An Autonomous and Distributed Mobility Management Scheme in Mobile Core Networks

Hua Yang, Naoki Wakamiya
and Masayuki Murata
Graduated School of Information Science
and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
{h-yang, wakamiya, murata}@ist.osaka-u.ac.jp

Takanori Iwai
and Satoru Yamano
Cloud System Research Labs, NEC Corporation
1753 Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8666, Japan
{t-iwai@hx, yamano@cb}.jp.nec.com

ABSTRACT

The 5th generation mobile and wireless communication systems are expected to accommodate exploding traffic, increasing number of devices, and heterogeneous applications driven by proliferation of IoT and M2M technologies. The centralized mobility management architecture in a current mobile core network cannot satisfy these emerging requirements. In this paper, we introduce novel architecture of distributed mobility management and an autonomous and adaptive mobility management scheme which distributes mobility management function on nodes in a mobile core network in accordance with mobility characteristics of UEs and a management policy. We adopt a biologically-inspired adaptation algorithm, called attractor selection, to accomplish adaptive selection taking into account multiple objectives. Through simulation experiments, we confirmed that our proposal could accomplish lower delay, higher load balancing, and lower C-plane overhead comparing to other methods including the current standard.

Keywords

mobile core network, distributed mobility management, C-plane overhead, attractor selection

1. INTRODUCTION

In recent years, with proliferation of IoT (Internet of Things) and M2M (Machine to Machine) technologies, the number of M2M terminal devices such as sensors and actuators are increasing exponentially [8], which further leads to the huge growth of mobile data traffic in a mobile communication network. It is foreseen that the 5th generation mobile communication systems will experience challenges of exploding mobile data traffic, considerable number of devices, and heterogeneous applications [7][5].

The current 3.9G LTE/EPC (Long Term Evolution / Evolved Packet Core) networks adopt the centralized architecture. An SGW (Serving Gateway) handles the U-Plane, i.e. user traffic, and an MME (Mobility Management Entity) supports the most relevant mobility management functions in the C-Plane of connected UEs (User Equipments). As a corollary of centralized control, it suffers from congestion of not only user data but also control traffic. More specifically, an MME becomes easily overloaded by providing full mobility management functions to each of considerable number of M2M devices intermittently generating very short messages. In addition, physical distances between those management nodes in an EPC network and UEs results in excessive bandwidth consumption and introduces large response delay in both of U-plane and C-plane.

To tackle the problem, a direction toward distributed network architecture has drawn a lot of attentions in industry, academia, and government, being led by for example METIS (Mobile and wireless communications Enablers for Twenty-twenty Information Society) of Europe [12][13]. Recently DMM (Distributed Mobility Management) solution is a hot topic which adopts flat mobile network architecture. DMM shortens the distance from gateways to UEs by distributing mobility anchors leading to distribution of U-plane traffic [6][9]. In addition, a software distributed architecture of DMME (Distributed Mobility Management Entity) was proposed to implement distributed mobility management in the C-plane [3]. However, servers specifically dedicated to distribution of C-plane tasks need to be pre-allocated and load balancing among servers is not considered. For highly flexible management of a mobile core network, network virtualiation technologies such as SDN and NFV are considered to be incorporated [4]. Virtualization enables a high freedom of choice in topology and functional layout.

In this paper, we first propose conceptual architecture of autonomous and adaptive distribution of mobility management tasks among nodes, i.e. a PGW (Packet data network GateWay), SGWs, and eNBs (evolved NodeB). That is, our proposed architecture is compromised with the current centralized architecture of the 3.9G LTE/EPS mobile core network. For distribution of mobility management tasks, we consider ADMME (Autonomous

BICT 2015, December 03-05, New York City, United States

DOI 10.4108/eai.3-12-2015.2262427

Distributed Mobility Management Entity), a virtual node or a virtual machine which has the same functionality of MME. In our proposed architecture, ADMMEs can be deployed at any node in a mobile core network. They can be dynamically generated and removed. They can communicate with each other by using for example the S10 interface.

In addition, we propose a scheme to dynamically and adaptively select an ADMME appropriate for a UE based on its mobility characteristics and a management policy, i.e. delay mitigation or load balancing, of a mobile core network. In our proposal, each ADMME receiving a C-plane request from a UE determines whether to delegate mobility management of the UE by using information about delay, load status of nodes, and C-plane overhead of ADMME relocations. As delay between a UE and a node reflects their distance, the response time and the bandwidth consumption in the C-Plane can be reduced by appointing an ADMME closer to a UE as a serving ADMME. However, greedy delay minimization to select an eNB of a cell where a mobile UE resides causes considerable C-plane overhead by frequent ADMME relocation. Therefore we take into account the mobility characteristics of UEs by using a history record of delay information. In addition, load status of nodes indicates the degree of load concentration and it is used for load balancing among nodes. For adaptive selection of an ADMME under dynamically changing constraints, we adopt a biologically-inspired heuristics, called the attractor selection model [10]. It is a mathematical model of behaviour of biological systems that can adapt themselves to dynamically changing environment without well-designed adaptation rules. Through simulation experiments using two mobility scenarios, we show the superiority of our proposal to five other methods.

In the rest of this paper, firstly, in Section 2 we propose our ADMME selection scheme including conceptual architecture, selection mechanism, and algorithm. Then we show results of simulations and evaluate our proposal through comparisons with other methods in Section 3. Finally, we conclude this paper and describe future work in Section 4.

2. ADMME SELECTION SCHEME

In this section, we first introduce our distributed mobility management architecture. Next we describe an outline of our ADMME selection scheme and then give details of our proposed algorithm.

2.1 Distributed Mobility Management Architecture

Figure 1 illustrates our conceptual architecture of distributed mobility management. In the figure, each of a PGW, SGWs, and eNBs has one ADMME, but there could be nodes with multiple ADMMEs or no ADMME. It also is possible that some nodes are unable to serve ADMMEs due to resource limitation. Each ADMME is responsible for mobility management of connected UEs. For example, ADMME3 on an SGW maintains context information of UE1. The maximum number of UEs per ADMME depends on the capacity of a host node and available bandwidth.

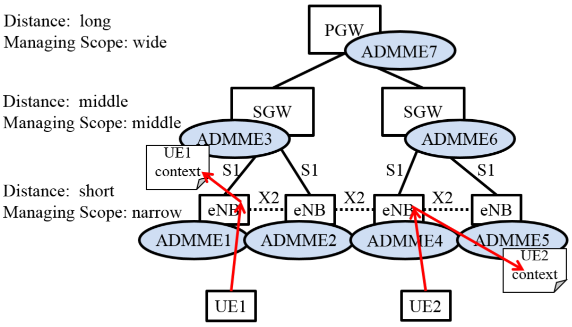

Figure 1: Conceptual architecture of distributed mobility management

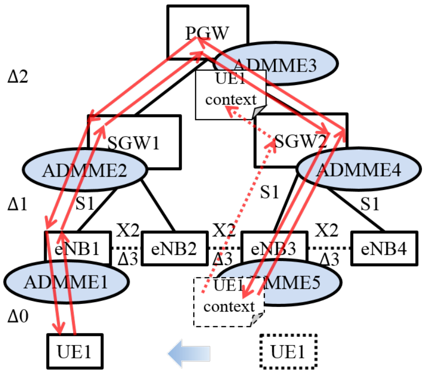

Figure 2: Example of UE movement and ADMME relocation

Since a UE must be able to communicate with an ADMME located on an arbitrary node by using standardized protocols, here we describe one of implementations for an eNB to handle a request from a UE. According to the 3GPP specifications, eNBs periodically acquire the mapping table for GUMMEIs (Globally Unique MME Identity) and IP address from operation servers in EPC like HSS (Home Subscriber Server). When an eNB receives a request from a UE, it translates a GUMMEI which uniquely identifies an ADMME, to the corresponding IP address and then communicates with the ADMME by S1AP (S1 Application Protocol) messages [1][2]. Among interconnected eNBs an X2 logical interface is also available and messages are exchanged by using X2AP (X2 Application Protocol). Further details of signaling procedures and mechanisms to realize our autonomous and distributed mobility management are out of scope of this paper and will be presented in the near future.

2.2 Outline of ADMME Selection Scheme

When a UE performs either of attach, handover, or TAU (Tracking Area Update) procedure, it sends a request to a designated serving ADMME, which we call the current ADMME. On receiving a request, the current ADMME selects an appropriate ADMME from a set of ADMMEs called possible ADMMEs by using an algorithm explained in the next section. The algorithm uses three information, i.e. the history of ERDs (Estimated Response Delays), the load status of ADMMEs, and the C-plane overhead related to ADMME relocation. An ERD of an ADMME is an

Table 1: An example of the history of ERDs

Times	ERD of possible ADMMEs				
h	eNB1	SGW1	PGW	SGW2	eNB3
	d_1(h)	d_2(h)	d_3(h)	d_4(h)	$d_5(h)$
h-1	eNB3	SGW2	PGW	Null	Null
	d_1(h-1)	d_2(h-1)	d_3(h-1)	Null	Null
h-2	eNB3	SGW2	PGW	Null	Null
	d_1(h-2)	d_2(h-2)	d_3(h-2)	Null	Null
⋮	⋮	⋮	⋮	⋮	⋮
h-W	eNB3	SGW2	PGW	Null	Null
	d_1(h-W)	d_2(h-W)	d_3(h-W)	Null	Null

estimation of the sum of the duration from emission of a request message to its reception at the current ADMME and the delay from the current ADMME to the ADMME. While the history of ERDs is a list of tuples of an ADMME identifier and the corresponding ERD derived from the past W procedures. Possible ADMMEs include those located at nodes on a path from the requesting UE to the current ADMME, those recorded in the ERD history, and those on the nearest SGW and PGW.

See Fig. 2 as an example. In Fig. 2, we denote delays between a UE and the nearest eNB, between an eNB and a connected SGW, between an SGW and a PGW as Δ_0, Δ_1, and Δ_2, respectively. Furthermore, the delay between a pair of connected eNBs is denoted as Δ_3. As UE1 first stays in a cell of eNB3 and several procedures are triggerred in this cell, ADMME5 located at eNB3 is selected to be the current ADMME for UE1. Assume that UE1 moves to the eNB1's cell and its current ADMME is still at eNB3, i.e. ADMME5 at time t_0. Then UE1 sends the h-th request the h-th request is sent to ADMME5. Since there is no X2 interface between eNB1 and eNB3, a request follows the tree structure through eNB1, SGW1, PGW, and SGW2. A request remembers a timestamp when it leaves a node. Now, timestamps that a request remembers on arriving at ADMME5 are; ADMME1=t_0+Δ_0, ADMME2=t_0+Δ_0+Δ_1, ADMME3=t_0+Δ_0+Δ_1+Δ_2, ADMME4=t_0+Δ_0+Δ_1+2Δ_2, and ADMME5=$t_0 + \Delta_0 + 2\Delta_1 + 2\Delta_2$. Then, ERDs of them are; ADMME5=$\Delta_0 + 2\Delta_1 + 2\Delta_2$, ADMME4=$\Delta_0 + 3\Delta_1 + 2\Delta_2$, ADMME3=$\Delta_0 + 3\Delta_1 + 3\Delta_2$, ADMME2=$\Delta_0 + 3\Delta_1 + 4\Delta_2$, and ADMME1=$\Delta_0 + 4\Delta_1 + 4\Delta_2$ as shown in Fig. 3. Therefore, an ADMME closer to a UE than the current ADMME has a larger ERD. Those ADMMEs located on the five nodes and their ERDs are also recorded in the ERD history of the h-th procedure as illustrated in Table. 1. Note that all the nodes in the Table. 1 are possible ADMMEs.

The second information is the load status of ADMMEs. Load balancing is one of crucial issues to mitigate influence of a node failure and avoid excessive expenditure of node and network resources especially when we consider overhead in managing a large number of M2M devices. For ADMME selection aiming at load balancing, each ADMME collects load status information from possible ADMMEs. The load status information can be for example the number of UEs that it manages, the amount of C-plane traffic, or the usage of limited node resources. In the evaluation section, we use the ratio of the number of UEs to the capacity of an ADMME.

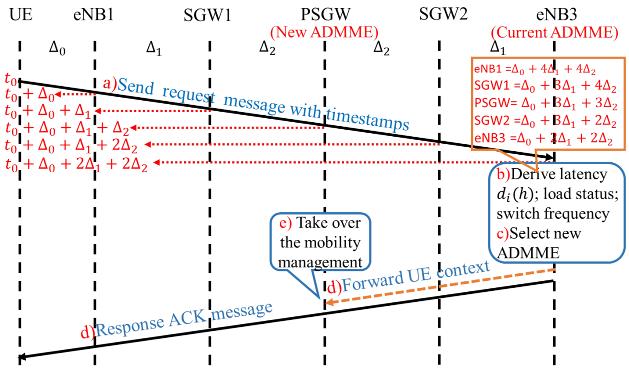
UE
eNB1
SGW1
PSGW
(New ADMME)
SGW2
eNB3
(Current ADMME)
Δ_0
Δ_1
Δ_2
Δ_2
Δ_1
t_0
$t_0 + \Delta_0$
$t_0 + \Delta_0 + \Delta_1$
$t_0 + \Delta_0 + \Delta_1 + \Delta_2$
$t_0 + \Delta_0 + \Delta_1 + 2\Delta_2$
$t_0 + \Delta_0 + 2\Delta_1 + 2\Delta_2$
a)Send request message with timestamps
eNB1 =$\Delta_0 + 4\Delta_1 + 4\Delta_2$
SGW1 =$\Delta_0 + 3\Delta_1 + 4\Delta_2$
PSGW= $\Delta_0 + 3\Delta_1 + 3\Delta_2$
SGW2 =$\Delta_0 + 3\Delta_1 + 2\Delta_2$
eNB3 =$\Delta_0 + 2\Delta_1 + 2\Delta_2$
b)Derive latency $d_i(h)$; load status; switch frequency
c)Select new ADMME
e) Take over the mobility management
d)Forward UE context
d)Response ACK message

Figure 3: the h-th TAU/Handover procedure

The last information that an ADMME uses to select an appropriate ADMME is the overhead to move UE contexts from the current ADMME to a new ADMME in ADMME relocation. When the current ADMME considers that another ADMME is more suitable to perform mobility management of a requesting UE, it delegates the mobility management task of the requesting UE by sending a Forward Relocation Request message carrying UE context information required for mobility management. Since such relocation of a serving ADMME takes time and consumes bandwidth, frequent relocation should be avoided from a viewpoint of C-plane overhead.

After making a selection, the current ADMME sends a response to a requesting UE and a Forward Relocation Request to a new ADMME at the same time. In Fig. 2, the current ADMME5 chooses ADMME3 on the PGW as a new ADMME. In this case, the response carries the updated context information of the UE with an identifier of ADMME3. Then, UE1 can recognize a new ADMME and sends succeeding requests directly to ADMME3.

2.3 Autonomous and Adaptive ADMME Selection Algorithm

Our selection algorithm adopts a nonlinear mathematical model called the attractor selection model. It is a heuristics inspired from biological systems, which can adapt themselves to dynamically changing and even unknown surroundings [10]. For its adaptability and robustness, it has been applied to a variety of network control such as routing and clustering [11].

In the general form, the attractor selection model is expressed as $d\vec{x}/dt = f(\vec{x}) \cdot \alpha + \vec{\eta}$. $\vec{x}$ corresponds to the state of a system, whose dynamics is governed by an energy function f. $f(\vec{x})$ defines attractors, i.e. a set of states where a system converges to. $\vec{\eta}$ corresponds to internal and / or external noise causing fluctuation. α $(0 \leq \alpha \leq 1)$ is a scalar value called activity. Activity α reflects the goodness of state

$\vec{x}$ in regard to the current condition. When α is large, that is, the state is appropriate for the condition, temporal dynamics of the system state is governed by the energy function f. As a result, the state converges to a nearby attractor and the system stably stays there. When the condition changes and the state becomes inappropriate, the activity decreases first. Then, the noise term dominates dynamics and the state randomly changes, i.e. random walk. Once the state approaches an attractor appropriate for the new condition, the activity gradually increases. Consequently the influence of f becomes larger and the state will be entrained to the new attractor. As a result of the increased activity, finally the state reaches the new attractor and the system becomes stable again.

In summary, the attractor selection model is heuristics combining deterministic dynamics corresponding to reinforcement of a solution and random search with mediation of the activity as feedback. The attractor selection model enables a system to find a state appropriate for the dynamically changing surrounding condition in an adaptive manner.

An ADMME maintains a scalar α $(0 \leq \alpha \leq 1)$ called activity for each UE it manages. The activity is an index of the goodness of the current ADMME serving the UE. A large activity means that the current ADMME is appropriate. In addition, an ADMME also maintains a vector $\vec{m}$=(m_1, m_2, $\cdots$, m_M) called a state vector for each UE it manages. m_i is a state value of possible ADMME i and M is the number of possible ADMMEs. A state value indicates the goodness of a possible ADMME as a serving ADMME. As explained in the previous section, a set of possible ADMME differs among UEs. In general, there are more possible ADMMEs, i.e. a larger M, for a UE with higher mobility. The current ADMME chooses a possible ADMME with the largest state value as a new ADMME.

When an ADMME receives the h-th request from a UE, it calculates the activity by using the following equation.

$$\alpha(h) = \rho \cdot \alpha_{delay}(h) + (1-\rho) \cdot \alpha_{load}(h), \tag{1}$$

where ρ $(0 \leq \rho \leq 1)$ is a weight parameter to take a balance of $\alpha_{delay}(h)$ and $\alpha_{load}(h)$ in accordance with a management policy of a mobile core network.

$\alpha_{delay}(h)$ is a delay-based activity which is derived as,

$$\alpha_{delay}(h) = \left(\frac{\sum_{k=0}^{W-1} \frac{d_{cm}(h-k)}{k+1}}{\max\limits_{1 \leq i \leq M} \sum_{k=0}^{W-1} \frac{d_i(h-k)}{k+1}} \right)^{\varepsilon}, \text{if } h > W. \tag{2}$$

or

$$\alpha_{delay}(h) = \left(\frac{\sum_{k=0}^{h-1} \frac{d_{cm}(h-k)}{k+1}}{\max\limits_{1 \leq i \leq M} \sum_{k=0}^{h-1} \frac{d_i(h-k)}{k+1}} \right)^{\varepsilon}, \text{if } h \leq W. \tag{3}$$

$d_i(h)$ is the delay of possible ADMME i measured by the h-th request message and cm means the current ADMME. Therefore, $\alpha_{delay}(h)$ is the ratio of the weighted sum of delays of the current ADMME to the maximum weighted sum of delays of possible ADMMEs for the past W requests.

A load-based activity $\alpha_{load(h)}$ is derived from the load status of ADMMEs as,

$$\alpha_{load}(h) = \frac{\min\limits_{1 \leq i \leq M} l_i(h)}{l_{cm}(h)}. \tag{4}$$

Here, $l_i(h)$ is the ratio of the number of UEs that possible ADMME i manages to the maximum number which is determined by taking into account computational capacity, memory, and bandwidth that ADMME i can use at a node. When the load of the current ADMME is higher than any of possible ADMMEs, $\alpha_{load}(h)$ becomes small.

Finally, we have one more parameter $\delta(h)$ to take into account C-plane overhead. It is derived from the number of UE context migrations, i.e. ADMME relocations, as,

$$\delta(h) = \frac{\max(N(h), N_{SGW}(h)) + 1}{N_{SGW}(h) + 1}. \tag{5}$$

$N(h)$ is the number of UE context migrations in the past W procedures regarding the requesting UE. $N_{SGW}(h)$ is the estimated number of UE context migrations in the current standard architecture. In the current architecture, MMEs are located at SGWs. When a UE moves from one TA to another, the corresponding context information is moved between MMEs serving those TAs. Therefore, $N_{SGW}(h)$ is identical to the number of UE movements between different TAs. When $N(h)$ is larger than $N_{SGW}(h)$, it is better not to change a serving ADMME to suppress the C-plane overhead.

Then $\delta(h)$ is combined with $\alpha(h)$ as,

$$\alpha(h) \leftarrow \begin{cases} \alpha(h) \cdot \delta(h), & \text{if } \alpha(h) \cdot \delta(h) < 1, \\ 1, & \text{if } \alpha(h) \cdot \delta(h) \ \geq 1 \end{cases} \tag{6}$$

As a result, ADMME relocation is avoided even with $\alpha(h)$ derived by Eq. 1 is very small when the number of UE context migrations is larger than the current standard, i.e. $\delta(h)$ is large.

After activity α is derived, an ADMME updates state vector $\vec{m}$ by using the following equation.

$$\frac{dm_i}{dt} = \frac{s(\alpha(h))}{1 + m_{max}^2 - m_i^2} - d(\alpha(h)) \cdot m_i + \eta_i. \tag{7}$$

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

If the current ADMME is appropriate for a UE, α becomes high and dynamics of $\vec{m}$ is governed by the first two terms of the right-hand side of Eq. 7. It pushes the largest state value, corresponding to the current ADMME, to increase while making the other state values decrease. As a consequence of reinforcement, the system will reach a stable state where one state value out of M is the largest and the others are small. State values are stably kept with small perturbation of a noise term. On the contrary, if the current ADMME has a large delay or unfair load status, α becomes small and the dynamic system is not stable any more. By being driven by the noise term, state values randomly change and the role of mobility management of the requesting UE would be delegated to another ADMME. If an ADMME leading to smaller delay or fairer load status is selected,

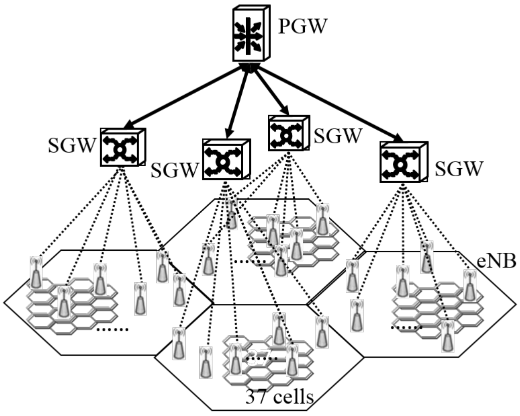

Figure 4: Simulation topology

the activity eventually increases and selection becomes stable. Therefore, the attractor selection-based heuristics is a combination of random search and reinforcement of a good solution. Furthermore by using the activity as a feedback, it can adapt to dynamically changing conditions.

However, when the number of UE context migrations is large, the random search does not take place by the effect of $\delta(h)$. When $N(h) > N_{SGW}(h)$, $\delta(h)$ becomes larger than one. Then, activity $\alpha(h)$ in Eq. 6 becomes larger than that in Eq. 1, which disturbs random search and maintains the current selection. Therefore, introduction of $\delta(h)$ spoils adaptive selection of appropriate ADMME to some extent, but it contributes to suppression of frequent and sensitive relocation of ADMME. In the next section, we evaluate the effectiveness of $\delta(h)$ by comparing to two other schemes without $\delta(h)$, called Simple and Deterministic.

3. EVALUATION AND DISCUSSION

In this section we evaluate our proposal by comparing with five other methods from viewpoints of delay, load, and C-plane overhead using two mobility scenarios.

3.1 Simulation Setting

A mobile core network used for simulation experiments has one PGW, four SGWs, and 37 eNBs per SGW as illustrated in Fig. 4. Each eNB covers a hexagonal cell of diameter Φ in which 100 UEs are located at the beginning of a simulation run. Therefore there are 14800 UEs. Cells of 37 eNBs connected with an SGW organizes a large hexagonal TA. We consider torus topology and each TA shares borders with all the other TAs. As for delays we set $\Delta_0 = 2$ ms, $\Delta_1 = 20$ ms, $\Delta_2 = 3$ ms, and $\Delta_3 = 4$ ms depending on their average physical distances. To demonstrate load balancing performance, we empirically set the capacity of ADMMEs at a PGW, an SGW, and an eNB as 8000, 4000, and 200 UEs per ADMME, respectively, which are set according to the process capacity of servers in our mobile network model. The initial location of an ADMME for a UE is the nearest eNB in our proposal, but it can be any other place in reality. Parameters used are $\beta = 10$, $\gamma = 10$, and $W = 5$ which are determined based on preliminary experiments. We change ρ as 0, 0.5, and 1 to investigate the influence of a weight parameter.

All UEs are attached throughout a simulation run, but only 30% out of them randomly selected at every minute are connected and the remaining 70% are in the idle state. They move from one cell to another based on a stay timer. The stay time interval of a UE is set at random following the Gaussian distribution whose average is T_s and variance is 1 at the beginning of a simulation run. The initial value of stay timer of a UE is set at random from 0 to its stay time. When a stay timer expires a UE moves to randomly selected one of neighbor cells. We consider two mobility scenarios with different T_s setting. In Scenario 1, all UEs use the same T_s=0.5, 1, 2, or 10 hr. We also consider immobile UEs. In Scenario 2, 50% of UEs use $T_s = 0.5$ hr and the other half use $T_s = 10$ hr. A TAU timer for periodic TAU is identically 30 min, but an initial value is set at random for each UE.

We compare six methods, i.e. Proposal, Simple, Deterministic, PPGW, PSGW, and PeNB. Proposal is our proposal. Simple is our proposal but without $\delta(h)$. Deterministic does not use $\delta(h)$ as well and deterministically stops relocation of ADMME when $N(h) \geq N_{SGW}(h)$ in the past W. Thus, Simple and Deterministic are used to evaluate the effectiveness of $\delta(h)$ in reduction of C-plane overhead. PPGW represents a case without ADMME relocation, in which ADMMEs for all the UEs are persistently located at PGW. PSGW is the existing method corresponding to the current 3GPP/LTE standard, where an ADMME is persistently located at an SGW nearest to a UE. PeNB shows an extreme case where an ADMME is persistently located at an eNB nearest to a UE, which should lead to delay minimization.

For comparison, we consider three measures. The first is the average response delay, which is the average duration from emission of a request from a UE to reception of a response. The second is the fairness of load. We use the Jain's fairness index as,

$$f(t) = \frac{(\sum_{i \in \{PGW,SGW,eNB\}} \bar{l}_i)^2}{3 \cdot \sum_{i \in \{PGW,SGW,eNB\}} \bar{l}_i^2}, \tag{8}$$

where $\bar{l}_i$ is the average load of nodes of type $i \in \{PGW, SGW, eNB\}$ at the end of a simulation run. The third is the number of UE context migrations per time step per UE, here, time step is 10 minutes in our simulations. For Scenario 1, we also evaluate by the average C-plane packets overhead, which is the average number of messages per UE per hour for C-plane mobility management, including the messages both for C-plane mobility management between UE and current ADMME and for ADMME relocations. In the following we show average values over 100 simulation runs.

3.2 Results and Discussion

First Figs. 5, 6, and 7 show results of Scenario 1 with $\rho = 1$. Therefore, our algorithm only consider delay in Eq. 1. X-axes shows the mobility of UEs corresponding to the cell diameter Φ (for regular setting Φ=10 km) divided by the average stay time T_s. 0 means immobile. Obviously, PeNB has smallest response delay between UE and current MME node, however, its average delay increases greatly in proportion to the UE mobility. The great increase of average delay in PeNB is caused by the increase of UE context

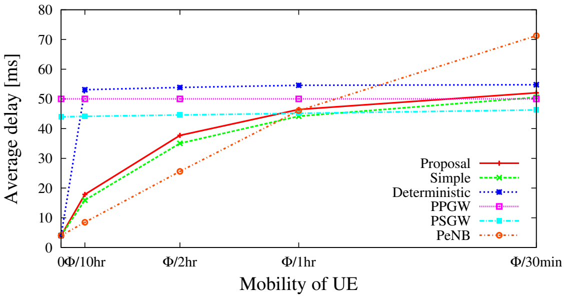

Figure 5: Average delay in Scenario 1 ($\rho = 1$)

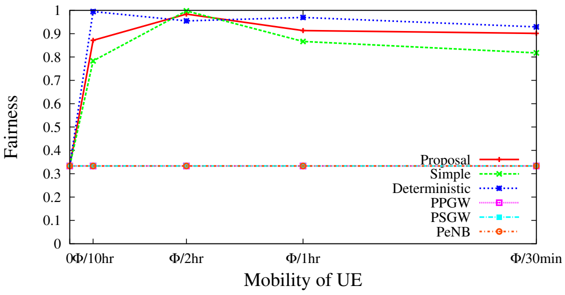

Figure 6: Fairness in Scenario 1 ($\rho = 1$)

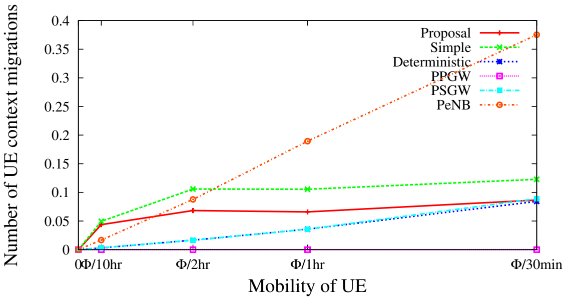

Figure 7: Number of UE context migrations in Scenario 1 ($\rho = 1$)

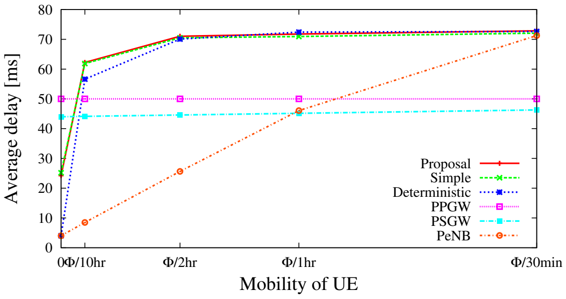

Figure 8: Average delay in Scenario 1 ($\rho = 0$)

migrations. From Fig. 7 we can know that UE context migrations of all the methods increase by UE mobility. For the same reason, PSGW also has a small increase in ADMME relocations and a slight increase in average delay. Increase of Average delays in Proposal and Simple are mainly caused by the random search to find an adaptive ADMME and the distance from UE to current ADMME. For most UE mobility cases their average delays are smaller than PPGW, which is a constant value, i.e. 50 ms.

When a UE moves within one TA, our algorithm is more likely to select an ADMME on the corresponding SGW. Furthermore, when a UE frequently moves from one TA to another, an ADMME on the PGW would be selected. A reason why the nearest eNB is not preferred as a location of a serving ADMME in our algorithm is that our proposal is not greedy. Although the nearest eNB soon becomes a distant eNB resulting in large delay for highly mobile UEs, it takes time for a state value of the nearest ADMME to become larger than that of the current and further ADMME. For immobile or less mobile UEs, a node closer or nearest to a UE is eventually selected and the delay becomes smaller than PSGW as shown in Fig. 5. As for Deterministic, because of strict restriction on ADMME relocations, the average delay cannot be reduced enough except the immobile case, in which there are no ADMME relocations. On the contrary, Proposal can stochastically change a serving ADMME even with a large $\delta(h)$ for $N(h) \geq N_{SGW}(h)$, which results in as small delay as Simple. However, the number of UE context migrations in Proposal is reduced a lot comparing to Simple. Interestingly, even with $\rho = 1$, the fairness of our proposed methods is very high as shown in Fig. 6. This is because that in the random search phase, an ADMME on a PGW is selected most and then those on SGWs, because they become a possible ADMME more often than ADMMEs on eNBs.

Next Figs. 8, 9, and 10 show results of Scenario 1 with $\rho = 0$, where our algorithms only consider load balancing in Eq. 1. It is apparent that Proposal, Simple, and Deterministic result in larger delay than PPGW, PSGW and PeNB except the immobile case. It is because our proposal is likely to select ADMMEs on a PGW and SGWs, i.e. distant nodes, having larger capacity than eNBs for the sake of load balancing. A reason why average delay of immobile UEs is smaller than PSGW and larger than PeNB is that an ADMME of each UE is located at either of the nearest SGW or the nearest eNB. On the contrary, because of mobility, an ADMME is not necessarily located at a nearest node in the other cases. As a result of sacrifice of delay, load is fully balanced among nodes in Proposal and Simple as shown in Fig. 9. A reason why Deterministic cannot achieve high fairness for immobile UEs is that there is no ADMME relocations. PPGW, PSGW, and PeNB have low fairness as they locate their MME on only one type of nodes in mobile core network. As shown in Fig. 10, PeNB which has the minimum delay between UEs and current ADMMEs suffers from the largest C-plane overhead. On the contrary, in Proposal, Simple, Deterministic, there is no or quite few ADMME relocations after convergence, since the fairness is already satisfied.

Then Figs. 11, 12, and 13 show results of Scenario 1 with $\rho = 0.5$ where both of delay and load are considered in our algorithm. By comparing with cases of $\rho = 1$ and $\rho = 0.5$, we can find that the average delay and the number of UE context migrations per UE are similar to those with $\rho = 1$

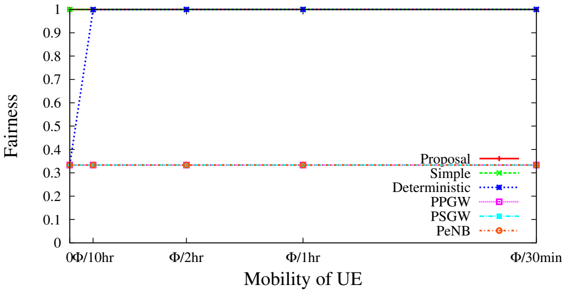

Figure 9: Fairness in Scenario 1 ($\rho = 0$)

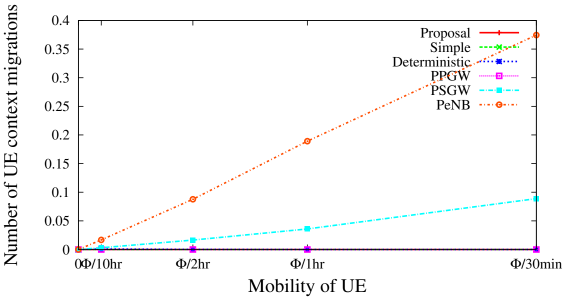

Figure 10: Number of UE context migrations in Scenario 1 ($\rho = 0$)

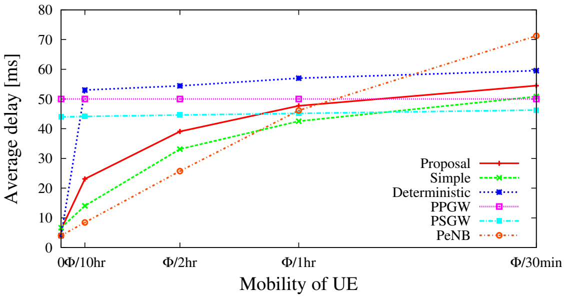

Figure 11: Average delay in Scenario 1 ($\rho = 0.5$)

while the fairness is close to that with $\rho = 0$. Regarding fairness, Proposal and Simple achieves as high fairness as with $\rho = 0$ except for low mobility cases, where average delay is more dominant in ADMME selection. Apparently Deterministic inferiors to the others from viewpoint of delay, while it has the smallest C-plane overhead. The gap between Proposal and Simple in Fig. 11 becomes larger than in Fig. 5, but Simple leads to much higher C-plane overhead than Proposal as shown in Fig. 13.

For Scenario 2, we set $\rho = 0.5$. Results are summarized in Figs. 14. There are five or three sets of bars in the figures. For comparison purposes, we show results of Scenario 1 (S1) in the leftmost and rightmost positions, respectively. A set of bars at the center corresponds averaged values over all UEs in Scenario 2 (S2), while two sets besides are only for UEs of T_s=0.5 and T_s=10 in Scenario 2 (S2), respectively. Since it is not possible to derive the fairness index for each of stay timer settings, the second graph in Fig. 14 has only three sets of bars.

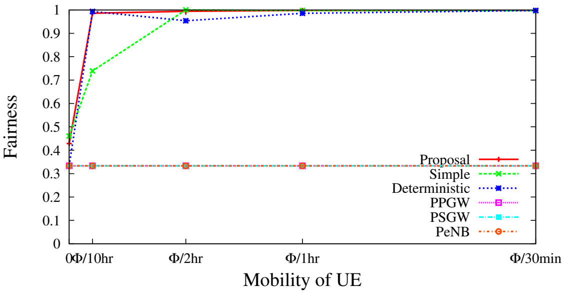

Figure 12: Fairness in Scenario 1 ($\rho = 0.5$)

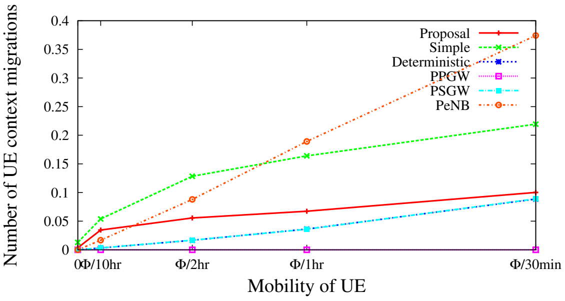

Figure 13: Number of UE context migrations in Scenario 1 ($\rho = 0.5$)

Relative relationships between methods are similar to each other in all sets of bars. We can see that Deterministic, PPGW and PSGW have relatively large delay in all cases while C-plane overhead is small. On the contrary, the average delay with Proposal, Simple and PeNB depends on the mobility of UEs as shown in Fig. 14, especially for PeNB which has a great increase. In Scenario 2, ADMMEs on a PGW or SGWs are selected for highly mobile nodes and those on SGWs and eNBs are selected for sedentary UEs. It means that our algorithm can select ADMMEs appropriate for mobility characteristics of UEs. As shown in the third graph of Fig. 14, high mobility increases the number of UE context migrations to react to UE movement in all methods except PPGW. However, Proposal can mitigate frequent ADMME relocations owing to $\delta(h)$. In addition, Proposal and Deterministic accomplishes fair allocation of tasks among nodes independently of mobility scenarios as shown in the second part of Fig. 14.

Therefore, we can conclude that Proposal can accomplish autonomous and adaptive ADMME selections taking balance between delay, load balancing, and C-plane overhead in a mobile core network.

4. CONCLUSION

In this paper we propose conceptual architecture of distributed mobility management and an autonomous and adaptive ADMME selection method. Through simulation experiments using two mobility scenarios, we confirmed that our proposal could accomplish good trade-off between delay mitigation, load balancing, and reduction of C-plane overhead. As future work we consider other scenarios such as dynamic mobility in more complex and practical simulation

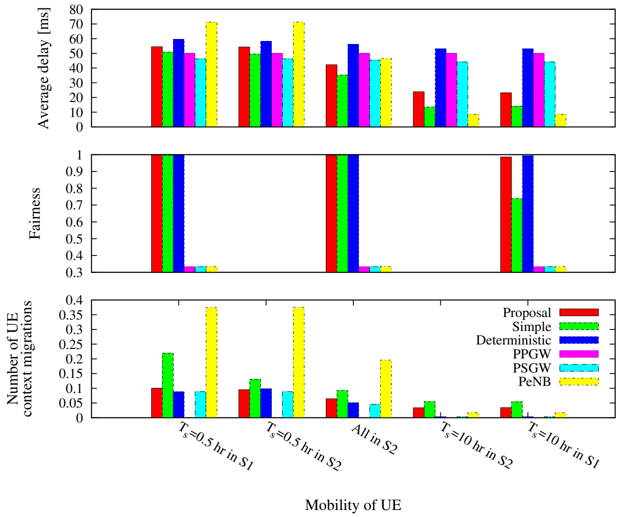

Figure 14: Results in Scenario 2 v.s. Scenario 1 ($\rho = 0.5$)

environments. We expect that our adaptive algorithm is effective in those complicated situations. Furthermore, we plan to consider mechanisms to enhance a mobile core network to accommodate M2M devices and traffic from all aspects of the U, C, and M-planes.

5. REFERENCES

[1] 3GPP. 3rd generation partnership project; technical specification group core network and terminals; numbering, addressing and identification (release 9), June 2010.

[2] 3GPP. 3rd generation partnership project; technical specification group radio access network; evolved universal terrestrial radio access network (E-UTRAN); architecture description (release 10). Sept. 2011.

[3] X. An and F. Pianese. DMME: A distributed lte mobility management entity. *Bell Labs Technical Journal*, 17(2):97–120, Feb. 2012.

[4] A. Basta, W. Kellerer, and et. al. Applying NFV and SDN to LTE mobile core gateways, the functions placement problem. In *Proceedings ofthe 4th workshop on All things cellular: operations, applications, & challenges*, pages 33–38, Chicago, USA, Aug. 2014.

[5] F. Boccardi, R. Heath, A. Lozano, T. Marzetta, and P. Popovski. Five disruptive technology directions for 5G. *IEEE Communications Magazine*, 52(2):74–80, May 2014.

[6] H. Chan, D. Liu, P. Seite, H. Yokota, and J. Korhonen. RFC 7333: Requirements for distributed mobility management. Aug. 2014.

[7] W. H. Chin, Z. Fan, and R. Haines. Emerging technologies and research challenges for 5G wireless networks. *IEEE Wireless Communications*, 21(2):106–112, 2014.

[8] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2014-2019. White Paper, Feb. 2015.

[9] F. Giust, A. D. la Oliva, and C. Bernardos. Mobility management in next generation mobile networks. In *Proceedings of IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3, Madrid, Spain, June 2013.

[10] A. Kashiwagi and I. Urabe. Adaptive response of a gene network to environmental changes by fitness-induced attractor selection. *PLoS ONE*, 1(1):e49:1–10, Dec. 2006.

[11] K. Leibnitz and M. Murata. Attractor selection and perturbation for robust networks in fluctuating environments. *IEEE Network*, 24(3):14–18, May 2010.

[12] A. Osseiran, F. Boccardi, V. Braun, and K. K. et. al. Scenarios for the 5G mobile and wireless communications: the vision of the METIS project. *IEEE Communications Magazine*, 52(5):26–35, May 2014.

[13] Y. Park. 5G vision and requirements of 5G forum, korea. Technical report, ITU, Feb. 2014.