Tier to Edge Tier and Edge Tier to Server Tier there are $L_{s,e}^{end}$ and $L_{e,c}^{end}$, which means end-to-end latency of packets processed in Server Tier includes both segments.

For $L_{proc}(t)$, we also consider 2 aspects, processing delay measured from maximum transfer rate (MTR) for CPU addressing and queue time caused by finite memory (in our model, read from in-memory or disk). In consideration of the real-time level, packets form in prior level can take the lead in transmitting and processing within the time-slot t.

$$L_{proc}(t) = \frac{Z_{proc}^{pkt}(t) \sum_{j=t-\tau}^{t} \sum_{i=1}^{n} Z_{wat}^{pkt}(j)}{v_{MTR}} \tag{4}$$

where $Z_{proc}^{pkt}(t)$ and $Z_{wat}^{pkt}(t)$ respectively represent the total size of n packets under processing within time-slot τ and waiting in order.

2) Workload Allocation Proportion: Besides transmission latency and processing latency, we also pursue rational workload allocation on each device in the same tier and overall proportion of Edge Tier and Server Tier.

$$\begin{aligned} P_{edge}(t) &= \frac{Z_{edge}^{pkt}(t)}{Z_{edge}^{pkt}(t) + Z_{ser}^{pkt}(t)} \times 100\% \\ P_{ser}(t) &= \frac{Z_{ser}^{pkt}(t)}{Z_{edge}^{pkt}(t) + Z_{ser}^{pkt}(t)} \times 100\% \end{aligned} \tag{5}$$

$P_{edge}(t)$ and $P_{per}(t)$ stand for workload proportions of in-memory scheduling method by percentages after summing up all packets in bytes. We are going to compare the simulation results according to amount of data in unit time and time-slot.

IV. Scheduling Method

In this section, we propose a closed-loop feedback based scheduling method for applying in-memory processing to solve the real-time aware multimedia big data problem.

A. Closed-Loop Feedback Scheduling

As shown in Figure. 3, we regard one central server and edge devices connected to it as two systems. When the central server in *System 1* faces multimedia data input exceeding the current processing capacity, it may send instructions back to all lower edge devices to raise the proportion of workload distribution later. The relatively decentralized *System 2* may also have to deal with the balance of utilization rate on computing resource of each edge device besides the overall capacity. *Input* and *Output* respectively stands for the workload allocation before and after a closed-loop feedback, which means our target is to schedule the succeeding transmission

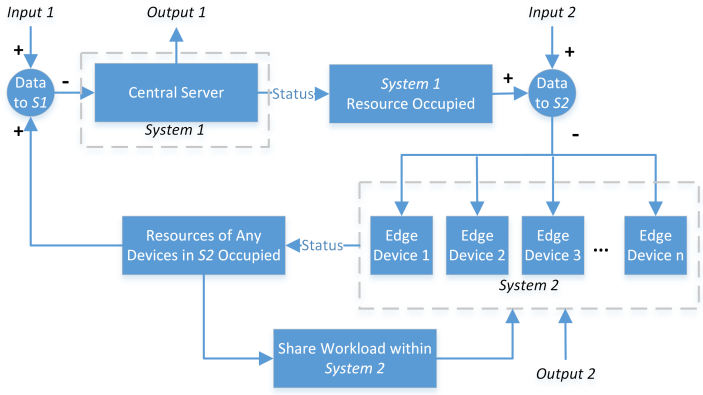


Fig. 3. A Closed-Loop Feedback Based 2-System Model

rather than causing extra end-to-end latency by returning multimedia data.

$$Z_{total}^{pkt}(t) = \underbrace{Z_1^{pkt}(t)}_{Input\ 1} + \underbrace{\sum_{i=1}^{n} Z_{2,i}^{pkt}(t)}_{Input\ 2} \tag{6}$$

As shown in Equation (6), *Input 1* represents the all the packets sent upstream to Server Tier while *Input 2* consists of packets allocated to each edge devices. $Z_1^{pkt}(t)$ and $Z_{2,i}^{pkt}(t)$ are total sizes of packets summed up. n stands for the number of overloaded edge devices.

$$\begin{aligned} Output\ 1 &= (1 - \frac{Z_1^{pkt}(t) - pC_1^{proc}}{C_1^{proc}}) \cdot Z_1^{pkt\prime}(t) + Z_1^{sup} \\ &= \frac{(p+1)C_1^{proc} - Z_1^{pkt}(t)}{C_1^{proc}} \cdot Z_1^{pkt\prime}(t) + Z_1^{sup} \end{aligned} \tag{7}$$

$$\begin{aligned} Output\ 2 &= \sum_{i=1}^{n} \{[1 - \frac{Z_{2,i}^{pkt}(t) - q_i C_{2,i}^{proc}}{C_{2,i}^{proc}}] \cdot Z_{2,i}^{pkt\prime}(t)\} + Z_2^{sup} \\ &= \sum_{i=1}^{n} [\frac{(q_i+1)C_2^{proc} - Z_{2,i}^{pkt}(t)}{C_2^{proc}} \cdot Z_{2,i}^{pkt\prime}(t)] + Z_2^{sup} \end{aligned} \tag{8}$$

In Equation (7) and (8), C_1^{proc} and $C_{2,i}^{proc}$ respectively denotes processing capacity of single device in two systems. p and q_i are integers to make sure that both *Input 1* and *Input 2* are positive which keeps the feedback loop going.

$$\begin{cases} Z_1^{pkt}(t) - pC_1^{proc} \geq 0 \\ Z_1^{pkt}(t) - (p+1)C_1^{proc} < 0 \end{cases} \tag{9}$$

$$\begin{cases} Z_{2,i}^{pkt}(t) - q_i C_{2,i}^{proc} \geq 0 \\ Z_{2,i}^{pkt}(t) - (q_i+1)C_{2,i}^{proc} < 0 \end{cases} \tag{10}$$

The prime symbol means the *Input* of next t time-slot. Z_1^{sup} and Z_2^{sup} are supplements of packet size generated by current scheduling in time-slot t which yield

$$Z_1^{sup} + Z_2^{sup} = \frac{Z_1^{pkt}(t) - C_1^{proc}}{C_1^{proc}} Z_1^{pkt\prime}(t) + \sum_{i=1}^{n} [\frac{Z_{2,i}^{pkt}(t) - C_{2,i}^{proc}}{C_{2,i}^{proc}} Z_{2,i}^{pkt\prime}(t)] \tag{11}$$

We prefer the cut packets from *Input* of next time-slot t after scheduling be reallocated to two systems according to processing capacities of devices.

$$Z_1^{sup} = \frac{C_1^{proc}}{C_1^{proc} + \sum_{i=1}^{N} C_{2,i}^{proc}} \{\frac{Z_1^{pkt}(t) - C_1^{proc}}{C_1^{proc}} Z_1^{pkt\prime}(t) + \sum_{i=1}^{n} [\frac{Z_{2,i}^{pkt}(t) - C_{2,i}^{proc}}{C_{2,i}^{proc}} Z_{2,i}^{pkt\prime}(t)]\} \tag{12}$$

$$Z_2^{sup} = \frac{\sum_{i=1}^{N} C_{2,i}^{proc}}{C_1^{proc} + \sum_{i=1}^{N} C_{2,i}^{proc}} \{\frac{Z_1^{pkt}(t) - C_1^{proc}}{C_1^{proc}} Z_1^{pkt\prime}(t) + \sum_{i=1}^{n} [\frac{Z_{2,i}^{pkt}(t) - C_{2,i}^{proc}}{C_{2,i}^{proc}} Z_{2,i}^{pkt\prime}(t)]\} \tag{13}$$

where N represents the number of all devices in *System 2*.

Since number of packets waiting for processing in front of *System 1* and *System 2* is not the same, workload allocation after last adjustment on data stream may be affected in various degrees. That is, the scheduling work can not be done at once, we need to continually approach the steady state by closed-loop feedback.

B. Real-Time Level Priority Processing Algorithm

In case of finite computing resource and our different demands on multimedia content forms and contents themselves, the priority of data transmission also may bring variations to latency calculation.

As shown in Algorithm 1, when multimedia mass data flood in and overpass the processing capacity of edge devices & servers at once. Packet set S^{pkt} represents the data size that can be handled currently according to the hardware indexes of devices. Before grouping, *line 4-12* sort all packets in the order of real-time levels (RT_{lv}) and size in bytes (Z_{pkt}), that is, packets with higher real-time level and smaller size can obtain priority among all the others in time-slot t.

Then we consider the quantified analysis of the proposed framework and algorithm in terms of complexity. In this paper we design a closed-loop feedback scheduling method and a real-time level priority processing algorithm to help solve the problem of multimedia big data in IoT. First we assume that there are N edge devices in *Edge Tier*, in which n ones are overloaded in current time-slot ($n \leq N$). In the worst case, which means we have to reschedule computing resources in all edge devices ($n = N$), as a result it takes $O(Output\ 1)$+$O(Output\ 2)$=$O(1 + |N + 1|^2) + O(|N| + |N + 1|^2)$=$O(|N|^2)$. Second to respectively sort the packets in different types, suppose that there are M packets being sent and m sets of packets to be processed in order in the current time-slot, it takes $O(|M \times (M-1)/2| + |m|)$ to finish the whole procedure. Thus the overall time complexity is $O(|N|^2 + |M|^2 + |m|)$.

Algorithm 1 RPPA: RTL Priority Processing Algorithm

Input: $P_i^{kt}, i \in [1, M]$ ←all M packets sent in time-slot t
Output: $S_i^{pkt}, i \in [1, m]$ ←all m sets of packets to be processed in order

```
1: Z_pkt ←packet size in bytes
2: RT_lv ←real-time level of data packet
3: C_proc ←processing capacity of device
4: for i = 2 to M do
5:    for j = M to i do
6:       if P_j^kt.RT_lv < P_{j-1}^kt.RT_lv then
7:          swap P_j^kt and P_{j-1}^kt
8:       else if P_j^kt.RT_lv = P_{j-1}^kt.RT_lv && P_j^kt.Z_pkt <
         P_{j-1}^kt.Z_pkt then
9:          swap P_j^kt and P_{j-1}^kt
10:      end if
11:   end for
12: end for
13: for i = 1 to m do
14:   while ∑ P_x^kt ≤ C_proc do
15:      put the front x P^kt into S_i^pkt
16:   end while
17: end for
```

V. Simulation and Analysis

In this section, we carry out experimental simulations to validate the performance of proposed scheduling method and compare with the other two existing methods on network latencies and workload allocation proportion.

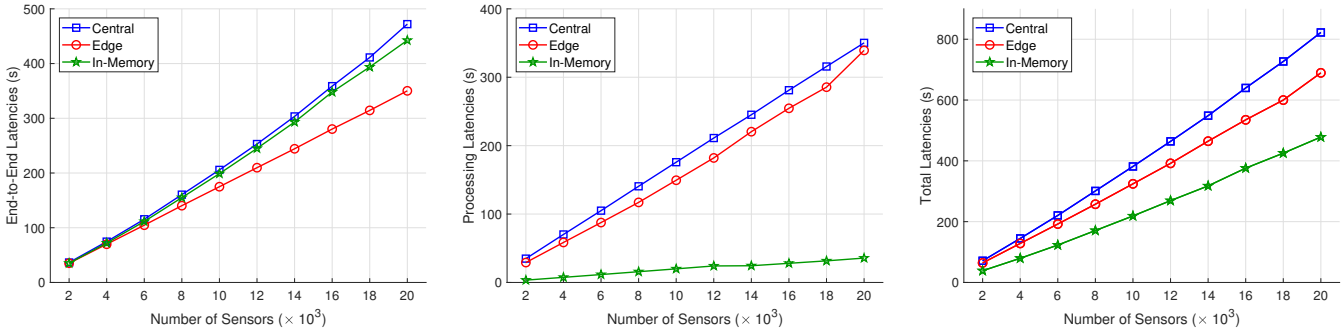
The experimental scenario is a 10 km^2 square open area in urban city, and we set up 2,000 to 20,000 wireless multimedia sensors which are moving continuously under the RWP model after each packet delivery. There are 150 routers as edge devices and three central servers. We carry out simulation experiments by taking time-slots for packets sending and processing. In a single time-slot t, the number of packets generated by each sensor obeys a Poisson distribution.

Table I gives details of experiment setups, including bit rates & wave propagation speeds of transmission part and device settings. We carry out 10 time-slots for experiment and the simulation environment is *MATLAB R2016a*.

A. Total Latencies of Three Methods

Fig. 4 gives the simulation results of end-to-end latencies, processing latencies and their totals. On the whole, with the increase in sensor numbers which means more packets are generated in the same time, all three latencies show linear growth patterns.

In Fig. 4a, when there are only 2,000 sensors in Sensor Tier, end-to-end latencies in three methods stay the same.



(a) End-to-End Latency (b) Processing Latency (c) Total Latency

Fig. 4. Latencies in Three Methods of Varied Sensor Numbers

TABLE I
EXPERIMENT SETUPS

Bit Rate of Transmission	
Wireless (802.11ad)	6.8 Gbit/s
Ethernet	10 Gbit/s
Wave Propagation Speed of Transmission	
Wireless (air)	**c** (speed of light)
Ethernet (thick coax)	0.77 **c**
Device Settings	
MTR of Disk (SATA3)	750 MB/s
MTR of In-Memory (DDR3)	6.4 GB/s
Disk Volume of Edge	256 MB
Disk Volume of Server	500 GB
In-Memory Volume of Edge	16 MB
In-Memory Volume of Server	4 GB

Then as more sensors join in, the gap between *Center* (blue broken line) and *Edge* (Red broken line) appears. Based on Fig. 2, multimedia packets being processed in Server Tier may have to travel longer and be forwarded more hops than those in Edge Tier, resulting in higher latencies on propagation and transmission. When the sensor number reaches 20,000, packets generated take about 20% time on transmission in Central Method than Edge Method. By virtue of both Edge Tier and Server Tier, multimedia packets in our proposed *In-Memory* Method can choose to give their workload to either tier according to scheduling strategies on reducing the total latencies, workload allocation on 2 tiers and requirements from real-time levels & content forms. End-to-end latencies of *In-Memory* Method (green broken line) stay between *Central* and *Edge* and closer to the former one which means our method seems to rely more on Server Tier. Some more experiment results are needed to help supporting this inference.

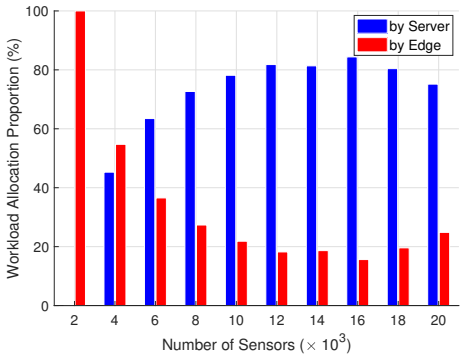
In Fig. 4b, the other main part of total latency, response time intervals in packet processing procedure show different patterns to Fig. 4a. Firstly we also consider the comparison between *Central* and *Edge*, latencies of packet processing on Server Tier cost more time than Edge Tier, which means in the current 3-tier network model, edge computing does help reducing time while facing mass multimedia data compared to conventional way. Besides, at the minimum and maximum number of sensors, *Edge* comes closer to *Central*, i.e. broken line of *Central* owns higher linearity degree than *Edge*. Consequently, *Edge* method that multiple distributed devices sharing workload behaves more sensitive to amount of data (sensor number) in a unit time. Although taking the advantage of locality and low risk on network congestion, relatively lower processing latencies can be achieved compared to *Central*, *Edge* method still faces unbalanced workload allocation among edge devices in-tier especially when amount of data is too much or too little. *In-Memory* Method aims to combine the advantages of the other two existing methods to approach the real-time target as well as rational workload allocation to maintain stable performance under different levels of workload. From Fig. 4b, *In-Memory* gets very low in numerical value and growth rate which means under ideal conditions in the current model computing based on in-memory rather than disk can greatly reduce time cost on processing part.

In summary, the total latencies of three methods show smooth linear relationships between sensor numbers and latency values. Taking account of the analysis above, our proposed scheduling method using in-memory processing can further reduce time cost in the 3-tier network model shown in Fig. 2. The great advantages on maximum transmission rate of in-memory reserve large space for continuous growth of multimedia big data.

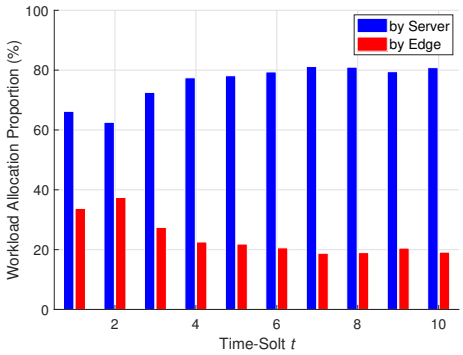
B. Workload Allocation Proportion of In-Memory Method

As the other target besides total latency, we also focus on how workload are assigned to each device in Edge Tier and Sensor Tier. 2 subfigures in Fig. 5 display the workload allocation results of *In-Memory* method by percentages.

First in Fig. 5a, the same as Fig. 4, the proportion of workload on Server Tier and Edge Tier of different sensor numbers shows the task allocation between servers and edge devices facing increase of data amount. Blue and red bars stand for percentage results of 2 tiers, respectively. When there are not many multimedia sensors, we assign most workload to Edge Tier since edge can well deal with small quantity of data (even 100 % by Edge for 2,000 sensors). Then as the



(a) Sensor Number Result



(b) Time-Solt Result

Fig. 5. Workload Allocation Proportion by Tier 2&3 in In-Memory Method

number rises, proportion of *Edge* rapidly decrease lower to *Server*. Combined with analysis of Fig. 4a, our scheduling method depends more on Server Tier according to different current computing capacities. Moreover, when amount of data reaches another magnitude, about 16,000 sensors as packet senders, the situation has been reversed. Proportion of *Edge* increases like a rebound after reaching the bottom which even shows that our method owns some scalability.

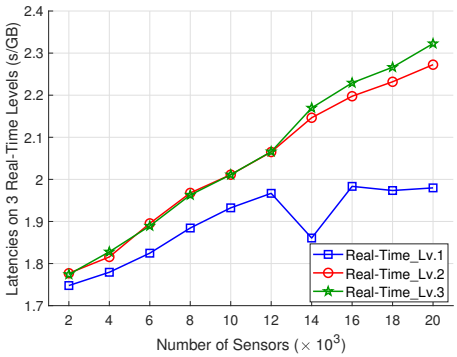
Second in Fig. 5a, we change the x axis to time-slots to analyze and verify the closed-loop feedback design shown in Fig. 3. Percentage values of 2 tiers do not experience much violent change like in Fig. 5a. After a small peak from 2,000 to 4,000 sensors, the ratio of *Server* and *Edge* is gradually stabilized to 4:1. Although still fluctuating within a narrow range, under the function of closed-loop feedback, our proposed scheduling method does play the role of rationally allocating workload to infinite computing resources as well as keeping certain stability in the 3-tier network model.

C. Latencies in consideration of Real-Time Levels and Content Forms

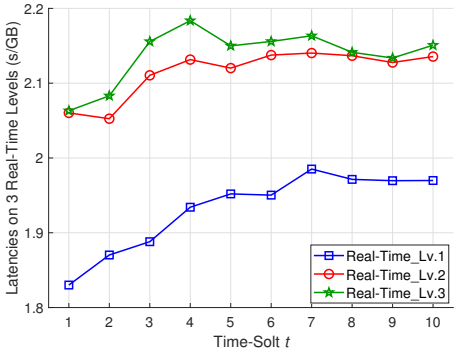
In order to be competent in real-time multimedia big data computing in IoT network infrastructure, we also have to consider the conditions of different content forms and time requirements. For example, in social network service (SNS) there exist many interactive contents which require service users' active engagement and get real-time, hyper-relevant returns they immediately care about. As a result, we can

TABLE II
CONTENT FORMS OF MULTIMEDIA PACKETS

Content Form	Packet Size Scope (byte)	Real-Time Lv.
image	34 ~ 1024	2 or 3
audio	512 ~ 1536	2 or 3
video	1024 ~ 2304	2 or 3
interactive content	34 ~ 2304	1



(a) Sensor Number Result



(b) Time-Solt Result

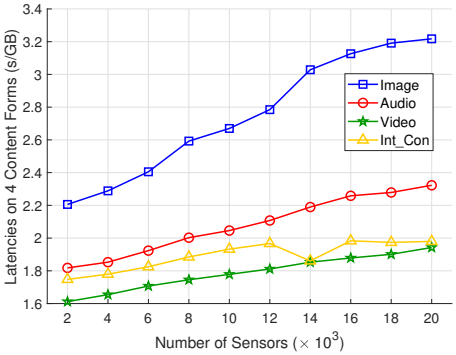
Fig. 6. Latencies of Three Real-Time Levels in In-Memory Method

give interactive contents a higher real-time level for priority transmission/processing to ensure the quality of service (QoS).

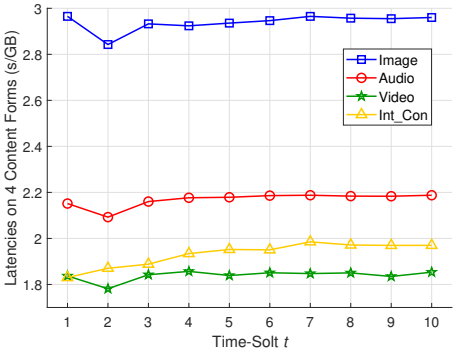
As shown in Table II, we set 4 content forms in minimum size of 34 composed by frame check sequence (FCS) & header and maximum size of 2304 as maximum transmission unit (MTU) in IEEE 802.11 standards. Then we use a real-time level to make distinctions among varied requirements on different contents and evaluate the latency performance in the current network model with different transmission priorities and multimedia content forms.

In Fig. 6, we add up latencies of all packets in the same real-time level respectively with the increase of sensor number and time-slot. Now that the packet numbers being generated in each real-time level (so does each content form) are on the basis of Poisson distribution whose latency results can not be compared directly, we adopt the total packet sizes to get the time cost per gigabyte (GB).

First in Fig. 6a, with more sensors involved in simulation, latencies of three real-time levels show varied patterns. $Lv.2$ (red) and $Lv.3$ (green) are marked upward trends and until



(a) Sensor Number Result



(b) Time-Solt Result

Fig. 7. Latencies on Total Packet Size of Four Content Forms in In-Memory Method

sensor number reaching 14,000 that the priority is reflected in figure. $Lv.1$ (blue) is slightly lower in numerical value and maintain similar increase pattern in the left half. When the amount of data is far from reaching the capacity of our designed 3-tier network model, most packets can be transmitted and processed instantaneously (in such condition, packets in $Lv.1$ seize the few opportunities of faster *Green Channel*), resulting in a relatively narrow gap of latencies in different real-time levels. Then a critical point in sensor number is reached (about 12,000), besides the differentiation of $Lv.2\&3$ even an abnormal valley appears which may be explained by a rebound pattern of Fig. 5a that during the interval of workload proportion adjustment, packets in $Lv.1$ are just filling up the regular network throughput.

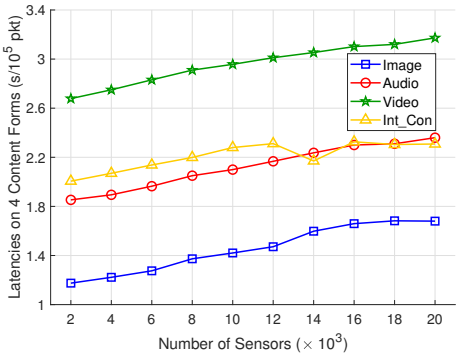
Second in Fig. 6b, we also can obtain the performance of our closed-loop feedback based scheduling method. Growth trends of packets in three real-time levels are alike that tend to fluctuate narrowly after climbing. The gap between $Lv.1$ and $Lv.2\&3$ is relatively stable that about 0.2 more second per gigabyte. As a result, function of closed-loop feedback does help multimedia packets in higher real-time level gain advantage in reducing latencies.

Except direct time requirement, in multimedia big data computing we also pay attention to performance in different content forms with respective packet size scopes.

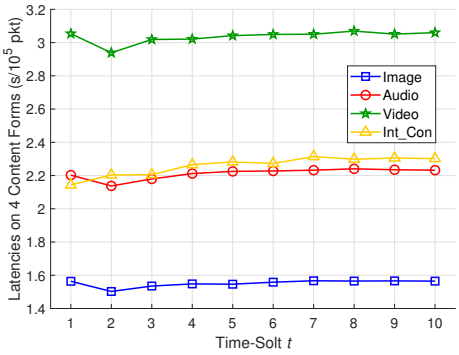
Fig. 7 gives the latency results of four content forms in sensor number and time-slot. In both Fig. 7a and 7b, the order of latency values from high to low is *Image* (blue), *Audio* (red), *Video* (green) and *Int_Con* (yellow) which means

TABLE III
AVERAGE PACKET SIZE OF FOUR CONTENT FORMS

Content Form	Packet number	Average Packet Size (byte)
image	277237475	528.09
audio	274960500	1020.09
video	266098575	1650.64
interactive content	280357075	1169.37



(a) Sensor Number Result



(b) Time-Solt Result

Fig. 8. Latencies on Packet Number of Four Content Forms in In-Memory Method

dealing with the same amount of data, larger packet size can save more time. As shown in Fig. 7, broken line of *Image* with the smallest average packet size rises very high to 3.2/3 second per gigabyte while the other three lines stay relatively close to each other. Packets in interactive content form have the widest packet size scope in wireless environment, both the average packet size and its position in figure are between *Video* and *Audio*. That is, growth patterns of three forms in two subfigures yield to the average packet size order. The small valley at 14,000 sensors in Fig. 7a also confirms the previous analysis since only packets in *Int_Con* own real-time level 1.

Finally, to verify the analysis about average value of packet size in four content forms and look for new experimental conclusions, we add the statistical results by packet numbers.

In Fig. 8, total latencies added up by number of generated packets show different pattern in four content forms compared to Fig. 8. The order of latency numerical values in figure is reversed which means packets in larger size do need more time to get transmitted and processed. Rising trends in Fig. 8a

shows that when there are more sensors competing for infinite computer resources, any content form has to cost more time. Moreover, gaps between either 2 of them in Fig. 8a and Fig. 8b are distinct and stable which even correspond with the average size differences of three forms in Table III. Lastly, higher real-time level earns no advantages in reducing latencies for same number of packets more than right half of Fig. 8b with sensor number more than 14,000. When amount of data is large enough, our in-memory based scheduling method can lay more emphasis on interactive content packets.

VI. Conclusion

In this paper, we focus on a real-time awareness scheduling method using in-memory processing to integrate the finite computing resources and allocate to multimedia data according to different real-time requirements. With great advantage in maximum transfer rate, in-memory processing can earn large space for continuous growth of multimedia big data. Based on this point, our work starts from a designed 3-tier IoT network model including a mass of multimedia sensors for data collection and sending, edge devices and central servers for data processing in different methods. We adopt the idea of closed-loop feedback in scheduling method aiming to continually allocate the varied workload to the finite in-memory processing capacities of edge devices and servers. In simulation part, we respectively compare the latencies of end-to-end, processing and their addition as well as real-time levels and content forms of multimedia data. From analyzing the experiment results, we prove that in-memory processing can greatly reducing total latencies dealing with multimedia big data and our proposed scheduling method achieves the rational workload allocation in facing different amount of data. In consideration of packets in different real-time levels and content forms, the method also can make corresponding priorities.

In the future, we will consider more details memory hierarchy and computational model inside device. The simulation part also needs to be designed more specific and use more metrics to analyze the performance in different experiment settings.

Acknowledgment

This work is partially supported by JSPS KAKENHI Grant Number JP16K00117, JP15K15976, and KDDI Foundation. Mianxiong Dong is the corresponding author.

References

[1] G. Mulligan, "The 6LoWPAN Architecture," *Proceedings of the 4th Workshop on Embedded Networked Sensors*, pp. 78-82, Jun. 2007.

[2] A. V. Dastjerdi and R. Buyya, "Fog Computing: Helping the Internet of Things Realize Its Potential," *Computer*, vol. 49, no. 8, pp. 112-116, Aug. 2016.

[3] H. Zhang, G. Chen, B. C. Ooi, K. L. Tan and M. Zhang, "In-Memory Big Data Management and Processing: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1920-1948, Jul. 2015.

[4] Y. Tian, S. C. Chen, M. L. Shyu, T. Huang, P. Sheu and A. D. Bimbo, "Multimedia Big Data," *IEEE MultiMedia*, vol. 22, no. 3, pp. 93-95, Jul. 2015.

[5] C. Hu, Z. Xu, Y. Liu, L. Mei, L. Chen and X. Luo, "Semantic Link Network-Based Model for Organizing Multimedia Big Data," *IEEE Transactions on Emerging Topics in Computing*, vol.2, no. 3, pp. 376-387, Sep. 2014.

[6] S. Ehsan and B. Hamdaoui, "A Survey on Energy-Efficient Routing Techniques with QoS Assurances for Wireless Multimedia Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. 14, no. 2, pp. 265-278, 2012.

[7] K. Kambatla, G. Kollias, V. Kumar and A. Grama, "Trends in Big Data Analytics," *Journal of Parallel and Distributed Computing*, vol. 74, no. 7, pp. 2561-2573, Jul. 2014.

[8] Y. Li, K. Gai, Z. Ming, H. Zhao and M. Qiu, "Intercrossed Access Controls for Secure Financial Services on Multimedia Big Data in Cloud Systems," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 4s, pp. 67:1-67:18, Nov. 2016.

[9] Y. Zhang, M. Qiu, C. W. Tsai, M. M. Hassan and A. Alamri, "Health-CPS: Healthcare Cyber-Physical System Assisted by Cloud and Big Data," *IEEE Systems Journal*, vol. 11, no. 1, pp. 88-95, Mar. 2017.

[10] K. Ota, M. S. Dao, V. Mezaris and F. G. B. De Natale, "Deep Learning for Mobile Multimedia: A Survey," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13(3s), no. 34, Jul. 2017.

[11] A. Ahmad, M. A. Rahman, B. Sadiq, S. Mohammed, S. Basalamah and M. R. Wahiddin, "Visualization of a Scale Free Network in a Smartphone-Based Multimedia Big Data Environment," *2015 IEEE International Conference on Multimedia Big Data (BigMM)*, pp. 286-287, Apr. 2015.

[12] Y. Sun, H. Luo and S. K. Das, "A Trust-Based Framework for Fault-Tolerant Data Aggregation in Wireless Multimedia Sensor Networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 785-797, Nov. 2012.

[13] H. Luo, J. Luo, Y. Liu and S. K. Das, "Adaptive Data Fusion for Energy Efficient Routing in Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 55, no. 10, pp. 1286-1299, Oct. 2006.

[14] H. Li, K. Ota, M. Dong, A. Vasilakos and K. Nagano, "Multimedia Processing Pricing Strategy in GPU-Accelerated Cloud Computing," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1-1, Feb. 2017.

[15] H. Li, K. Ota, M. Dong and M. Guo, "Mobile Crowdsensing in Software Defined Opportunistic Networks," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 140-145, Jun. 2017.

[16] J. Lee, H. A. Kao and S. Yang, "Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment," *Procedia CIRP*, vol. 16, pp. 3-8, 2014.

[17] S. Robbins, "RAM is the New Disk...," [Online]. Available: www.infoq.com/news/2008/06/ram-is-disk, accessed Jul. 2017.

[18] L. Liu, X. Zhang and H. Ma, "Optimal Node Selection for Target Localization in Wireless Camera Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 7, pp. 3562-3576, Sep. 2010.

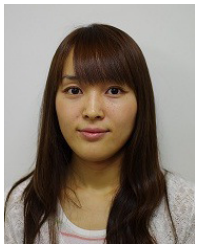
[19] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury and A. T. Campbell, "A Survey of Mobile Phone Sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140-150, Sep. 2010.

[20] C. Bettstetter, G. Resta and P. Santi, "The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 257-269, Jun. 2003.

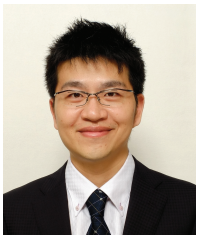
[21] S. Sarkar, S. Chatterjee and S. Misra, "Assessment of the Suitability of Fog Computing in the Context of Internet of Things," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1-1, Oct. 2015.

[22] T. Camp, J.Boleng and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483-502, Aug. 2002.

Jianwen Xu received the B.Eng degree in Electronic and Information Engineering from Dalian University of Technology (DLUT), China, in 2014, and M.Eng degree in Information and Communication Engineering from Shanghai Jiaotong University (SJTU), China, in 2017. He is currently pursuing the Ph.D. degree in Electrical Engineering at Muroran Institute of Technology, Japan. His main fields of research interest include distributed system, Internet of things.



Kaoru Ota was born in Aizu-Wakamatsu, Japan. She received M.S. degree in Computer Science from Oklahoma State University, USA in 2008, B.S. and Ph.D. degrees in Computer Science and Engineering from The University of Aizu, Japan in 2006, 2012, respectively. She is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. From March 2010 to March 2011, she was a visiting scholar at University of Waterloo, Canada. Also she was a Japan Society of the Promotion of Science (JSPS) research fellow with Kato-Nishiyama Lab at Graduate School of Information Sciences at Tohoku University, Japan from April 2012 to April 2013. Her research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems. Dr. Ota has received best paper awards from ICA3PP 2014, GPC 2015, IEEE DASC 2015, IEEE VTC 2016-Fall, FCST 2017 and IET Communications 2017. She is an editor of IEEE Transactions on Vehicular Technology (TVT), IEEE Communications Letters, Peer-to-Peer Networking and Applications (Springer), Ad Hoc & Sensor Wireless Networks, International Journal of Embedded Systems (Inderscience) and Smart Technologies for Emergency Response & Disaster Management (IGI Global), as well as a guest editor of ACM Transactions on Multimedia Computing, Communications and Applications (leading), IEEE Communications Magazine, IEEE Network, etc. Also she was a guest editor of IEEE Wireless Communications (2015), IEICE Transactions on Information and Systems (2014), and Ad Hoc & Sensor Wireless Networks (Old City Publishing) (2014). She was a research scientist with A3 Foresight Program (2011-2016) funded by Japan Society for the Promotion of Sciences (JSPS), NSFC of China, and NRF of Korea. She is the recipient of IEEE TCSC Early Career Award 2017.



Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently an Associate Professor in the Department of Information and Electronic Engineering at the Muroran Institute of Technology, Japan. He was a JSPS Research Fellow with School of Computer Science and Engineering, The University of Aizu, Japan and was a visiting scholar with BBCR group at University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong was selected as a Foreigner Research Fellow (a total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. His research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems. He has received best paper awards from IEEE HPCC 2008, IEEE ICESS 2008, ICA3PP 2014, GPC 2015, IEEE DASC 2015, IEEE VTC 2016-Fall, FCST 2017 and 2017 IET Communications Premium Award. Dr. Dong serves as an Editor for IEEE Transactions on Green Communications and Networking (TGCN), IEEE Communications Surveys and Tutorials, IEEE Network, IEEE Wireless Communications Letters, IEEE Cloud Computing, IEEE Access, as well as a leading guest editor for ACM Transactions on Multimedia Computing, Communications and Applications (TOMM), IEEE Transactions on Emerging Topics in Computing (TETC), IEEE Transactions on Computational Social Systems (TCSS). He has been serving as the Vice Chair of IEEE Communications Society Asia/Pacific Region Meetings and Conference Committee, Leading Symposium Chair of IEEE ICC 2019, Student Travel Grants Chair of IEEE GLOBECOM 2019, and Symposium Chair of IEEE GLOBECOM 2016, 2017. He is the recipient of IEEE TCSC Early Career Award 2016, IEEE SCSTC Outstanding Young Researcher Award 2017 and The 12th IEEE ComSoc Asia-Pacific Young Researcher Award 2017.