A Control method for autonomous mobility management systems toward 5G mobile networks

Daichi Kominami*, Takanori Iwai†, Hideyuki Shimonishi†, and Masayuki Murata‡
*Graduate School of Economics, Osaka University, Japan
Email: d-kominami@econ.osaka-u.ac.jp
†Graduate School of Information Science and Technology, Osaka University, Japan
Email: murata@ist.osaka-u.ac.jp
‡System Platform Research Labs, NEC Corporation, Japan
Email: t-iwai@hx.jp.nec.com, h-shimonishi@cd.jp.nec.com

***Abstract*—The 5th generation mobile and wireless communication system has been actively studied as a next-generation mobile communication network. In the future communication network, all people and devices will connect to the Internet via wireless networks and many challenges remain to be solved for the 5G network. Centralized control in the control plane is found to be difficult for the management of connected user equipments. Previously we proposed an architecture for autonomous and distributed mobility management and a biologically inspired mobility management scheme that adaptively selects a location of management entity. However, this scheme has some problem in a network-wide perspective due to entities' autonomous decision-making. In this paper, we introduce a control node to monitor and manage the network for carrying out the resolution of the unsolvable problem of autonomous distributed control. We show the control node can improve the stability of the network without further loss of the performance of the entire network.**

***Keywords*-5G network, attractor selection, bio-inspired algorithm, controlled self-organization**

I. Introduction

Future communication networks will connect all people and devices with each other via wireless networks, that is, so-called the Internet of things (IoT) and machine-to-machine (M2M) communication will be critical techniques for our secure, safe and affluent living. For that, the 5th generation mobile and wireless communication system has greatly attracted many researchers as a next-generation mobile communication network.

There remain a lot of technical challenges for the 5G network [1], [2]. In the IoT and M2M communication, the 5G network should allow a large number of small-size packets transmission with low latency and communication overhead to enhance network resource utilization and network performance. In the current long term evolution/evolved packet core (LTE/EPC) network, a serving gateway (SGW) handles the user plane (U-Plane) and a mobility management entity (MME) manages the control plane (C-Plane) in a centralized manner. This centralized information management of connected user equipments (UEs) has caused bottleneck problems [3].

Fully autonomous and distributed control techniques, called self-organization control techniques, can be a means of resolving these problems. The self-organization control is described in the time evolution equation based on local information for deriving a solution, which avoids the explosion of the amount of collecting information and calculating solutions. Self-organization controls intend to optimize the system by emergence of the individuals' decision-making.

We proposed a network architecture that placed MMEs as a logical function on the mobile network [4]. These MMEs autonomously perform the management of UEs in a distributed manner and we call them an autonomous distributed MME (ADMME). The role of the mobility management for a UE can be delegated from one ADMME to another ADMME (*ADMME switching*) and the selection of the new ADMME is made at the previous ADMME's discretion. We call it *ADMME selection*. Determining the best placement of ADMMEs in the mobile-core network and determining which ADMME manages which UE by taking into consideration of UEs' position are quite challenging.

In [4], ADMME selection method based on the *attractor-selection algorithm* [5] was proposed for load balancing between ADMMEs and for suppressing the communication delay between an ADMME and UEs, where each ADMME makes a decision using only local information. However, the autonomous and distributed control generally suffers from its instability when there exist multiple states that satisfy the local optimality [6]. Our proposed method in [4] cannot solve this problem and generates the excessive ADMME switching. The movement of the context and an increase in signaling accompanied with this switching lead to an increase in the load of the C-Plane. In addition, the local decision-making of ADMMEs cannot directly control the performance of the entire network. Even worse, the performance of the entire network cannot be expected till the network initiates operations.

Main contribution of this paper is to propose a novel network architecture that solves the problems in the autonomous ADMME selection method while keeping its ad-

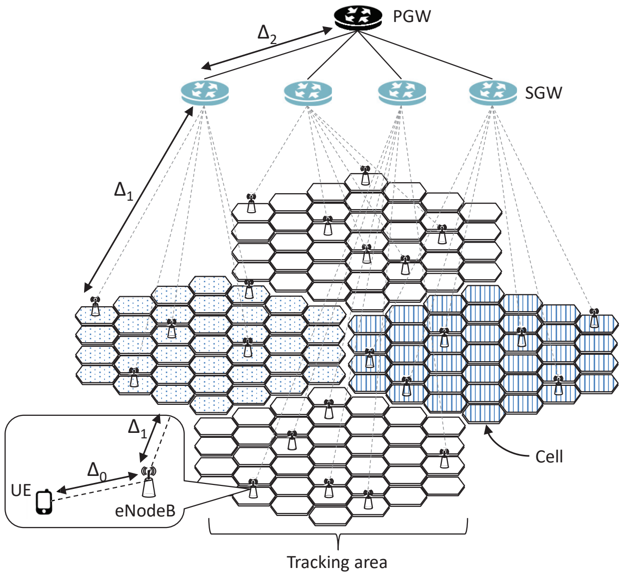


Figure 1. Network and delay model

vantages. Note that our proposal is designed to achieves the suppression of ADMME switching even when considering the mobility of UEs. In order to meet the performance requirements and to suppress the instability of the entire network, we introduce a *control node* [7] that monitors and manages the network. The purpose of the control node in this paper is to induce the network performance to a sub-optimal value and to suppress the switching of the connection between an ADMME and a UE as long as it does not involve deterioration from the desired performance. To this end, the control node has a feedback mechanism, where it periodically observes the network performance and provides a control input to each ADMME according to the observed value.

The remainder of this paper is organized as follows. First, we briefly explain attractor selection-based ADMME selection algorithm in Section II. Then, we propose a control architecture for controlling an autonomous mobility management system in Section III. The performance of our proposal is shown by computer simulation in Section IV. Finally, in Section V, we conclude our work and suggest areas for future work.

II. Autonomous ADMME Selection based on Attractor Selection

An attractor selection-based ADMME selection algorithm is proposed in [4]. At first, we briefly explain the ADMME selection problem and then, explain how the attractor selection algorithm solves this problem.

A. ADMME Selection Problem

An ADMME can run on any node of eNodeBs, SGWs, or PGWs. In this paper, we assume that all nodes have ADMME functionality as well in [4]. Each ADMME manages the mobility information of UEs. ADMME switching can be happened when a UE transmits a tracking area update (TAU) request, a handover request, or an attach request to a current ADMME. Here, a TAU request is transmitted when a TAU timer of a UE has been expired or a UE has moved to another tracking area. A handover request starts when a UE has moved to another cell. An attach request is sent when a UE joins in a mobile network.

When an ADMME receives such a request from a UE, the ADMME determines which ADMME is eligible to transfer the management role of the requesting UE to reduce the communication delay between the UE and the new ADMME and to reduce the load concentration on the new ADMME. Here, we define the candidate set for a new ADMME. An ADMME selects a new ADMME from the candidate set that consists of ADMMEs in all nodes on the path between a requesting UE and the ADMME. In addition, the candidate set includes ADMMEs in the SGW and the PGW nearest from the UE. In [4], from the candidate set, an ADMME selects a new ADMME according to *delay history* and *load status* of nodes (details are described in the following subsection). We assume that communication delay is estimated by a node that has an ADMME through the request of a UE. Regarding the load status, each ADMME collects load status of nodes in the current candidate sets periodically. After the determination of a new ADMME, a current ADMME transmits a delegation message to the new ADMME with the context information of the requesting UE. The response message for the UE's request is sent by the new ADMME.

B. Attractor Selection-based ADMME Selection

The attractor selection model mathematically explains the biological systems that adapt themselves to unexpected changes in their surroundings [5]. In this model, each component of the system periodically updates its state. Therefore, this model provides not simply a heuristic algorithm, but a remarkable adaptation to dynamically changing environments according to the current fitness to the environments. Various network control methods have been applied it [8]. We applied this algorithm to solve the ADMME selection problem for delay reduction and load balancing in [4]. Figure 1 shows our assumed network model. The delay between a UE and an eNodeB (Δ_0), between an eNodeB and an SGW (Δ_1), and between an SGW and a PGW (Δ_2) are constant value for the sake of simplicity. Load status is the number of UEs that the corresponding ADMME manages.

An ADMME has a vector $\mathbf{m} = (m_1, m_2, \cdots, m_M)$ for each UE that the ADMME manages. M is the cardinality of the candidate set for the corresponding UE. m_i is a state value that corresponds the adequacy of ADMME i to select. An ADMME also has a scalar value α for each UE, which is called *activity*. Activity expresses the goodness of the current selection of an ADMME. In the attractor selection

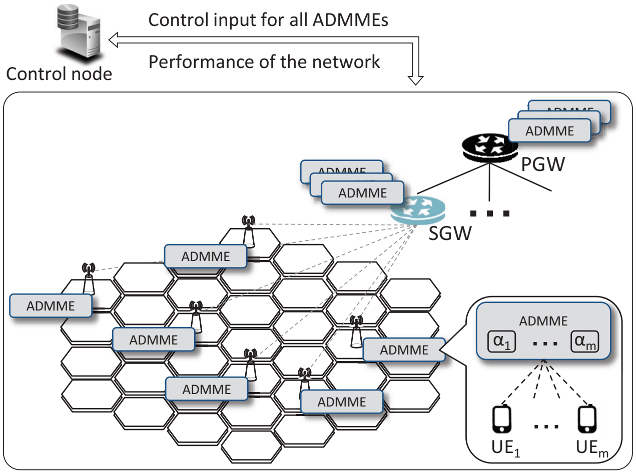


Figure 2. Control node and ADMMEs in the assumed network

algorithm, $\mathbf{m}$ is updated by the following equation, where f is the derivative of a function that has M attractors and $\boldsymbol{\eta}$ is a noise.

$$\frac{\mathrm{d}\mathbf{m}}{\mathrm{d}t} = \alpha f(\mathbf{m}) + \boldsymbol{\eta}. \tag{1}$$

When activity is high, an $\mathbf{m}$ converges to an attracted state and when activity is low, it seeks another state with a random behavior. An ADMME updates its activity when a request from a UE arrives and then, it also updates $\mathbf{m}$ using the activity. Then, an ADMME in the candidate set whose state value is the largest is selected as a new ADMME. The state vector, activity, and delay history are transmitted to the new ADMME and taken over by it.

For the h-th request, an ADMME calculates its activity α as follows:

$$\alpha(h) = \rho \cdot \alpha_d(h) + (1-\rho) \cdot \alpha_l(h), \tag{2}$$

where ρ is a parameter from 0 to 1 for determining the weight of $\alpha_d(h)$ and $\alpha_l(h)$.

In the attractor selection-based ADMME selection, the estimated delay (denoted by $\hat{d}$) is used by an ADMME. An ADMME estimates $\hat{d}_i$ at an expected communication delay from node i to a UE via the ADMME. If node i is on the shortest path between the UE and the ADMME, $\hat{d}$ is the sum of two one-way delay: from the UE to the ADMME and from the ADMME to node i. Otherwise, $\hat{d}_i = 0$. This delay information during past W steps is stored as a delay history. For this, an ADMME periodically sends probe packets to nodes in its candidate set. This is done along with collecting load status of nodes in the current candidate sets. It is noteworthy that in the candidate set, as $\hat{d}_i$ is larger, node i is closer to the corresponding UE.

$\alpha_d(h)$ is an activity based on $\hat{d}$ calculated from

$$\alpha_d(h) = \left(\frac{\sum_{k=1}^{W} \frac{\hat{d}_{cm}(h-k)}{k}}{\max_{1 \le i \le M} \sum_{k=1}^{W} \frac{\hat{d}_i(h-k)}{k}} \right)^{\epsilon}, \tag{3}$$

where $\hat{d}_{cm}$ is the delay of a current ADMME and ϵ is a parameter to determine the output level for $\alpha_d(h)$.

$\alpha_l(h)$ is an activity based on load status calculated from

$$\alpha_l(h) = \frac{\min_{1 \le i \le M} l_i(h)}{l_{cm}(h)}, \tag{4}$$

where $l_i(h)$ is the latest load status of node i for the h-th request. l_{cm} is the load of a current ADMME.

Using the calculated $\alpha(h)$ from the above $\alpha_d(h)$ and $\alpha_l(h)$, an ADMME updates $\mathbf{m}$ as follows:

$$\frac{\mathrm{d}m_i}{\mathrm{d}t} = \frac{s(\alpha(h))}{1 + m_{max}^2 - m_i^2} - \alpha(h) \cdot m_i + \eta_i, \tag{5}$$

where $m_{max} = \max_{1 \le j \le M}\{m_j\}$, $s(\alpha) = \alpha(h)[\beta \cdot \alpha(h)^{\gamma} + 1/\sqrt{2}]$, and η_i is the white Gaussian noise with mean of 0 and variance of 1.

III. Control Method for Autonomous ADMME Selection

The ADMME selection based on attractor selection has difficulty to predict the performance emergence in the entire network particularly due to the non-linearity of its dynamics. Also, the algorithm cannot deal with its instability near the local optima. Connection switching between UEs and ADMMEs results in increases in the C-Plane load and in delay caused by the handover procedures. To solve these problems, we propose a novel architecture for controlling such systems, where a new activity function is defined so that a system manager can control the stability and performance of the system.

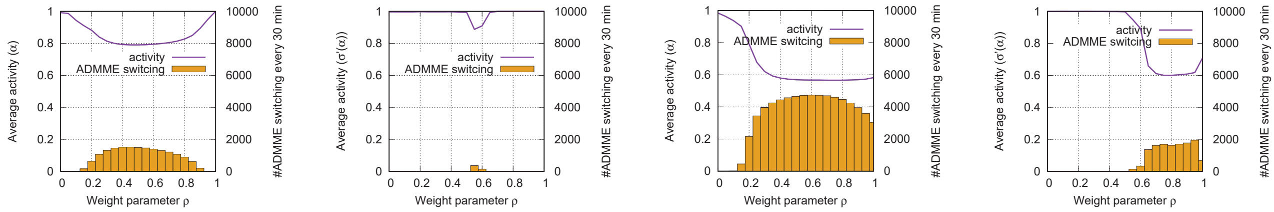
A. *New activity function*

The activity α in the attractor selection algorithm expresses the goodness of the current state of an ADMME and therefore, α determines the behavior of an ADMME. When α for a UE is larger than about 0.8, an ADMME does not quit hold of the UE. Otherwise, as α is smaller, an ADMME is more likely to delegate the mobility management of the UE to another ADMME. Since the prediction of the performance of the system is difficult, the activity α in (2) does not necessarily exceed 0.8. Therefore, we use a sigmoid function ($\sigma(\alpha)$) instead of α itself to guarantee that activity becomes more than 0.8.

$$\sigma(\alpha) = \frac{1}{1 + e^{-g(\alpha - \alpha_{th})}}, \tag{6}$$

where g is a non-negative parameter that determines the decay characteristics of the sigmoid function. The sigmoid function decays quickly with large g. α_{th} is the inflection point of the function.

Although the sigmoid function yeilds the advantage of the stability, it produces another problem. When α_i, which is an activity for UE i, is smaller than α_{th}, even if a current ADMME is a good for UE i, the ADMME may



(a) Static & autonomous (b) Static & controlled (c) Random walk & autonomous (d) Random walk & controlled

Figure 3. Activity and ADMME switching vs. weight parameter ρ

Table I
Parameters on the attractor selection algorithm

Parameter	Value
β	10
γ	10
ϵ	2
W	5
g	30
α_{th}	0.6

hand over UE i to another ADMME since $\sigma(\alpha_i)$ is nearly zero, which causes less convergence of the system. To solve this problem, $\sigma(\alpha)$ should not be too small value between $0 \leq \alpha_i \leq \alpha_{th}$. We use (7) as a new activity function and control its threshold parameter α_{th}, which satisfies the high stability and convergence speed of the system.

$$\sigma'(\alpha) = \max\bigl(\alpha, \sigma(\alpha)\bigr), \tag{7}$$

B. System control

As described in the previous section, an ADMME determines a new ADMME that are to manage the UE currently managed by the ADMME. This determination is on the basis of local information. The attractor selection model dynamically updates the state of the system to adapt to changing environments. When activities of all ADMMEs are close to one, load balancing and delay minimization in all ADMMEs are realized. However, load balancing and delay minimization among all the nodes are not compatible. If multiple objectives are incompatible, activities of ADMMEs are unlikely to become one in the attractor selection-based ADMME selection algorithm. At this time, since each ADMME operates in an autonomous distributed manner, it cannot know about what the value of the performance of the entire network is before the system operation. An attractor structure of each ADMME is determined by its activity and the performance locally experienced by it.

The difficulty of predicting the system performance indicates that we cannot determine the best value of α_{th} beforehand. When α_{th} is comparatively high, frequent ADMME switching may happen if $\sigma'(\alpha)$ cannot be larger than 0.8. On the other hand, when α_{th} is comparatively small, an ADMME may stop with a worse choice, which aggravates the system performance.

To determine the best α_{th}, a control node manages α_{th} to achieve the high stability and performance of the system as shown in Fig. 2. The control node should control the behavior of an individual ADMME and the performance of the network. In our proposal, the control node monitors the performance of the whole network and introduces control input to each ADMME so that it satisfies the performance requirement and reduces the number of switching of connection between a UE and an ADMME.

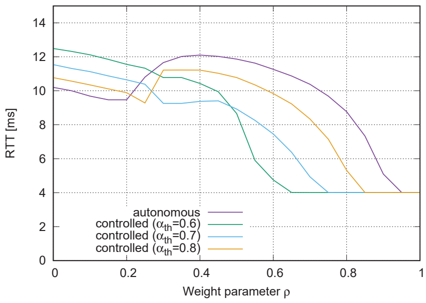
IV. Simulation Experiments

In this section, we evaluate our proposal by using computer simulation. We show the simulation results of the purely autonomous distributed system and the controlled system. Comparing these results, not only we show the advantages of the control method, but we show the disadvantage.

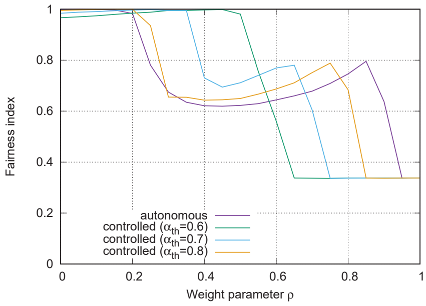
A. Settings

We assume the mobile core network consisting of one PGW and four SGWs. Each SGW corresponds to one TA and each TA is made up with 37 hexagonal cells as shown in Fig. 1. In each cell, there is one eNodeB, therefore the number of eNodeBs is 148. For the sake of simplicity, delays between nodes are static; $\Delta_0 = 2$ ms, $\Delta_1 = 20$ ms, and $\Delta_2 = 3$ ms, respectively. The number of ADMMEs in an eNodeB, an SGW, and a PGW is set to 1, 5, and 5, respectively. At the beginning of the simulation, 100 UEs are deployed in each cell and they connect to an ADMME on the nearest SGW, that is, each ADMME on a SGW has 740 UEs. In our simulation, TAU timer of every UE is set to 30 min.

We consider two situations: a static pattern and a random-walk pattern. In the static pattern, all UEs do not move to other cells. In the random-walk pattern, each UE stays in a cell for T_s and then it moves to one of neighboring cells. After the movement, it sends a handover request to a current ADMME via an eNodeB in a new cell. T_s is set to 100 min. Parameters in the attractor selection algorithm is shown in



(a) Average RTT between ADMMEs and UEs



(b) Fairness index

Figure 4. Performance in immobility scenario

Table. I. We set the same value of α_{th} for all ADMME in this evaluation.

B. Results

1) ADMME switching: At first, we show ADMME switching occurs more frequently when α gets smaller. Figure 3 presents the average activity over ADMMEs and the average number of ADMME switching every 30 min with changing ρ. The notation "autonomous" means the original ADMME selection method in [4] and "controlled" means our proposal. In the situation where UEs do not move, the average α in Fig. 3(a) does not reach one except for $\rho = 0$ or 1. Therefore, even in the case with immobile UEs, ADMME switching happens especially when ρ is about 0.5. This is because α_l and α_d do not become one at the same time for almost all of ADMMEs when ρ is not zero or one. By setting α_{th} to 0.6, the average value of $\sigma'(\alpha)$ gets larger than 0.9 at any value of ρ, which drastically reduces ADMME switching as shown in Fig. 3(b). Here, average $\sigma'(\alpha)$ is comparatively small when ρ is 0.55 and 0.60 for the same reason in Fig. 3(a).

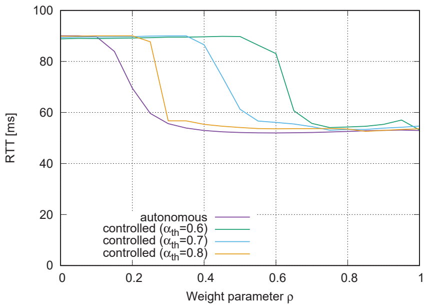
As it can be seen from Fig. 3(c), the number of ADMME switching becomes very large as the node moves. Activities of almost all nodes become one only when ρ is one because load balancing does not depend on the current position of the UEs. If ρ is other than one, the delay time between a UE and an ADMME increases as the UE moves to another cell, resulting in a decrease in its activity. When ρ is greater than 0.3, the value of the activity little changes. By using a control threshoud α_{th}, we can approach the average activity to almost one when ρ is less than or equal to 0.5 as shown in Fig. 3(d). As a result, the number of ADMME switching can be greatly reduced. On the other hand, when ρ is 0.5 or more, the average activity is slightly larger than that in Fig. 3(c), but the number of ADMME switching can be reduced to about half. The reason is that the number of ADMMEs whose activities take values close to one is increasing. Thus, by setting an adequate value to α_{th}, we can reduce the number of ADMME switching. In the following section, we examine the effect of α_{th} on the network performance.

2) Network performance: Figure 4 shows an average round-trip delay time (simply denoted by RTT) and fairness index with changing the parameter ρ. As shown in the figure, in the original method, performance irregularly varies depending on the value of ρ. The point to be emphasized is that ρ, which is a weight parameter between delay reduction and load balancing, cannot fulfill its role. Briefly, this implicates the difficulties to predict the performance of self-organizing systems.

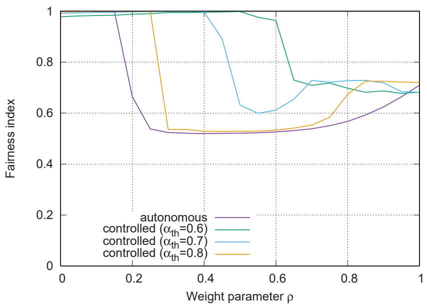
Using a control node, RTT and fairness index can be adjusted by a network manager. The lower the value of α_{th}, the greater the influence of ρ, which makes it possible to give difficult weight to the performance. In the figure, the results when α_{th} is set to 0.6, 0.7, and 0.8 are shown. Increasing α_{th} means to get closer to "autonomous." Also, if α_{th} is small, the system will tend to fall into local optima during the search of the best state, so we need to set the appropriate α_{th}. Fortunately, α_{th} can be controlled by a network manager through the control node in our proposal. The network manager can confirm the information on performance and on the number of ADMME switching, and then gives feedback to α_{th}.

With respect to the network performance, when activity is higher than about 0.8, $\mathbf{m}$ converges and ADMME switching is suppressed. However, as mentioned above, this leads the system to fall into local optima and therefore, the network performance may become worse than those in the original.

Figure 4 shows the network performance without consideration of node movement. In Fig. 4(a), When ρ is 0.2 or less, RTT is increased by adding control. On the other hand, when ρ is larger than 0.3, the control method reduces RTT. In both cases, it can be said that the effect of the weight parameter ρ emerges powerfully from the control method. In the figure, as α_{th} gets smaller, RTT approaches to about 4 ms with smaller ρ. Here, 4 ms of RTT means that all UEs belong to the ADMMEs in eNodeBs. Focusing on fairness, by the control method, when ρ is 0.8 or more, the fairness index of the control method is lower than that of the original method as shown in Fig. 4(b). However, by setting a lower value to α_{th}, fairness index gets about one within a broader range of ρ. From the above results, although the original



(a) Average RTT between ADMMEs and UEs



(b) Fairness index

Figure 5. Performance in random-walk scenario

method fails to adjust the balance between delay reduction and load distribution by ρ, our proposal can improve this point particularly when α_{th} is 0.6.

Figure 5 describes the network performance in consideration of node movement. The difference between the RTT of "controlled" and "autonomous" is widening as compared with the case where node movement is not considered. When the node moves with a sojourn time of 100 min, the result of "autonomous" shows the shortest value in any case as shown in Fig. 5(a). By the control method, although RTT becomes longer, control the effect of ρ can be achieved by α_{th}. Here, RTT of any results approach to about 52 ms, which means that most UEs connect to SGWs' or PGW's ADMMEs. Contrary to the result of RTT, the fairness index of the control method outperforms that of the original almost every ρ.

It is unknown what results will be obtained in advance even when using α_{th}, but it is important to bring in a control method that can manage them. Since the appropriate threshold value changes according to the moving speed of UEs, it is important to deal with the environment in which various moving patterns coexist. One solution for this is to set an appropriately value of α_{th} for individual ADMME according to the frequency of node movement, which is our future work.

V. Conclusion and Future Work

In this paper, we propose a control method for controlling the network stability and performance in an autonomous system for the 5th generation mobile and wireless communication system. We introduce a control node into the self-organized ADMME selection algorithm. Through the computer simulation, at first, we show that although our proposed method is very simple, the introduction of a control method provides very useful impact on terms of system stabilization and performance control. This result can be referred as a reduction in the overhead of the control plane in the 5G network. We also show that the original attractor selection-based ADMME selection method cannot predict its performance due to its nonlinear dynamics. This indicates that control methods are indispensable for actual use of such autonomous and distributed methods. Our proposed algorithm can be utilized in various distributed systems. One of the applications, we will focus on for future work is the mobile edge computing architecture. In the mobile edge computing system, virtual computing functions are dispersed over the network and users should have to select an appropriate one for asking to process of some applications. Our mechanism can be helpful for such systems.

References

[1] A. Osseiran and F. Boccardi, "Scenarios for 5G mobile and wireless communications: the vision of the METIS project," *IEEE Communication Magazine*, vol. 52, no. 5, pp. 26–35, Feb. 2014.

[2] "Updated scenarios, requirements and KPIs for 5G mobile and wireless system with recommendations for future investigations," May 2015. [Online]. Available: https://www.metis2020.com/wp-content/uploads/deliverables/METIS_D1.5_v1.pdf

[3] R. 7333, "Requirements for distributed mobility management," Oct. 2014.

[4] H. Yang, N. Wakamiya, M. Murata, T. Iwai, and S. Yamano, "An autonomous and distributed mobility management scheme in mobile core networks," in *Proceedings of 9th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT)*, Dec. 2015.

[5] 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.

[6] Q. Wu, G. Ding, Y. Xu, S. Feng, Z. Du, J. Wang, and K. Long, "Cognitive internet of things: a new paradigm beyond connection," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 129–143, 2014.

[7] D. Clément and B. Cyrille, "Toward organic computing approach for cybernetic responsive environment," *arXiv*, vol. 1601.01614, Jan. 2016.

[8] K. Leibnitz and M. Murata, "Attractor selection and perturbation for robust networks in fluctuating environments," *IEEE Network*, vol. 24, no. 3, pp. 14–18, May 2010.