A Context-Aware Content Delivery Framework for QoS in Mobile Cloud

Rim Haw, Md. Golam Rabiul Alam and Choong Seon Hong*
Department of Computer Engineering
Kyung Hee University
Yong In, Korea
{rhaw, robi, cshong}@khu.ac.kr

Abstract—**According to increasing performance of mobile devices, like smart phone, tablet PC and etc, and diffusing network infrastructures, like LTE, WiFi and etc, various types of content delivery services based on PC services can serve into mobile devices using cloud. In this paper we proposed content delivery framework with SDN (Software Defined Networking) and CCN (Content Centric Networking) to improve content delivery QoS in mobile cloud environment. Additionally to serve autonomic optimal services, we proposed reinforcement learning based context-aware content delivery scheme. Using our framework, we can guarantee QoS to provide context-aware content delivery scheme.**

Keywords-Mobile Cloud, Software Defined Networking, Content Centric Network, Content Delivery, Machine Learning, Context-Aware

I. INTRODUCTION

Recently, according to increasing performance of mobile devices like smart-phone, tablet PC and etc., diffusing network infrastructures like LTE, WiFi, 3G and etc. and improving performance of technology like cloud computing, users can use various services such as video/music streaming services, other content download services and etc. via mobile devices. Also, user requirement of new services which are used on PC is increasing as improving performance of mobile devices. To satisfy user requirements, mobile cloud [1][2] is appeared.

Mobile cloud can support computing environment which can overcome limited performance of mobile devices and provide services such as storing data, sharing data, streaming services, providing performance to use computation, applications, soft wares and etc. via mobile cloud, as services which can be used on private PC can be used on mobile devices, various researches have been progressing steadily to optimize services from PC to mobile devices.

In this paper, we considered content delivery services in mobile cloud as increasing content related services can cause network overhead and content delivery delay in mobile cloud environment. For this reason, we propose context-aware mobile cloud framework to guarantee QoS, where we propose a content delivery framework to reduce control plane related node mobility with SDN (Software Defined Networking) [3-6] and to decrease replica contents in the network using CCN (Content Centric Networking) [7-9]. Also we propose a learning engine to support optimized content delivery with reinforcement learning algorithm. Using proposed framework, if network condition or service performance is changed, our proposed scheme can support optimal content delivery service with machine learning operation.

The remainder of this paper is organized as follows. In section II, we briefly introduce about related works and our previous work. Then we introduce about the problem statement of our previous work [10]. Section III introduces our proposed framework using reinforcement learning for providing optimal service and guaranteeing QoS. For the verification of our framework, we present the evaluation result in section IV and we explain the conclusion and our future works in section V.

II. RELATED WORKS

A. *Software Defined Networking (SDN)*

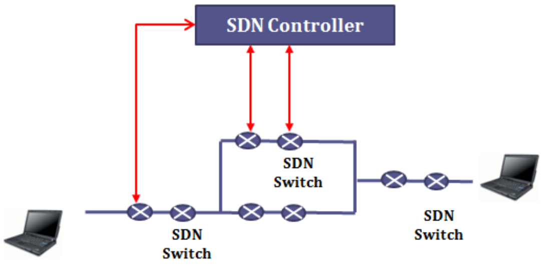


Figure 1. The Concept of SDN

The concept of SDN (Software Defined Networking) is modeling networking technology via computer technology. Using SDN we can control and manage networks using software programs. Figure 1 shows basic architecture of SDN.

In SDN, there are two important network components, SDN controller and SDN switch. Each SDN switch contains a flow table. The flow table consists of rules, action and stats. Using flow table, SDN switch can be make a decision to send packets. When SDN switch receive a packet, if there are no entries in the flow table, SDN switch transmits a packet-in-message to SDN controller. Then SDN controller checks flow table entries. If there are rules (or actions) for requested packet in the flow table, SDN controller forwards a new flow table for requested packet to SDN switch. Then SDN switch updates its flow table and sends a packet with new flow table. If SDN controller does not have flow information of requested packet in flow table, SDN controller can create new rules (or actions) for the requested packet.

B. *Content-Centric Network*

Recently to solve current problems of Internet, many researches are proposed to keep the current Internet architecture and to modify some parts improved the network performance. However, some researchers suggested and proposed some clean slate approaches to rebuild the internet infrastructure from scratch. Content Centric Networking (CCN) is one part of Future Internet researches. CCN is related with sharing data and reducing replica contents in the network. To achieve these, CCN uses interest packet for request and data packet for response.

Figure 2 is introduced a basic routing scenario for CCN. A basic routing scenario in CCN is explained in the following sequences:

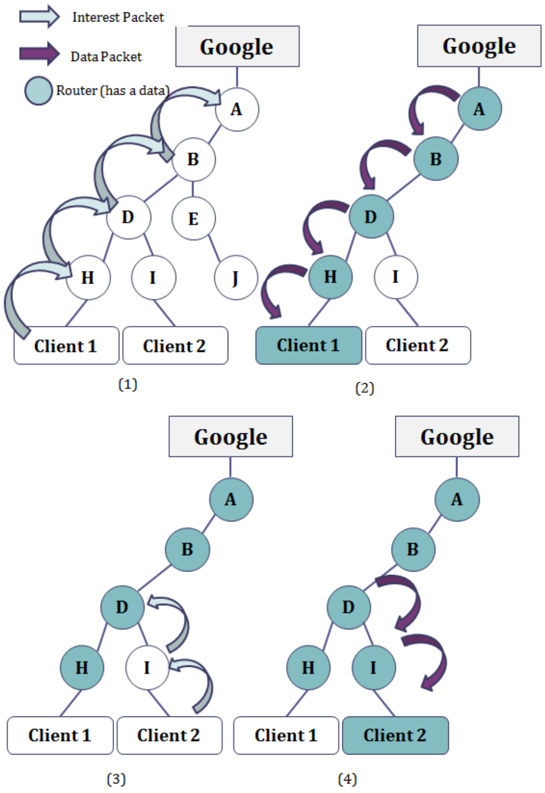


Figure 2. Basic Routing Scenario in CCN

(1) The client 1 transmits an interest packet to CCN router H. When CCN router H receives interest packet from client, CCN router H checks its content store(CS) to find whether the requested content is in the router or not. If the requested content is found within CS, CCN router H sends a data packet to client 1. However, if the content is not in CCN router's CS, CCN Router H forwards interest packets to neighbor CCN routers. According to this way, interest packet is finally arrived to the CCN Router A which has the requested content.

(2) When CCN router A receives an interest packet from CCN router B, CCN router A checks its CS. As CCN router A has requested content, CCN router A transmits data packet with reverse route to router H. In the middle of transmission, each CCN router stores the contents into CS when it receives the contents. Finally, client 1 receives the requested content from CCN router H.

(3) The client 2 requests same content. CCN router I receives an interest packet. However, as CCN router I doesn't have the requested content in its CS. CCN router I forwards interest packet to CCN router D.

(4) When CCN router D receives the interest packet, it replies a data packet including requested content to client 2.

C. *Mobile Cloud Computing*

Mobile cloud computing provides N screen cloud computing environment regardless of the types of devices and the kinds of OS. Mobile cloud can support computing environment which can overcome limited performance of mobile devices and provide services such as storing data, sharing data, streaming services, providing platforms for computing etc. Also mobile cloud has another definition i.e. a kind of cloud computing technology for DaaS (Device as a Services) to reuse devices resources.

Figure 3. Mobile Cloud

D. *The Content Dlivery Framework for QoS in Mobile Cloud*

When user requests for any content service in the mobile cloud, it will experiences a content delivery delay and the it imposes huge network overhead because of the numerous mobile users request for the contents. Among these content request a number of mobile devices request for same contents in a definite time slot which decreases network performance because of congestions and link failures. Also when mobile user moves to other place, it requires so many massage exchanges in control plane to support seamless mobility. For that reason, we previously proposed a content delivery framework for ensuring QoS in mobile cloud services [10].

Our previous work also ensures content delivery service while reducing network overheads and message overheads in control plane to support node mobility. As shown in figure 3, the proposed SDN VM to reduce control plane and content management package to support content delivery services. Also we assumed functionality of EPC [11] is included in SDN VM. It works like S/P-GW and MME. Each network entity, such as eNB, Femto Cell AP, WiFi AP, router and etc, has SDN agent to communicate with SDN VM. When new message, which is related with node mobility or content request, is arrived into SDN agent, SDN agent checks its flow table to find rule (or action) for received packet. If there are no rules (or actions) in flow table for received packet, SDN Agent sends information of received packet to SDN VM. SDN controller module in SDN VM can define received packet's flow table with agent handling function and flow management function. After defining flow table, SDN VM sends new flow table to SDN agent. If user requests content delivery service, our framework can support content delivery using content management package.

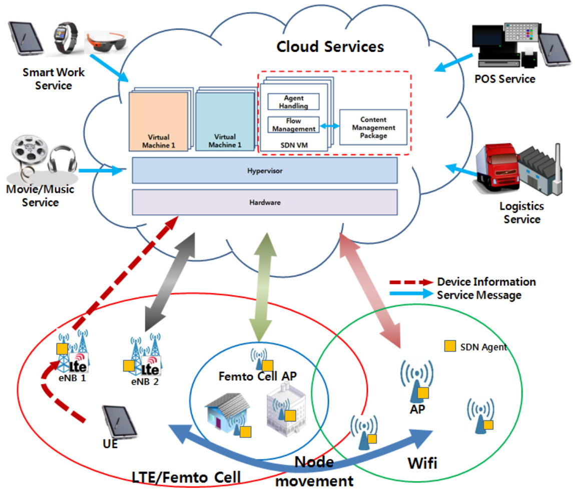


Figure 4. The content delivery framework for QoS in mobile cloud

E. Problem Statements

In [10], proposed content delivery framework has no intelligence to change the type of content delivery scheme if network status and services which users used were changed. To solve this problem, we proposed machine learning based context-aware framework.

III. Proposed Scheme

In this paper, we try to solve problem stated in section II. To improve [10]'s framework, we proposed machine learning [12][13] based context-aware content delivery framework.

A. Reinforcement Learing (RL)

Reinforcement learning (RL) is a kind of machine learning inspired by behaviorist psychology and deals with the autonomic learning from the environment by exploration and exploitation actions of software agents, and by providing rewards for intelligent actions and punishing for wrong actions of software agents. In RL, a node can learn state from trial and error by performing some behaviors. Then a node makes a decision for action related with policy. After doing some actions, a node can get reward and a node modifies its policy based on reward. In this paper we adapt RL procedure for selecting content delivery scheme.

B. The Architecture of Proposed Framework

Figure 4 shows the architecture of the proposed framework. To adapt RL technology, we proposed learning engine and service analyzer content delivery scheme. Using learning engine, proposed framework can learn which content delivery scheme has better performance. To guarantee content delivery QoS, we consider network overhead, network throughput, content delivery time, the kind of service, network performance. User's service information is gathered into service analyzer using SDN agent. The gathered information is used by learning engine. In this paper, we define two kinds of user's content delivery service. One is Non-Content Service, such as SMS, e-mail and etc., which has a better performance when using TCP/IP protocol to delivery. And the other is Content Service, such as movie streaming, big data delivery and etc. Also we assume user can select access network between LTE and WiFi.

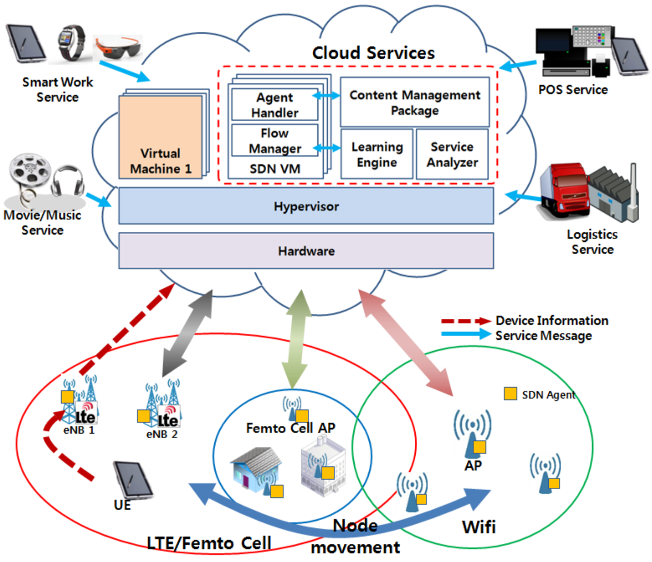


Figure 5. Proposed Context-Aware Content Delivery Framework for QoS

C. Operation of Context-Aware Content Delivery Framework

In proposed framework, learning engine can learn network environment using RL operation. To adapt the RL operation into our proposed framework, we define state, action, reward and etc. The definition of state, action, reward and punishment ar shown in equation (1). In table 1, we explain the meaning of each parameter. In equation (2), we define reward of probability of services. There are two services. One is using content services *(CS)* such as multimedia streaming services. The other is using non content services *(NCS)* such as simple SMS message which doesn't need to use CCN. Probability of using TCP and CCN's maximum value is 1. In this work, we define a reward of services where proposed framework serves non content service *(NCS)* using TCP/IP network and content service *(CS)* using CCN. Thus R_{srv} is consist of probability of *NCS* using TCP (*P(TCP|NCS*) and probability of *CS* using CCN (*P(CCN|CS*), where $R_{srv} \leq 1$.

$$
\begin{aligned}
&S = \{U_{TCP/IP}, U_{CCN}\} \\
&A = \{D_{TCP/IP}, D_{CCN}, NOP\} \\
&R = \frac{H}{H_{SLA}} + \frac{T_{SLA}}{T} + \frac{O_{th}}{O} + \frac{AN}{AN_{SLA}} + R_{srv} - P \\
&P = P_{thrpt} + P_{dl} + P_{ovhd} + P_{AN} + C_{srv}
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
P(TCP) &= P(TCP|NCS) + P(TCP|CS) \\
P(CCN) &= P(CCN|NCS) + P(CCN|CS) \\
R_{srv} &= P(TCP|NCS) + P(CCN|CS) \leq 1
\end{aligned}
\tag{2}
$$

TABLE I. PARAMETERS OF EQUATOIN 1

Parameter	Meaning
S	A set of states
A	A set of actions
R	Reward from current behavior
P	Penalty from current behavior
H	Network Throughput
T	Content Delivery Time
O	Overhead in the network
AN	Access network throughput
R_{srv}	Reward from services
C_{srv}	Penalty from services
P(TCP)	Probability of using services by TCP/IP
P(CCN)	Probability of using services by CCN
P(TCP|NCS)	Probability of using non content services
P(TCP|CS)	Probability of using content services
P(CCN|NCS)	Probability of using non content services
P(CCN|CS)	Probability of using content services

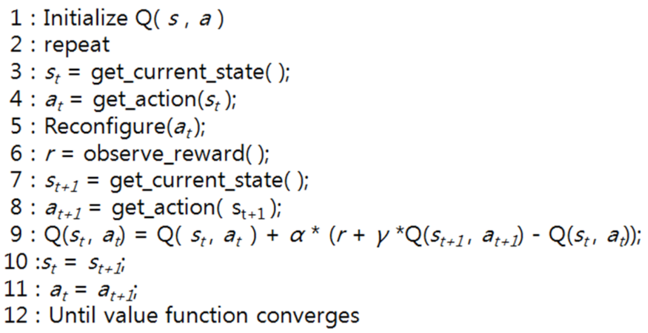
$$
C_{srv} = \begin{cases} 1 - R_{srv} & ; If\ R_{srv} \leq 1 \\ 0 & ; Otherwise \end{cases}
\tag{3}
$$

$$
\begin{aligned}
P_{thrpt} &= \begin{cases} \omega_{thrpt} * \dfrac{H_{SLA}}{H} & \text{if } H < H_{SLA}; \\ 0 & otherwise, \end{cases} \\
P_{dl} &= \begin{cases} \omega_{dl} * \dfrac{T}{T_{SLA}} & \text{if } T > T_{SLA}; \\ 0 & otherwise, \end{cases} \\
P_{ovhd} &= \begin{cases} \omega_{ovhd} * \dfrac{O}{O_{th}} & \text{if } O > O_{th}; \\ 0 & otherwise, \end{cases} \\
P_{AN} &= \begin{cases} \omega_{AN} * \dfrac{AN_{SLA}}{AN} & \text{if } AN < AN_{SLA}; \\ 0 & otherwise, \end{cases}
\end{aligned}
\tag{4}
$$

Equation (3), (4) show penalties for each reward parameter. “ω”is the weight which is used to handle the effect of SLA (or throughput) violations on the reward. If the SLA (or throughput) violation is occurred, a lot of penalties will be added into reward every time as a feedback. Therefore we set values of all penalties(ω_{thrpt} , ω_{dl} , ω_{ovhd} , ω_{AN}) are 5. If learning engine gets minus reward over some actions, learning engine didn’t select this action for next step.

$$
\begin{aligned}
&Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \\
&\quad \alpha \left[r + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]
\end{aligned}
\tag{5}
$$

To learn the optimal Q value, reinforcement learning engine operates like a pseudo code in figure 5.



```
Initialize Q( $s$ , $a$ )
repeat
$s_t$ = get_current_state( );
$a_t$ = get_action($s_t$ );
Reconfigure($a_t$);
$r$ = observe_reward( );
$s_{t+1}$ = get_current_state( );
$a_{t+1}$ = get_action( $s_{t+1}$ );
Q($s_t$, $a_t$) = Q( $s_t$, $a_t$ ) + $\alpha$ * ($r$ + $\gamma$ *Q($s_{t+1}$, $a_{t+1}$) - Q($s_t$, $a_t$));
$s_t$ = $s_{t+1}$;
$a_t$ = $a_{t+1}$;
Until value function converges
```

Figure 6. Pseudo Code of Reinforcement Learning Engine to find out optimal Q value

To find out optimal learning value Q*(s, a), RL uses temporal different methods of reinforcement learning. In temporal difference method, we can update Q(s, a) using equation (5) to find out optimal Q value in any state *s* and action *a* i.e. Q*(s, a), which we used to find out the optimal policy to change the states based on the context of content delivery network. In equation (5), α is learning rate where $0 < \alpha \leq 1$ and γ is discount rate where $0 < \gamma \leq 1$.

$$
\begin{aligned}
V(s_t) &= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots \\
&= \sum_{i=0}^{\infty} \gamma^i r_{t+i} \quad 0 \leq \gamma < 1
\end{aligned}
\tag{6}
$$

Equation (6) shows value function to find optimal Q*(s, a) for each state. While learning, value function $V(S_t)$ is also needed to be updated until value function converges. After value function converge, learning engine selects the best Q value for each state and run action. After finding optimal Q*(s, a), as learning engine maintains this situation, it will be happen another problem. One of the our future works is solving this problem using exploration mechanism.

D. State Transition Diagram

In this section, we want to describe a simple example of operation for RL. Figure 6 show State Transition Diagram for a set of state {*S0, S1*} and a set of action {*A0, A1, A2*}. *S0* stands for a state using TCP/IP and *S1* signifies a state using CCN. *A0* means changing action from using TCP/IP to using CCN. *A1* is an action which changes an action from using CCN to using TCP/IP. *A2* is an action of non operation.

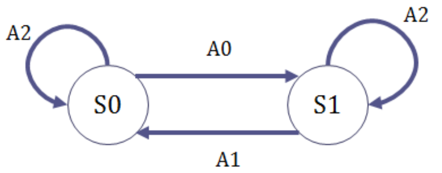


Figure 7. A State Transition Diagram for Proposed States and Actions

E. Initialization of Reward and Q value

Matrix (1), (2) stand for initialized Reward and Q value. In the reward initialization, we set “1” when state is changed, set “-1” when there are no action to moving state and set “0” when an action of non operation. For example, *S0* which stand for state using TCP/IP can move to state S1 using action *A0*. Therefore R(0,0) is “1”. However there are no selection to choose action 1 (*A1*) for state 0 (*S0*). Thus R(0,1) is “-1”. In matrix (2), we set the initial Q value is “0”. During learning operation, Q value update immediately.

$$R = \begin{array}{cc|ccc} & & \multicolumn{3}{c}{\text{Action}} \\ & & 0 & 1 & 2 \\ \hline \text{State} & 0 & 1 & -1 & 0 \\ & 1 & -1 & 1 & 0 \end{array} \qquad (1)$$

$$Q = \begin{array}{cc|ccc} & & \multicolumn{3}{c}{\text{Action}} \\ & & 0 & 1 & 2 \\ \hline \text{State} & 0 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 \end{array} \qquad (2)$$

F. Example of Q Value Update

In previous section, we already mention about a procedure of Q value update using figure 5. To help understanding Q value update easily, in this section, we describe 1st step of Q value update. A procedure Q value update is following as a sequence in figure 5. After the initialization of reward and Q value, proposed framework serve content service using TCP/IP protocol. In this time, current state is S0 and next action is *A0*. Then we can calculate the 1st step of Q value. Figure 7 shows the 1st step of Q value update.

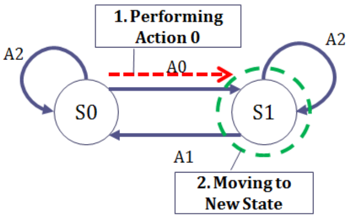


Figure 8. The Operation of the 1^{st} Step of Q Value Update

The calculated result is shown in equation (7). In this example, it is not considered reward update, as reward update can be measured in simulation environment. We set next action is changing from using TCP/IP to using CCN (*A1*). After finishing update, learning engine updates Q value immediately. Matrix (3) shows the result of updated Q value.

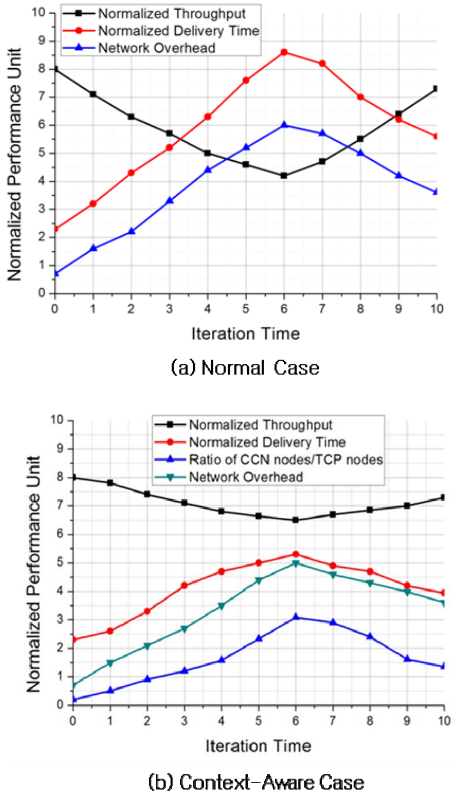
$$Q(0,0) = \underbrace{0}_{\text{Current Q Value}} + 0.3(1 + 0.8 * \underbrace{0}_{\text{Next State Q Value Q(1, 1)}} - \underbrace{0}_{}) = 0.3 \qquad (7)$$

Q Value Update

$$Q = \begin{array}{cc|ccc} & & \multicolumn{3}{c}{\text{Action}} \\ & & 0 & 1 & 2 \\ \hline \text{State} & 0 & .3 & 0 & 0 \\ & 1 & 0 & 0 & 0 \end{array} \qquad (3)$$

IV. PERFORMANCE EVALUATION

In this paper, we have introduced our proposed scheme which is reinforcement learning based context-aware content delivery framework for QoS in mobile cloud. To evaluate the performance of our framework, we have performed simulation using Java. As shown in figure 6, we have comparison between the performance of normal case without RL operation and performance of context-aware case. For the evaluation, we consider normalized network throughput, normalized content delivery time and normalized network overhead. When iteration operation is performed, context-aware case shows increasing network throughput and reducing content delivery time and network overhead against normal time.



(a) Normal Case

(b) Context-Aware Case

Figure 9. The result of comparing RL case and normal case.

V. CONCLUSION AND FUTURE WORKS

In this paper, we try to improve the content delivery QoS in mobile cloud. To achieve motivation, we proposed RL based context-aware content delivery framework. In proposed framework, we consider SDN technology to reduce control plane and CCN technology to support content delivery in mobile cloud. Also we proposed RL based learning engine to get optimal Q*(s, a) value to learn dynamically knowledge about performance of content delivery service. Via our proposed framework, even though network performance is changed in the middle of using content delivery service which uses TCP/IP protocol, framework can change content delivery scheme to CCN. To evaluate our proposed framework, we perform simulation with java. When we adapt RL based context-aware function, we can get better performance with throughput, overhead and content delivery time. However in proposed framework, it needs to adapt more parameters related network environment and services into RL operation to get more optimal Q value. Also as we find out via simulation RL machine learning is too slow to compute operations, it is need to find out algorithm to reduce calculating time. When learning engine finds out optimal Q value, our framework operates action over the best Q value in the state. This operation is make a problem when service environment is changed. As Q value update operation is stopped by the value function, framework maintains actions. Thus it is need to be update Q values using exploration mechanism such as ε-greedy exploration. Then we implement cloud test bed and perform simulation using test bed.

ACKNOWLEDGEMENT

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2012-0006421). Dr. CS Hong is the corresponding author.

REFERENCES

[1] Rakpong Kaewpuang, Dusit Niyato, Ping Wang, Member, Ekram Hossain, "A Framework for Cooperative Resource Management in Mobile Cloud Computing.",IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 31, NO. 12, pp. 2685-2700 DECEMBER 2013

[2] Md. Golam Rabiul Alam, Eung Jun Cho, Eui-Nam Huh, Choong Seon Hong, "Cloud Based Mental State Monitoring System for Suicide Risk Reconnaissance Using Wearable Bio-sensors",ACM IMCOM 2014, Jan 9-12, 2014, Siem Reap, Cambodia.

[3] OpenFlow Forum - http://archive.openflow.org/

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, Mar. 2008

[5] ONF, “Software-defined networking:the new norm for networks,”ONF(Open Network Foundation) White Paper, 2012

[6] Rim Haw, Phuong Luu Vo, Choong Seon Hong, Sungwon Lee, "An Efficient Content Delivery Framework for SDN Based LTE Network," ACM IMCOM 2014, Jan 9-12, 2014, Siem Reap, Cambodia

[7] Jacobson, Van, et al. "Networking named content." Proceedings of the 5th international conference on Emerging networking experiments and technologies. ACM, 2009.

[8] CCNX http://www.ccnx.org/

[9] Wang, Sen, Jun Bi, and Jianping Wu. "Collaborative caching based on hash-routing for information-centric networking." Proceedings of the ACM SIGCOMM, pp 535-536, 2013, New York, USA

[10] Rim Haw, Md. Goalm Rabiul Alam and Choong Seon Hong, "The Content Delivery Framework for QoS in Mobile Cloud", Smartmedia Society Spring Conference 2014, April 25th, 2014, Daejun, Korea

[11] 3GPP LTE – http://www.3gpp.org/LTE

[12] Christopher J.C.H. Watkins, Peter Dayan, "Technical Note: Q-Learning." Machine Learning, Vol 8, Isuue 3-4, pp. 279-292, May 1992.

[13] Xiangping Bu, Jia Rao, Cheng-Zhong Xu, “Coordinated Self-Configuration of Virtual Machines and Appliances Using a Model-Free Learning Approach”, IEEE Transactions on Parallel and Distributed Systems, Vol 24, pp. 681-690, April 2013.