

# Mobile Network Architectures and Context-Aware Network Control Technology in the IoT Era

Takanori IWAI<sup>†,††a)</sup>, *Member*, Daichi KOMINAMI<sup>†††</sup>, Masayuki MURATA<sup>††††</sup>, Ryogo KUBO<sup>††</sup>,  
and Kozo SATODA<sup>†</sup>, *Nonmembers*

**SUMMARY** As IoT services become more popular, mobile networks will have to accommodate a wide variety of devices that have different requirements such as different bandwidth limitations and latencies. This paper describes edge distributed mobile network architectures for the IoT era based on dedicated network technology and multi-access edge computing technology, which have been discussed in 3GPP and ETSI. Furthermore, it describes two context-aware control methods that will make mobile networks on the network architecture more efficient, reliable, and real-time: autonomous and distributed mobility management and bandwidth-guaranteed transmission rate control in a networked control system.

**key words:** mobile network, internet of things, dedicated network, multi-access edge computing, context-aware control

## 1. Introduction

The shape of the fifth generation (5G) of mobile network is receiving more attention because of the explosive growth of Internet of Things (IoT) devices and the popularity of high-performance smartphones and tablets. The requirements for future mobile networks are as follows [1], [2]: (1) a thousand-fold increase in traffic capacity and (2) an approximate hundred-fold increase in device connections in comparison with what is available currently; (3) a transmission rate of 10 Gbps during peak periods and 100 to 1000 Mbps during normal periods; (4) a latency of less than 1 msec. A number of companies have published white papers on 5G wireless technology [3], and studies of novel mobile network architectures including those of core networks are in progress at standards bodies such as the 3rd Generation Partnership Project (3GPP) and European Telecommunications Standards Institute (ETSI).

This paper includes the following sections: Section 2 describes technological prospects of mobile network architectures for accommodating a wide variety of IoT devices and our concept of context-aware control. Sections 3 and 4

describe case studies on the context-aware control method described in Sect.2. In particular, Sect.3 describes autonomous and distributed mobility management technology that achieves load balancing between network equipment while reducing its delay. Section 4 describes wireless distributed sensor-actuator network control technology that improves quality of performance (QoP)/quality of control (QoC) in control systems. The Sect. 5 is the conclusion.

## 2. Network Architecture in the IoT Era

In this section, we describe the elemental technologies related to mobile network architectures that have been discussed in 3GPP and ETSI for accommodating a wide variety of IoT devices and prospects for these architectures. Furthermore, we describe our proposed migration scenario of mobile architecture and concept of context-aware control method.

### 2.1 Elemental Technologies Related to Network Architectures for IoT

Three technologies are described below: dedicated network technology that improve the efficiency and reliability of networks, multi-access edge computing (MEC) technology that enhances the real-time performance of networks, and cooperative control technology that increases the affinity between networks and services. The relation between the three technologies and IoT architecture is shown in Fig. 1.

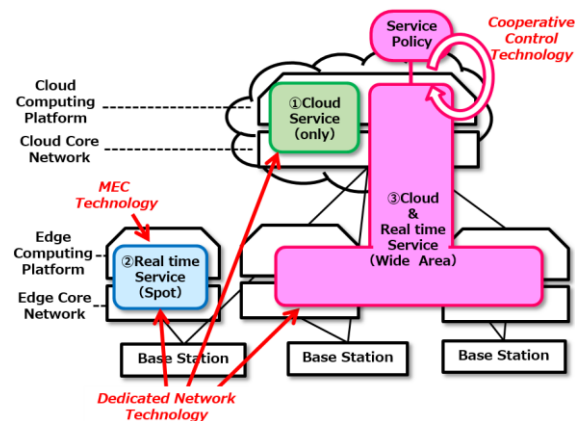


Fig. 1 Network architecture in the IoT-era.

Manuscript received December 8, 2017.

Manuscript revised February 11, 2018.

Manuscript publicized April 13, 2018.

<sup>†</sup>The authors are with the System Platform Research Laboratories, NEC Corporation, Kawasaki-shi, 211-8666 Japan.

<sup>††</sup>The authors are with the Department of Electronics and Electrical Engineering, Keio University, Yokohama-shi, 223-8522 Japan.

<sup>†††</sup>The author is with the Graduate School of Economics, Osaka University, Toyonaka-shi, 560-0043 Japan.

<sup>††††</sup>The author is with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan.

a) E-mail: t-iwai@hx.jp.nec.com

DOI: 10.1587/transcom.2017NEI0001

### 2.1.1 Dedicated Network Technology

The characteristics of IoT services are quite different from those of conventional mobile services, including smartphones, in terms of network requirements, communication intervals of devices, and frequency of movement. The consequent heterogeneity of services in one network poses a risk of degrading the efficiency and reliability of that network.

As a way of solving this problem, it is thought that separating and dedicating networks to individual services can minimize the influences between services and efficiently orchestrate services that have different bandwidth limitations, latency requirements, etc. We will discuss two technologies in particular: network slicing, which can make separate networks for each service, and dedicated core networks (DCNs), which can select a destination network for a device in accordance with the services in use and their characteristics.

### Network Slicing

Network slicing is a technology that creates a logical network (a slice) on the network virtualization infrastructure for each requirement, such as quality, security, and protocol, and allocates slices to individual services [4]. There have been numerous technological developments that work in this way by virtualizing the core network equipment of mobile networks, such as Mobility Management Entity (MME), Serving Gateway (S-GW), and Packet Data Network Gateway (P-GW) with Network Functions Virtualization (NFV) and by using Software-Defined Networks (SDNs) for control [5].

### Dedicated Core Networks

Dedicated Core Networks (DCN) is a technology for mobile networks with multiple separated core networks that decides a route of a device to a destination core network on the basis of its service type [6]. Concretely speaking, by adding new information called UE Usage Type to the subscriber information managed by the Home Subscriber Server (HSS), it becomes possible to route User Equipment (UE) to a core network corresponding to the UE Usage Type. Another advantage of DCN is that it has less impact on the existing 3GPP specifications, so that it would be relatively easy to introduce.

### 2.1.2 MEC Technology

Real-time services such as driving support, automated driving, and remote robot control will likely be widespread in the IoT era, meaning that faster and lower-latency mobile networks are needed [7]. For example, 5G networks will require transmission latencies of less than 1 ms in the wireless segment [8]. One of the challenges to building a lower latency mobile network is the distances of the communication paths between base stations and the core network equipment.

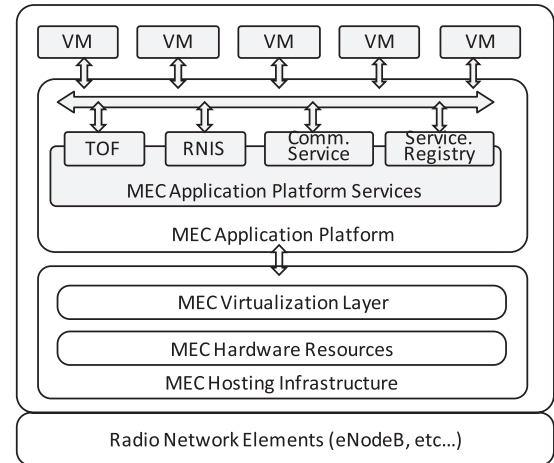


Fig. 2 Architecture of MEC.

Here, MEC has been discussed as a way to make these paths shorter.

### MEC

MEC achieves lower latencies by deploying computing infrastructures that process application data near base stations. ETSI MEC, an industry specification group (ISG), is currently carrying out a study that supposes the architecture depicted in Fig. 2 [7]. This MEC architecture is roughly classified into a MEC Hosting Infrastructure, which is a virtualization infrastructure, and a MEC Application Platform, which manages the services on the virtualization infrastructure. ETSI MEC ISG intends to standardize the MEC Application Platform, the virtualization infrastructures, and the network cooperation mechanism by coordinating with other organizations such as 3GPP.

### 2.1.3 Cooperative Control Technology

To provide communications that can be adapted to a wide variety of IoT services, mobile networks must be able to comprehend services and be flexible. As a means of achieving this goal, 3GPP has discussed the architecture and process method called Architecture Enhancements for Service capability Exposure (AESE).

### Service Capability Exposure

AESE formulates the information to be shared and the processing procedure for implementation of a cooperative control between services and networks in a framework called the Service Capability Exposure Framework [9], [10]. AESE has more than ten functions: as an example, a function called the Communication Pattern Parameters Provisioning Procedure leverages the communication pattern of a device for optimum network processing by sending it to a service and registering it as UE Context information of the HSS.

## 2.2 New Architectures for the IoT Era

To accommodate huge numbers IoT devices, we propose a novel mobile network architecture and a migration scenario that combines dedicated network technology, edge computing technology, and cooperative technology for the networks and services mentioned above.

### 2.2.1 Architectures in the IoT Era

One architecture for accommodating a huge number and a wide variety of IoT devices is shown in Fig. 1. It uses MEC technology for distributed deployment of network facilities including computing functions on the edge of mobile networks near base stations. Furthermore, it virtually allocates network facilities to the areas that need them for each service by dedicating networks. By cooperating with services, it optimizes the processes in the dedicated networks in accordance with the service requirements. These features provide optimized networks that meet a diverse range of requirements including the real-time limitations of various IoT services, while at the same time limiting the influences of services on each other.

### 2.2.2 Migration Scenario for Architectures

For migration to our mobile network architecture of the IoT era, it is important to develop it step by step while continuing existing services. Here, we propose a three-step migration scenario.

The first step is to make a dedicated network for IoT devices that is separate from the networks of conventional mobile services such as for smartphones, as shown in Fig. 3. This enables network optimization for each service while limiting influences on other services. This architecture can be realized with the above-mentioned network slicing and DCN technology.

The second step is to make a dedicated network for spot-type real-time services by introducing MEC locally, as shown in Fig. 4. Networking and computing infrastructures for services requiring real-time capabilities can be provided locally, and these services can be separated from existing ones as in the first step. The reasons why the spot type is used are as follows: the possibility of mobility in the application layer including mobility of the IP layer and computing is under discussion, and it is easy to introduce spot-type networks in a MEC environment that requires no mobility [1]. Moreover, by cooperating with Cloud radio access networks (C-RANs), the spot can cover a wider area.

The last step is to make efficient networks for real-time services offered over a wide area through MEC-to-MEC and MEC-to-cloud cooperation, as shown in Fig. 5. By developing mobility technology in the second step, real-time services can be deployed over wide areas, while efficiency and reliability can be enhanced system-wide by MECs and clouds sharing network and computing resources.

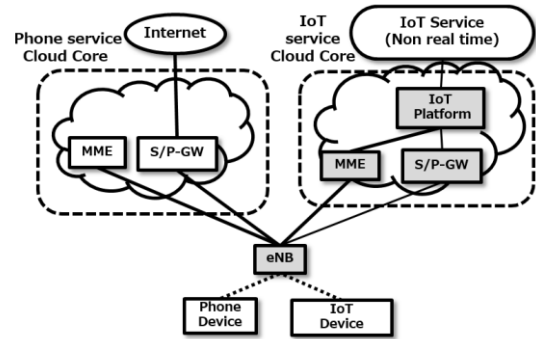


Fig. 3 First step of migration (separation of networks).

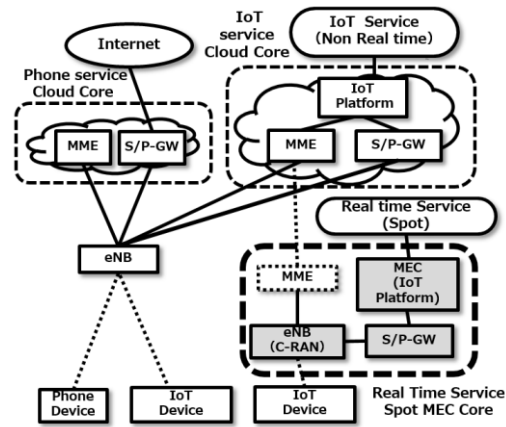


Fig. 4 Second step of migration (introducing MEC locally).

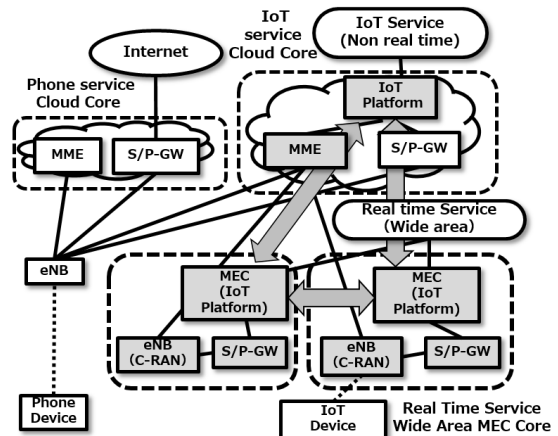


Fig. 5 Third step of migration (MEC and Cloud cooperation).

### 2.2.3 Trend of Technology toward the IoT Era

In addition to migration of architectures, it will be necessary to optimize the network processing dynamically on the basis of the characteristics of each device so that mobile networks can accommodate massive numbers of various IoT devices. Here, the distributed functions and increase in management information (parameters) that will be the result of introducing DCNs and MEC will make mobile networks more com-

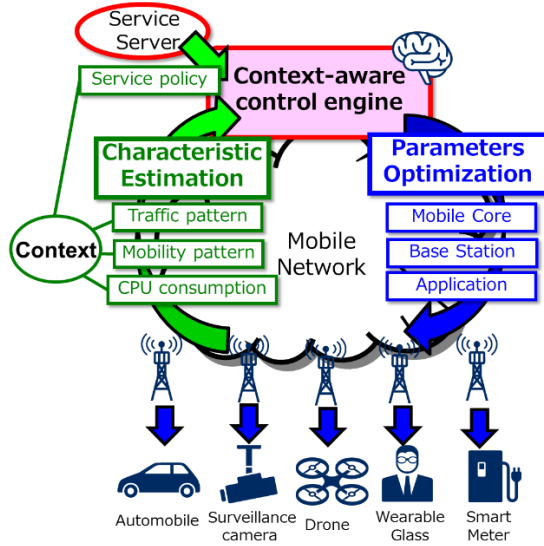


Fig. 6 Concept of context-aware control.

plicated. We therefore need to adapt complicated network environments to the numerous real-time services that will be available in the future. Since AESE only provides processing agreements for a portion of the services that can be expected, more flexible technology is needed for dynamic optimization of network processing according to AESE specifications.

As shown in Fig. 6, the authors comprehend the various characteristics (contexts) of communication and mobility of IoT devices to make more efficient, reliable, and real-time mobile networks for IoT. We furthermore want to promote research on context-aware methods that control mobile networks adaptively according to their characteristics [11], [12].

In Sects. 3 and 4, we describe two technological case studies on context-aware control. The technologies aim to reduce communication delays and improve CoP/QoC, which will be important in future IoT services.

### 3. Autonomous and Distributed Mobility Management

We describe an autonomous and distributed mobility control technology, called context-aware control for MMEs. We have proposed a network architecture that places MMEs as logical functions on the mobile network [13]. MMEs autonomously manage UE in a distributed manner, and for this reason, we call them autonomous distributed MMEs (ADMMEs). In this section, we show that the efficiency and stability of the ADMME-based system can be improved by taking into consideration of the mobility characteristics of IoT devices and UE as context information.

#### 3.1 ADMME Selection Based on the Attractor-Selection Model

An ADMME can run on any node of the core networks. The role of the mobility management for UE can be delegated by one ADMME to another ADMME (*ADMME switching*),

and the selection of the new ADMME is made at the current ADMME's discretion (*ADMME selection*). ADMME switching can happen when a UE transmits a tracking area update (TAU) request, a handover request, or an attach request to the current ADMME. When the ADMME receives such a request from the UE, it determines another ADMME to transfer the role of managing the requesting UE to so as to reduce the communication delay between the UE and the new ADMME and to reduce the load concentrated on the new ADMME. It selects the new ADMME from a set of candidates that consists of the ADMMEs in all nodes on the path between the requesting UE and itself. In addition, the candidate set includes ADMMEs in the S-GW and the P-GW nearest the UE.

The ADMME selects a new ADMME from the candidate set according to a *delay history* and the *load status* of the nodes (as described in the following section). We assume that the communication delay is estimated upon request from the UE by using a node that has an ADMME. Each ADMME periodically collects the load statuses of the nodes in the current candidate sets. After selection of the new ADMME, a current ADMME transmits a delegation message with the context information of the requesting UE to the new ADMME. The new ADMME then sends the response message for the UE's request.

For the ADMME selection, we use the attractor selection model, which mathematically describes biological systems that adapt themselves to unexpected changes in their surroundings [14]. In this method, an ADMME has a vector  $\mathbf{m} = (m_1, m_2, \dots, m_M)$  for each UE that it manages.  $M$  is the cardinality of the candidate set for the corresponding UE.  $m_i$  is a state value that corresponds to the adequacy of selecting ADMME  $i$ . The ADMME also has a scalar value  $\alpha$ , called *activity*, for each UE, which expresses the goodness of the current selection of an ADMME. In the attractor selection algorithm,  $\mathbf{m}$  is updated using the following equation, where  $f$  is the derivative of a function that has  $M$  attractors and  $\eta$  is Gaussian noise.

$$\frac{dm_i}{dt} = \alpha f(\mathbf{m}) + \eta \quad (1)$$

When  $\alpha$  is high,  $\mathbf{m}$  converges to an attractor state, and when  $\alpha$  is low, it randomly seeks another state. Each ADMME updates its activity when a request from a UE arrives; it also uses the activity to update  $\mathbf{m}$ . The candidate ADMME with the largest state value is selected as a new ADMME. The state vector, activity, and delay history are transmitted to the new ADMME, which then manages this information. For the  $h$ th request, an ADMME calculates its activity as follows:

$$\alpha(h) = \rho \alpha_d(h) + (1 - \rho) \alpha_l(h) \quad (2)$$

where  $\rho$  is a parameter from 0 to 1 for determining the weight of  $\alpha_d(h)$  and  $\alpha_l(h)$ . In the attractor selection, the ADMME estimates the expected communication delay, denoted by  $\hat{d}$ , from node  $i$  to UE via itself. If node  $i$  is on the shortest path between the UE and the ADMME,  $\hat{d}$  is the sum of two

one-way delays: from the UE to the ADMME and from the ADMME to node  $i$ . Otherwise,  $\hat{d}_i = 0$ . Delay information during the past  $W$  steps is stored as a delay history. For this, each ADMME periodically sends probe packets to nodes in its candidate set. This is done while collecting the load statuses of the nodes in the current candidate sets. Note that in the candidate set, node  $i$  is closer to the corresponding UE as  $\hat{d}_i$  becomes larger.  $\alpha_d(h)$  is the activity based on  $\hat{d}$  and is calculated as

$$\alpha_d(h) = \left( \frac{\sum_{k=1}^W \frac{\hat{d}_{cm}(h-k)}{k}}{\max_{1 \leq i \leq M} \sum_{k=1}^W \frac{\hat{d}_i(h-k)}{k}} \right)^\varepsilon \quad (3)$$

where  $\hat{d}_{cm}$  is the delay of the current ADMME and  $\varepsilon$  is a parameter to determine the output level for  $\alpha_d(h)$ .  $\alpha_l(h)$  is based on the load status, which is the number of UEs that the corresponding ADMME manages, and it is calculated as

$$\alpha_l(h) = \frac{\min_{1 \leq i \leq M} l_i(h)}{l_{cm}(h)} \quad (4)$$

where  $l_i(h)$  is the latest load status of node  $i$  for the  $h$ th request, and  $l_{cm}$  is the load of the current ADMME. Using the  $\alpha(h)$  calculated from  $\alpha_d(h)$  and  $\alpha_l(h)$ , the ADMME updates  $\mathbf{m}$  as follows:

$$\frac{dm_i}{dt} = \frac{s(\alpha(h))}{1 + m_{max}^2 - m_i^2} - \alpha(h)m_i + \eta \quad (5)$$

where  $m_{max} = \max_{1 \leq j \leq M} \{m_j\}$ .  $s(\alpha)$  determines the scale of the state value. We use the same definition of  $s(\alpha)$  as in [14], that is,  $s(\alpha) = \alpha(h) \left( \beta \alpha(h)^\gamma + \frac{1}{\sqrt{2}} \right)$ . In Eq. (5), when  $\eta = 0$  and  $\alpha(h) = 0$ , the right-hand side is a function that has  $M$  attractors where only one element of  $\mathbf{m}$  becomes higher in value than the other elements.

### 3.2 ADMME Control from an External Node

The above attractor selection has difficulty dealing with instability near local optima. Connection switching between UEs and ADMMEs increases the C-Plane load, and handover causes delays. To solve these problems, we propose an architecture for controlling such systems, where a new activity function is defined so that a system manager can stabilize the system.

The activity  $\alpha$  in the attractor selection algorithm expresses the goodness of the current state of an ADMME; therefore,  $\alpha$  determines the behavior of the ADMME. To ensure system stability, it is necessary to allow a network manager to maximize  $\alpha$ . For this, we use a sigmoid function as a new activity function  $\alpha_{new}$ :

$$\alpha_{new} = \max \left( \alpha, \frac{1}{1 + e^{-g(\alpha - \alpha_{th})}} \right) \quad (6)$$

where  $g$  is a non-negative parameter that determines the decay characteristics of the sigmoid function. The sigmoid function decays more quickly as  $g$  gets bigger.  $\alpha_{th}$  is the inflection point of the function. Network managers can control this threshold parameter  $\alpha_{th}$ , to make the system stable.

However, it is difficult to determine the best value of  $\alpha_{th}$  beforehand. When  $\alpha_{th}$  is comparatively high, frequent ADMME switching may happen. On the other hand, when  $\alpha_{th}$  is comparatively small, an ADMME may stop with a worse choice, which degrades system performance. To determine the best  $\alpha_{th}$ , we introduce an external node that manages  $\alpha_{th}$ . This control node monitors the performance of the whole network and gives control inputs to  $\alpha_{th}$  in each ADMME.

### 3.3 Context-Aware Control for Mobile Users

In our previous work, we found that the best value for  $\alpha_{th}$  depends on the frequency of the user's handover due to its mobility. When users do not move much, our external control increases delay and degrades load-balancing performance. On the other hand, when users frequently move, it significantly reduces the amount of ADMME switching without losing much performance. Therefore, we should give control inputs to nodes whose ADMME frequently delegates the role of managing UE. Otherwise, ADMMEs should behave in an autonomous and distributed manner.

Here, we describe an evaluation of our proposal by using computer simulation. It shows the effectiveness of context-aware control in comparison with a simple control in which an external controller gives the same input to all ADMMEs.

We assume a core network consisting of one P-GW and four S-GWs. Each S-GW corresponds to one tracking area (TA), and each TA is made up of 37 hexagonal cells, as shown in Fig. 7. Each cell has one eNodeB, so the number of eNodeBs is 148. For the sake of simplicity, delays between nodes are static, i.e., 2 ms between UE and an eNodeB ( $\Delta_0$ ), 20 ms between an eNodeB and an S-GW ( $\Delta_1$ ), and 3 ms between an S-GW and a P-GW ( $\Delta_2$ ). The ADMMEs in each eNodeB, S-GW, and P-GW number 1, 5, and 5, respectively. In our method, one S-GW and one P-GW are always included in the ADMME candidate. Since the number of S-GW and P-GW is fewer than that of eNodeB, the load tends to concentrate on these nodes. Therefore, the number of these ADMMEs is set to be larger. At the beginning of the simulation, 100 UEs are deployed in each cell, and they connect to an ADMME on the nearest S-GW; that is, each ADMME on a S-GW has 740 UEs. The TAU timer of every UE is set to 30 min.

First, we simulated two scenarios; each UE moves to one neighboring cell every 100 min or all UEs do not move. When the UE moves, one of the neighboring cells is selected with equal probability. For the parameters of the attractor selection in Eq. (3) and Eq. (5), we use the same values used in [15], such as  $\beta = 10$ ,  $\gamma = 10$ ,  $\varepsilon = 2$ . The other parameters are as follows:  $W=5$ ,  $g=30$ , and  $\rho=0.2$  or  $0.8$ .

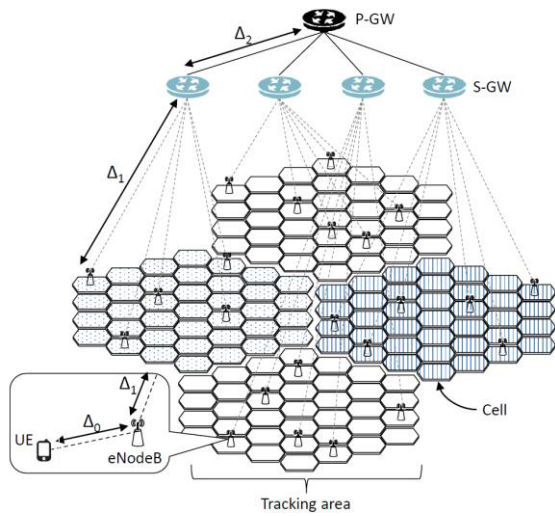


Fig. 7 Network model.

Figure 8 shows the average delay and Jain's fairness index when we give the same magnitude of input. As for the average delay, we measure the average round-trip delay time between all UEs and their corresponding ADMME. Jain's fairness index (FI) is used to evaluate how well our load-balancing mechanism works. FI is defined as follows:

$$FI = \frac{(L_e + L_S + L_P)^2}{3(L_e^2 + L_S^2 + L_P^2)}$$

where  $L_e$ ,  $L_S$ , and  $L_P$  are the average load statuses of eNodeBs, S-GWs, and P-GWs, respectively. In this paper, they correspond to the average number of UEs that the ADMME in an eNodeB, an S-GW, or a P-GW manages.

The attractor selection mechanism preferentially seeks the state that gives the minimum delay when  $\rho$  is 0.8 and seeks the state that achieves a balanced load when  $\rho$  is 0.2. Therefore, we will focus on the result with  $\rho=0.8$  when evaluating the delay and focus on the result with  $\rho=0.2$  when evaluating the fairness index.

As the control force increases (i.e.,  $\alpha_{th}$  decreases), ADMME selection tends to stop, which increases the delay as shown in Fig. 8(a), except for the case where UEs do not move and  $\rho=0.8$ . This is because in that case, the autonomous ADMME selection mechanism gets stuck in local optima with large  $\alpha_{th}$ . Figure 8(b) shows that fairness index decrease or does not change much as the control force increases when  $\rho=0.2$ . As well as the case in Fig. 8(a), a smaller  $\alpha_{th}$  tends to make ADMME selection stop, which decreases the fairness index. As the control force increases, the amount of ADMME switching decreases in Fig. 8(c). These results indicate that a stronger control input (a smaller  $\alpha_{th}$ ) should be given to nodes in which ADMME switching occurs more frequently. On the other hand, a weaker control input is needed when UEs do not move frequently.

Next, we consider mobility scenarios in which half of the UEs move with a sojourn time of 100 min and the other half do not move. These mobile UE moves within a 2-

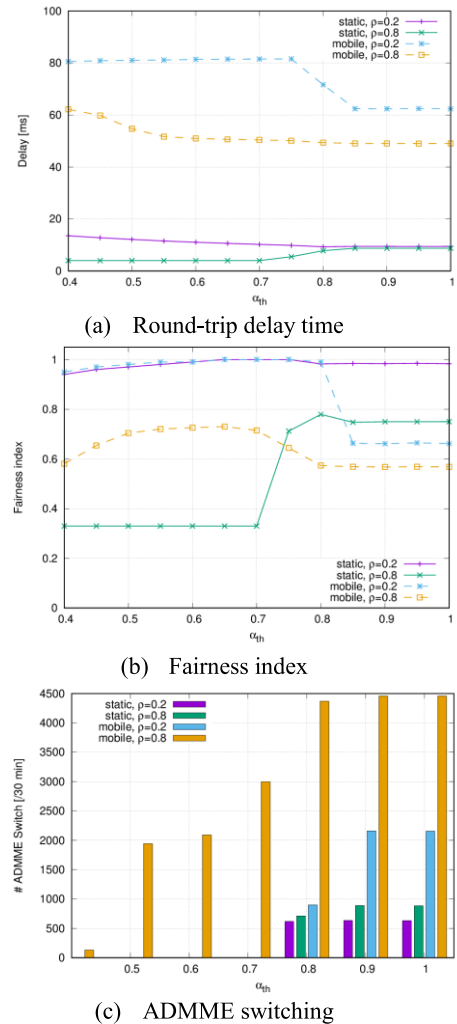


Fig. 8 Basic performance of ADMME control.

hop distance away from the gray-colored cell in Fig. 7. We consider two scenarios, one in which the same magnitude of the control force is provided to all ADMMEs and another in which comparatively strong control force ( $\alpha_{th} = 0.5$ ) is provided to the ADMMEs on the eNodeBs in the gray cells, on all S-GWs, and on all P-GWs, and all the other ADMMEs receive comparatively weak control force ( $\alpha_{th} = 0.8$ ).

Figure 9 shows result where the label "original" means the former scenario and "context-aware" means the latter one. This result show that the latter scenario performs better or nearly as good as the original one, while it can reduce the number of ADMME switching drastically. Therefore, our context (mobility)-aware control can improve the efficiency and stability of the system.

#### 4. Wireless Distributed Sensor-Actuator Networks

This section describes distributed control systems in wireless networks that include huge numbers of sensors and actuators. Networked control systems (NCSs), which require real-time communication, are based on MEC control architecture. In

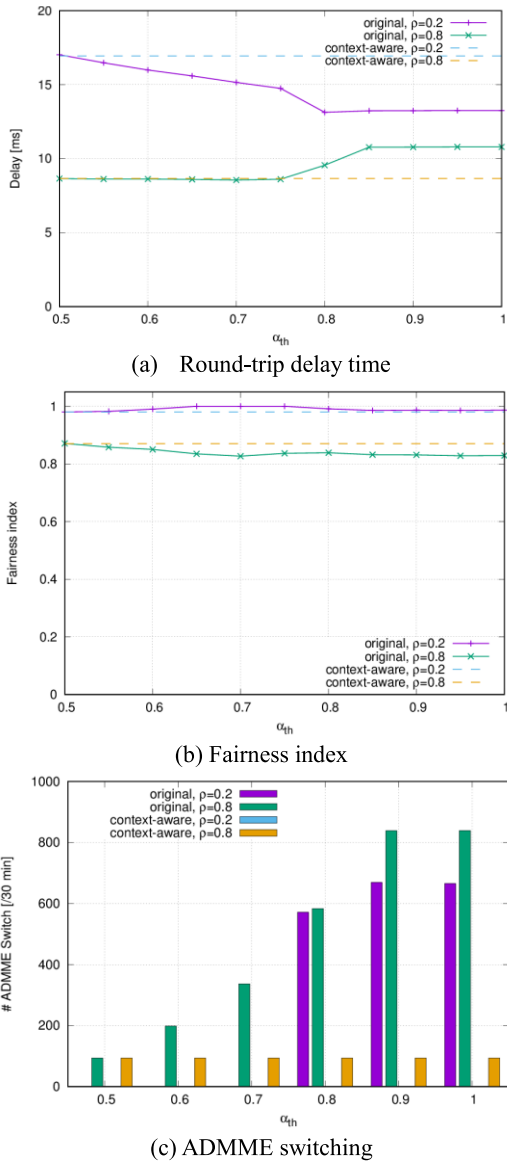
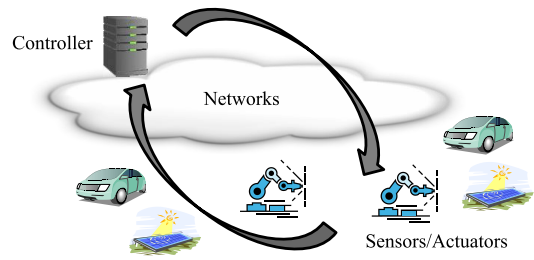


Fig. 9 Context-aware control.

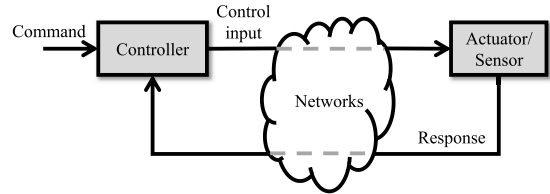
addition, co-design of the network and control systems improves both QoS in communication networks and QoP/QoC in control systems. As a co-design approach, we propose a transmission rate control for a wireless distributed control system. Simulation results confirm the effectiveness of this technique.

#### 4.1 Distributed Control Systems

An NCS is comprised of a controller, communication networks, sensors, and actuators, as shown in Fig. 10. In Fig. 10(a), the sensors and actuators are distributed in various systems and are connected to the controller through wireless and/or wired networks. For example, networked industrial robots in factories, generators/loads in power grids, and electric vehicles (EVs) in intelligent transport systems (ITSs) can be considered to be distributed sensor-actuator



(a) Concept of distributed sensor-actuator networks



(b) Block diagram of an NCS

Fig. 10 Distributed control over networks.

networks. As shown in Fig. 10(b), the command signal is input to the controller, and the response signal is sent from the sensor to the controller through the networks. The controller calculates the control input signal from the command and response signals, and it sends control input signals to the actuator through the networks. An NCS such as this can collect sensor data from different environments and control actuators in an integrated manner.

Various NCSs have been studied since the 1980s [16]. Gupta et al. [17] categorized NCS research into technologies that focus on “control of networks”, such as routing and congestion control on the network side, and those that exert “control over networks”, such as packet-delay and packet-loss compensation on the control side. Zhang et al. [18] presented network-induced constraints that should be considered when designing NCSs. These constraints are packet delays, packet losses, packet disorder, time-varying packet transmission/sampling intervals, competition of multiple nodes accessing network, data quantization, clock synchronization among local and remote nodes, network security, and network safety. In particular, packet delays in communication networks greatly affect the stability and performance of the NCSs, and many packet-delay compensation techniques have been studied in the field of control engineering.

#### 4.2 MEC-Based Real-Time Control Architecture

NCSs using wireless communication have begun to attract attention as more and more IoT devices, i.e., sensors and actuators, are deployed [19], [20]. The issues of wireless NCSs regard QoS, coverage, mobility, and security. Packet delays may degrade the stability and performance of networks. As shown in Fig. 11, it is difficult to maintain system stability by using a cloud-based control with an over 100-ms round-trip delay between the controller and sensor/actuator.

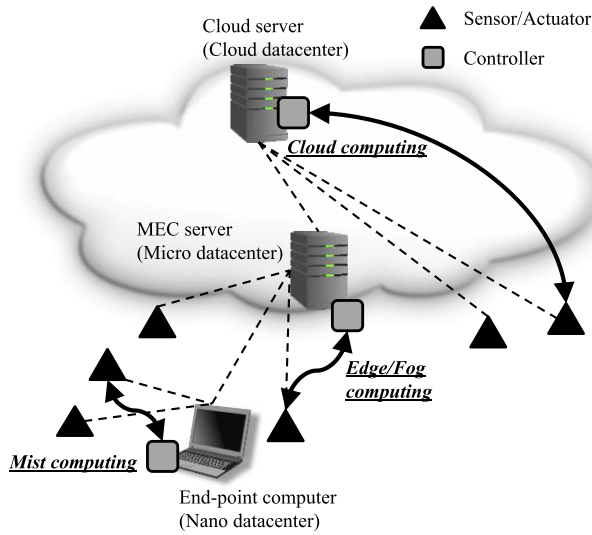


Fig. 11 Cloud, edge/fog, and mist computing for NCS applications.

On the other hand, edge/fog-based control using an MEC server reduces the round-trip delay and is suitable for real-time applications. Moreover, the recently suggested mist computing on the user side can improve the performance of NCSs. The Tactile Internet [21] is a next-generation network concept proposed by ITU-T, which achieves a 1-ms end-to-end latency. The applications of the Tactile Internet include robotics, telepresence, virtual reality, augmented reality, healthcare, road traffic, and smart grids.

There is a trade-off between delay and locality. The coverages of the MEC server and end-point computer are subject to the limitation of radio wave range. This means that we have to allocate controllers to various computing resources appropriately in accordance with the required QoS and coverage area. To apply edge/fog computing to mobile sensors and actuators, we have to develop a handover technique for computing resources. Security issues such as countermeasures against cyberattacks have to be resolved before wireless NCSs can be of practical use.

#### 4.3 Co-Design of Network and Control Systems

Co-design of network and control systems is necessary because the NCS is an integrated system for control of networks and control over networks. Wang et al. [22] proposed co-design of computer, communication, and control systems using a shared network. The sampling period of the NCSs affects control performance [23]. In digital controls, a smaller sampling period typically improves control performance. However, in an NCS using a shared network, an excessively small sampling period may degrade control performance, because the numerous signals generated may cause network congestion leading to large delays and packet losses. Priority scheduling based on error signals is one way to improve control performance [24]. Below, we describe a transmission rate control scheme for an NCS using a shared wireless network to improve control performance.

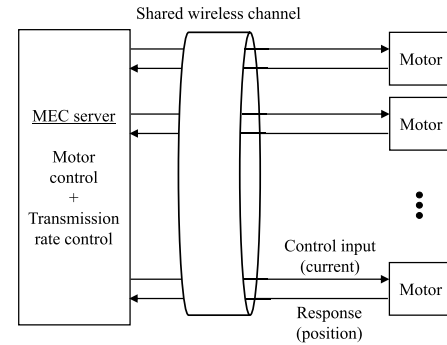


Fig. 12 A networked motor control system using a shared wireless channel and transmission rate control.

#### 4.4 Bandwidth-Guaranteed Transmission Rate Control

As a co-design approach, which is a network control technique taking control performance into account, we propose a transmission rate control for an NCS using a shared wireless channel. The networked motor control system shown in Fig. 12 will be used as an illustrative case. The networked motor control can be directly applied to motion control systems such as teleoperation systems, robots, drones, and automobiles. The system is composed of an MEC server, shared wireless channel, and distributed motors. The MEC server has motor control and transmission-rate control functions.

The centralized feedback controller implemented in the MEC server controls the positions of the distributed motors. Each motor has a position sensor that sends position response signals to the controller. Predetermined position command signals are input to the controller, and the controller calculates the control input signals based on a proportional-integral-derivative (PID)-based control algorithm.

In a conventional proration-based transmission rate control, the upstream and downstream bandwidths of the shared wireless channel are equally allocated to each distributed motor by the transmission rate controller in the MEC server. The bandwidths allocated to each motor are decreased excessively, which means that the transmission rates decrease as well, when a large number of motors are simultaneously connected to the MEC server. Therefore, no motors cannot be controlled even if most of connected motors do not operate. For example, suppose that in a motor control, the transmission rates of the control input and response signals are both 1000 packets/s, which means that the control period is 1 ms. Suppose as well that the capacity of the shared wireless channel for the upstream or downstream direction is 1000 packets/s, and 100 motors are simultaneously connected to the MEC server. If proration-based transmission rate control is applied to the system, signals can be exchanged at a rate of at most 10 packets/s. This low rate may degrade control performance and destabilize the control system.

Here, we propose a bandwidth-guaranteed transmission rate control for an NCS using a shared wireless channel. We assume the NCS shown in Fig. 12. As shown in Fig. 13,



each motor sends a request message including its operation time and requested transmission rate to the transmission rate

controller in the MEC server before starting its operation. The transmission rate controller adjusts the operation schedule of the motors so that the number of operating motors does not exceed an upper limit. The scheduling is on a first-come-first-served basis. The upper limit is calculated using the transmission rate requested by each motor. After the scheduling, the transmission rate controller sends a grant message including start/stop time and upstream/downstream transmission rate. At the designated start time, the centralized feedback controller in the MEC server starts sending the control input signals to the motors, and the motors start sending the response signals to the centralized feedback controller, with the transmission rate allowed by the transmission rate controller. When the operation ends, each motor sends the end message to the transmission rate controller. Since the proposed bandwidth-guaranteed transmission rate control reserves the operation time and transmission rate of the shared wireless channel for each motor, control performance is maintained at a constant level.

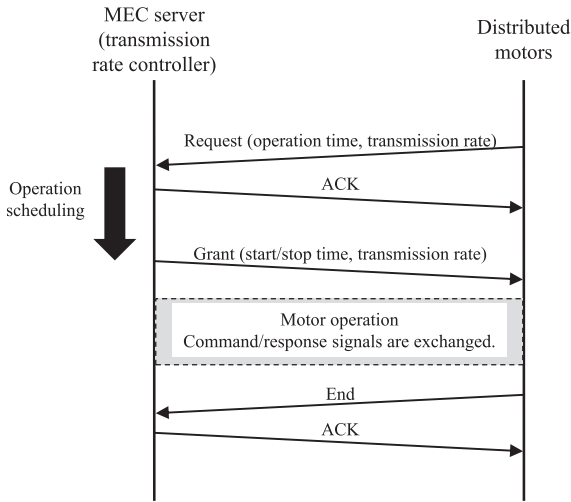
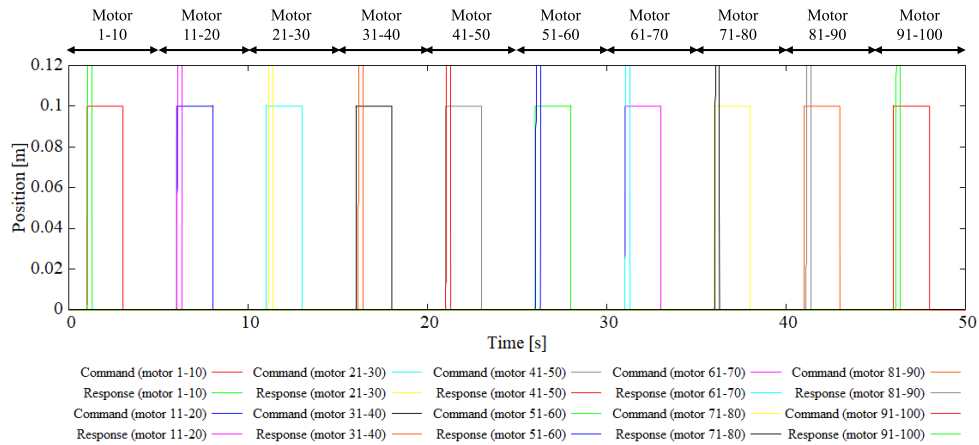
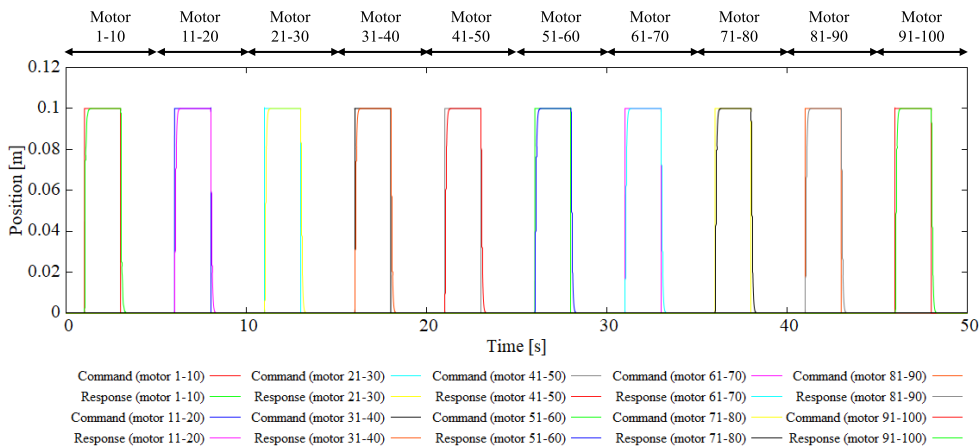


Fig. 13 Bandwidth-guaranteed transmission rate control.



(a) Proration-based transmission rate control



(b) Bandwidth-guaranteed transmission rate control

Fig. 14 Simulation results.

#### 4.5 Simulation

We performed simulations to determine the effectiveness of the proposed technique. In particular, the simulations compared the proration-based transmission rate control and bandwidth-guaranteed transmission rate control. The position control of distributed motors shown in Fig. 12 was assumed. The channel capacity was set to 1000 packets/s, and 100 motors were connected to the MEC server. Ten motors, which formed a motor group, were operated simultaneously. Motor groups 1-10, 11-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80, 81-90, and 91-100 were operated in sequence. The position command of each motor was set to 0.1 m in operation. Each operation time was set to 2 s. When the motor was not operated the position command was set to 0. A proportional-derivative (PD) controller was implemented as a centralized feedback controller, and a disturbance observer instead of an integral controller was implemented to achieve robust motion control. The proportional and derivative gains were set to 900 and 60, respectively. The cut-off frequency of the disturbance observer was set to 100 rad/s. The mass and thrust constant of each motor were set to 0.5 kg and 32.5 N/A, respectively. The requested transmission rate of each motor was set to 100 packets/s in the proposed bandwidth-guaranteed transmission rate control. The simulations assumed no packet delays or packet losses in the shared wireless channel.

The simulation results are shown in Fig. 14. The results of motors belonging to the same group are illustrated on the same line, since their operations were identical. Ideally, each position response signal tracks corresponding position command signal. In the conventional proration-based transmission rate control, the transmission rates of the control input and response signals were both 10 packets/s, as 100 motors were always connected to the MEC server. In the results shown in Fig. 14(a), no response signals of the motors converged to the corresponding command signals. On the other hand, in the proposed bandwidth-guaranteed transmission rate control, the transmission rates of the control input and response signals were both 100 packets/s, because 10 motors were operated simultaneously. The operation information was sent to the MEC server in advance, and the transmission rate required to maintain control performance at a constant level was guaranteed. The results in Fig. 14(b) show that the response signals of the motors converged to the corresponding command signals.

In practice, packet delays and losses in the shared wireless channel, which were not considered in this paper, affect the performance and stability of the NCS. However, various methods for compensating the packet delays and losses have been proposed [16]–[18]. They can be directly applied to the NCS to improve the performance and stability. This paper focused on a network-side approach, while the unavoidable delays and losses generated in the system can be compensated by using the control-side approach.

#### 5. Conclusion

We described novel mobile network architectures and context-aware controls that will make possible the efficient, highly reliable, and real-time mobile networks of the IoT era. We invite readers to study the presented architectures and research context-aware control methods in which applications and wireless networks cooperate.

#### Acknowledgments

This paper includes some of the results of “the research and development project for efficient and optimized use of radio resource in mobile networks with a variety of IoT devices” commissioned by the Ministry of Internal Affairs and Communications, Japan.

#### References

- [1] 5GMF, “5GMF White Paper ‘5G Mobile Communications Systems for 2020 and beyond’ Version 1.1,” Sept. 29, 2017.
- [2] Association of Radio Industries and Businesses (ARIB), “Mobile Communications Systems for 2020 and beyond,” Sept. 2014.
- [3] NEC Corporation, “Network Evolution toward 2020 and Beyond,” 2015.
- [4] T. Shimojo, Y. Takano, A. Khan, and S. Kaptchouang, “Future mobile core network for efficient service operation,” *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pp.1–6, April 2015.
- [5] H. Arimitsu, K. Matsumoto et al., “Current Status of Development of Network Virtualization Technologies,” *NTT Technical Review*, vol.12 no.8 Aug. 2014.
- [6] 3GPP TS 23.401, “GPRS enhancements for E-UTRAN access (Release 13),” 2015.
- [7] ETSI MEC ISG, “Mobile-Edge Computing – Introductory Technical White Paper,” Dec. 2014.
- [8] NTT DOCOMO, INC., “DOCOMO 5G White Paper 5G Radio Access: Requirements, Concept and Technologies,” July 2014.
- [9] 3GPP TR 23.708, “Architecture enhancements for service capability exposure,” June 2015.
- [10] 3GPP TS 23.682, “Architecture enhancements to facilitate communications with packet data networks and applications (Release 13),” Sept. 2015.
- [11] T. Iwai, A. Kohiga, and N. Itoh, “Direction of mobile network architecture and context-aware network control technology towards IoT era,” *Technical Report, CQ 2017*, Jan. 2017.
- [12] N. Itoh, H. Kaneko, T. Iwai, A. Kohiga, and H. Shimonishi, “Novel packet scheduling for supporting various real-time IoT applications in LTE networks,” *Proc. IEEE CQR 2017*, May 2017.
- [13] D. Kominami, T. Iwai, H. Shimonishi, and M. Murata, “A control method for autonomous mobility management systems toward 5G mobile networks,” *Proc. IEEE international conference on communication workshop*, May 2017.
- [14] A. Kashiwagi and I. Urabe, “Adaptive response of a gene network to environmental changes by fitness-induced attractor selection,” *PLoS One*, vol.1, no.1, pp.e49:1–10, Dec. 2006.
- [15] K. Leibnitz, N. Wakamiya, and M. Murata, “A bio-inspired robust routing protocol for mobile ad hoc networks,” *Proc. IEEE international conference on computer communication and networks*, pp.321–326, Aug. 2007.
- [16] J. Baillieul and P.J. Antsaklis, “Control and communication challenges in networked real-time systems,” *Proc. IEEE*, vol.95, no.1, pp.9–28, Jan. 2007.
- [17] R.A. Gupta and M.-Y. Chow, “Networked control system: Overview

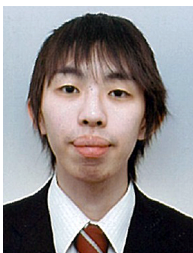
and research trends,” *IEEE Trans. Ind. Electron.*, vol.57, no.7, pp.2527–2535, July 2010.

- [18] L. Zhang, H. Gao, and O. Kaynak, “Network-induced constraints in networked control systems — A survey,” *IEEE Trans. Ind. Informat.*, vol.9, no.1, pp.403–416, Feb. 2013.
- [19] E.G.W. Peters, D.E. Quevedo, and M. Fu, “Controller and scheduler codesign for feedback control over IEEE 802.15.4 networks,” *IEEE Trans. Control Syst. Technol.*, vol.24, no.6, pp.2016–2030, Nov. 2016.
- [20] P. Park, P.D. Marco, and K.H. Johansson, “Cross-layer optimization for industrial control applications using wireless sensor and actuator mesh networks,” *IEEE Trans. Ind. Electron.*, vol.64, no.4, pp.3250–3259, April 2017.
- [21] ITU-T Technology Watch Report, “The Tactile Internet,” Aug. 2014.
- [22] Z.-W. Wang and H.-T. Sun, “Control and scheduling co-design of networked control system: Overview and directions,” *Proc. ICMLC 2012*, pp.816–824, July 2012.
- [23] F.-L. Lian, J. Moyné, and D. Tilbury, “Network design consideration for distributed control systems,” *IEEE Trans. Control Syst. Technol.*, vol.10, no.2, pp.297–307, March 2002.
- [24] P. Marti, J. Yezpez, M. Velasco, R. Villa, and J.M. Fuertes, “Managing quality-of-control in network-based control systems by controller and message scheduling co-design,” *IEEE Trans. Ind. Electron.*, vol.51, no.6, pp.1159–1167, Dec. 2004.



**Takanori Iwai** received his B.E. and M.E. degrees in electrical and electronic engineering from Shinshu University, Japan, in 2002 and 2004. In 2004, he joined NEC Corporation, Japan, and is a Principal Researcher at System Platform Research Laboratories. He is currently pursuing his Ph.D. degree in integrated design engineering at Keio University. His research interests include system control, mobile and wireless networking, and IoT service networks. He received the Electrical Science and Engineering

Award from them the Promotion Foundation for Electrical Science and Engineering in 2016, and the Best Paper Award from the IEEE ComSoc International Communications Quality and Reliability Workshop (CQR'17).



**Daichi Kominami** received his M.E. and D.E. degrees from Osaka University, Japan, in 2010 and 2013. He is currently an Assistant Professor at the Graduate School of Economics, Osaka University, Japan. His research interests include distributed control in communication networks.



**Masayuki Murata** received an M.E. and a D.E. in Information and Computer Sciences from Osaka University in 1984 and 1988. In April 1984, he joined the Tokyo Research Laboratory of IBM Japan as a researcher. From September 1987 to January 1989, he was an Assistant Professor at the Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. From 1992 to 1999, he was an Associate

Professor at the Graduate School of Engineering Science, Osaka University, and became a Professor at the same school in April 1999. He moved to the Graduate School of Information Science and Technology, Osaka University in April 2004. He has published more than 300 papers in international and domestic journals, and has given presentations at numerous conferences. His research interests include computer communication networks, as well as performance modeling and evaluation. He is a Fellow of IEICE and a Member of IEEE, the Association for Computing Machinery (ACM), The Internet Society, and IPSJ.



**Ryogo Kubo** received his B.E. degree in system design engineering and his M.E. and Ph.D. degrees in integrated design engineering from Keio University, Japan, in 2005, 2007 and 2009, respectively. In 2007, he joined the NTT Access Network Service Systems Laboratories, NTT Corporation, Japan. Since 2010, he has been with Keio University, Japan, where he is currently an Associate Professor at the Department of Electronics and Electrical Engineering.

His research interests include system control, optical communications, networking, sensor-actuator networks, smart grids and mechatronics. He received the Best Paper Award from the IEICE Communications Society in 2011, the IEEE International Conference on Communications (ICC '12) Best Paper Award in 2012, and the Leonard G. Abraham Prize from the IEEE Communications Society in 2013.



**Kozo Satoda** received his B.E. and M.E. degrees in electrical engineering from Kyoto University in 1991 and 1993 respectively. He joined NEC in 1993. He is a senior manager at System Platform Research Labs., NEC since 2016. He has received best paper award of IEEE CQR workshop 2010, best paper award of IEEE CCNC 2017, IEICE Communications Society Excellent Paper Award 2016, 63rd Electrical Science and Engineering Promotion Awards and 2016 IPSJ Industrial Achievement Award. His research

interests include multimedia communication, streaming and mobile traffic management. He is a member of IEICE and IEEE.