

## Research Article

# Learning-Based QoS Control Algorithms for Next Generation Internet of Things

**Sungwook Kim**

*Department of Computer Science, Sogang University, 35 Baekbeom-ro (Sinsu-dong), Mapo-gu, Seoul 121-742, Republic of Korea*

Correspondence should be addressed to Sungwook Kim; [swkim01@sogang.ac.kr](mailto:swkim01@sogang.ac.kr)

Received 24 July 2015; Revised 2 October 2015; Accepted 20 October 2015

Academic Editor: Yassine Hadjadj-Aoul

Copyright © 2015 Sungwook Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet has become an evolving entity, growing in importance and creating new value through its expansion and added utilization. The Internet of Things (IoT) is a new concept associated with the future Internet and has recently become popular in a dynamic and global network infrastructure. However, in an IoT implementation, it is difficult to satisfy different Quality of Service (QoS) requirements and achieve rapid service composition and deployment. In this paper, we propose a new QoS control scheme for IoT systems. Based on the Markov game model, the proposed scheme can effectively allocate IoT resources while maximizing system performance. In multiagent environments, a game theory approach can provide an effective decision-making framework for resource allocation problems. To verify the results of our study, we perform a simulation and confirm that the proposed scheme can achieve considerably improved system performance compared to existing schemes.

## 1. Introduction

In the past forty years, the Internet has grown into a network that connects an estimated 1.8 billion users and has attained a global penetration rate of almost 25%. Telecommunications and the Internet are forming an increasingly integrated system for processing, storing, accessing, and distributing information and managing content. This convergence is based on the rapid evolution of digital technology and the diffusion of the concept of the Internet. In recent years, the steps of penetration of digital technologies, the evolution towards integrated telecommunications, information technology, and the electronic media sector have been actively presented. Developments in many different technologies are creating a significant, innovative, technical potential for the production, distribution, and consumption of information services [1, 2].

In 1999, Ashton first presented the concept of the Internet of Things (IoT) [2], a technological revolution that promotes a new ubiquitous connectivity, computing, and communication era. The IoT is a vision wherein the Internet extends into our everyday lives through a wireless network of uniquely identifiable objects. Therefore, the development of the IoT depends on dynamic technical innovations in a number

of fields, including wireless sensors and nanotechnology [3]. Furthermore, the IoT service infrastructure is expected to promptly evaluate the Quality of Services (QoS) and provide satisfying services by considering things such as the preferences of users' device capability and current network status. However, the definition of QoS in the IoT is not clear because it has been poorly studied. To adaptively manage an IoT system, a new QoS control model is necessary. This model must be able to balance network availability with information accuracy in delivering data [4–8].

A fundamental challenge of QoS management is that relatively scarce network resources must be selected and allocated in a prudent manner to maximize system performance [7, 8]. To adaptively allocate network resources, game theory has been widely applied in mission-critical network management problems. Typically, game theory is used to study strategic situations where players choose different actions in an attempt to maximize their payoffs, depending upon the choices of other individuals. Therefore, game theory provides a framework for modeling and analyzing various interactions between intelligent and rational game players in conflict situations [6].

In traditional game models, it is important to define equilibrium strategies as game solutions. Equilibrium strategies are assumed to be the optimal reaction to others, given full knowledge and observability of the payoffs and actions of the other players. Therefore, most equilibrium concepts require that the payoffs and strategies of the other players be known in advance and observed by all players. However, this is a strong assumption that is not the case in the majority of real-life problems. Players in actual situations have only partial knowledge, or no knowledge at all, regarding their environments and the other players evolving around them [6]. To alleviate this difficulty, van der Wal developed the Markov game model [9]. This approach relaxes the strict game model assumptions by implementing learning algorithms. Through repeated plays, Markov game players effectively consider their current payoffs and a history of observations regarding the strategies of the other players [9, 10].

The main purpose of this paper is to develop an effective QoS control scheme for IoT systems. Based on the Markov game model, we build an intelligent decision-making process that addresses the critical QoS problem of an IoT system. With a real-time learning feedback mechanism, the proposed scheme adapts well to the dynamic requirements of IoT applications. Through online-oriented strategic decisions, the proposed scheme attempts to attain a self-confirming equilibrium, the new solution concept for real-time network systems.

*1.1. Related Work.* To improve IoT system performance, several QoS control schemes have been proposed to efficiently and integrally allocate IoT resources. The Time-Controlled Resource Sharing (TCRS) scheme [11] is a scheduling scheme that shares resources between Machine-to-Machine (M2M) and Human-to-Human (H2H) communication traffic services. This scheme analytically focuses solely on resource utilization and the QoS of the M2M and H2H traffic and derives expressions for blocking probabilities of the M2M and H2H traffic and percentage resource utilization [11].

The *IoT Service Selection (IoTSS)* scheme [12] is a model to select the appropriate service, from many services, that satisfies a user's requirements. This scheme considers three core concepts, device, resource, and service, while specifying their relationships. To dynamically aggregate individual QoS ratings and select physical services, the *IoTSS* scheme designs a Physical Service Selection (PSS) method that considers a user preference and an absolute dominance relationship among the physical services.

The *Approximate Dynamic Programming based Prediction (ADPP)* scheme [13] is a novel evaluation approach employing prediction strategies to obtain accurate QoS values. Unlike the traditional QoS prediction approaches, the *ADPP* scheme is realized by incorporating an approximate dynamic programming based online parameter tuning strategy into the QoS prediction approach. The *Services-oriented QoS-aware Scheduling (SQoSS)* scheme [5] is a layered QoS scheduling scheme for service-oriented IoT. The *SQoSS* scheme explores optimal QoS-aware service composition using the knowledge of each component service. This scheme

can effectively operate the scheduling problem in heterogeneous network environments. The main goal of the *SQoSS* scheme is to optimize the scheduling performance of the IoT network while minimizing the resource costs [5].

The *Intelligent Decision-Making Service (IDMS)* scheme [4] constructs a context-oriented QoS model according to the Analytical Hierarchy Process (AHP). Using this hierarchical clustering algorithm, the *IDMS* scheme can effect intelligent decisions while fully considering the users' feedback. The earlier study has attracted significant attention and introduced unique challenges to efficiently solve the QoS control problem. Compared to these schemes [4, 5, 13], the proposed scheme attains improved performance during the IoT system operations.

The remainder of this paper is organized as follows. The proposed game model is formulated in Section 2, where we introduce a Markov decision process to solve the QoS problem and explain the proposed IoT resource allocation algorithm in detail. In Section 3, we verify the effectiveness and efficiency of the proposed scheme from simulation results. We draw conclusions in Section 4.

## 2. Proposed QoS Control Algorithms for IoT Systems

In this section, we describe the proposed algorithm in detail. The algorithm implements a game theory technique and appears to be a natural approach to the QoS control problem. Employing a Markov game process, we can effectively model the uncertainties in the current system environment. The proposed algorithm significantly improves the success rate of the IoT services.

*2.1. Markov Game Model for IoT Systems.* Network services are operated based on the Open Systems Interconnection model (OSI Model). In this study, we design the proposed scheme using a three-layered (i.e., application, network, and sensing layers) QoS architecture. At the application layer, an application is selected to establish a connection and decisions are made by the user and the QoS scheduling engine. In general, the QoS module must allocate network resources to the services that are selected in the application layer [5]. At the network layer, the QoS module must allocate network resources to the selected services. The decision-making process at this layer may involve QoS attributes that are used in traditional QoS mechanisms over networks [5]. At the sensing layer, the decision-making process involves the selection of a basic sensing infrastructure based on sensing ability and the required QoS for applications. The QoS module at the sensing layer is responsible for the selection of the basic sensing devices [5].

In this study, we investigate learning algorithms using uncertain, dynamic, and incomplete information and develop a new adaptive QoS scheduling algorithm that has an intelligent decision-making process useful in IoT systems. For the interactive decisions of the IoT system agents, we formulate a multiple decision-making process using a game model while studying a multiagent learning approach. Using this

technique, the proposed scheme can effectively improve the QoS in IoT systems.

Learning is defined as the capability of making intelligent decisions by self-adapting to the dynamics of the environment, considering experience gained in the past and present system states, and using long-term benefit estimations. This approach can be viewed as self-play, where either a single player or a population of players evolves during competitions on a repeated game. During the operation of an IoT system, learning is driven by the amount of information available from every QoS scheduler [14]. As indicated in the traditional methods, complete information significantly improves performance with respect to partial observability; however, the control overhead results in a lack of practical implementations. Consequently, a tradeoff must be made considering that the capability to make autonomous decisions is a desirable property of self-organized IoT systems [5, 14].

The Markov decision-making process is a well-established mathematical framework for solving sequential decision problems using probabilities. It models a decision-making system where an action must be taken in each state. Each action may have different probabilistic outcomes that change the system's state. The goal of the Markov decision process is to determine a policy that dictates the best action to take in each state. By adopting the learning Markov game approach, the proposed model allows distributed QoS schedulers to learn the optimal strategy, one step at a time. Within each step, the repeated game strategy is applied to ensure cooperation among the QoS schedulers. The well-known Markov decision process can be extended in a straightforward manner to create multiplayer Markov games. In a Markov game, actions are the result of the joint action selection of all players and payoffs, and state transitions depend on these joint actions. Therefore, payoffs are sensed for combinations of actions taken by different players and players learn in a product or joint action space. From the obtained data, players can adapt to changing environments, improve performance based on their experience, and make progress in understanding fundamental issues [5, 9, 10].

In the proposed QoS control algorithm, the game model is defined as a tuple  $(\mathbf{S}, N, \mathbb{A}_{k,1 \leq k \leq N}, U_{k,1 \leq k \leq N}, \mathcal{T})$ , where  $\mathbf{S}$  is the set of all possible states and  $N$  is the number of players. In the proposed model, each state is the resource allocation status in the IoT system.  $\mathbb{A}_{k,1 \leq k \leq N} = \{a_1, a_2, \dots, a_m\}$  is the collection of strategies for player  $k$ , where  $m$  is the number of possible strategies. Actions are the joint result of multiple players choosing a strategy individually. In the proposed Markov game, QoS schedulers are assumed as game players and the collection of strategies for each player is the set of availabilities of system resources.  $U_{k,1 \leq k \leq N} : \mathbf{S} \times \mathbb{A}_1 \times \mathbb{A}_2 \times \dots \times \mathbb{A}_N \rightarrow \mathfrak{R}$  is the utility function, where  $\mathfrak{R}$  represents the set of real numbers.  $\mathcal{T} : \mathbf{S} \times \mathbb{A}_1 \times \mathbb{A}_2 \times \dots \times \mathbb{A}_N \rightarrow \Delta(\mathbf{S})$  is the state transition function, where  $\Delta(\mathbf{S})$  is the set of discrete probability distributions over the set  $\mathbf{S}$ . Therefore,  $\mathcal{T}(s^t, a_1, a_2, \dots, a_N, s^{t+1})$  is the probability of arriving in state  $s^{t+1}$  when each agent takes an action  $a_i$  at state  $s^t$ , where  $s^t, s^{t+1} \in \mathbf{S}$  [5, 9, 10].

In the developed game model, players seek to choose their strategy independently and self-interestedly to maximize

their payoffs. Each strategy represents an amount of system resource and the utility function measures the outcome of this decision. Therefore, different players can receive different payoffs for the same state transition. By considering the allocated resource amount, delay, and price, the utility function ( $U$ ) of each player is defined as follows:

$$U(x) = \omega \exp\left(\left(\frac{T(x)}{\bar{T}}\right)^{0.5-\epsilon} (\tau_M - \tau)^{0.5+\epsilon}\right) - c(x, \xi), \quad (1)$$

s.t.,  $\epsilon \in \{-0.5; 0.5\}$ ,

where  $\omega$  represents the player's willingness to pay for his perceived service worth.  $\bar{T}$  is the system's average throughput and  $T(x)$  is the player's current throughput with the allocated resource  $x$ ; this is the rate of successful data delivery over a communication channel.  $\tau_M$  and  $\tau$  are the maximum delay and the observed delay of the application services, respectively.  $\tau$  is measured from real network operations. In a real-time online manner, each QoS scheduler actually measures  $\bar{T}$ ,  $T(x)$ , and  $\tau$ .  $c(x, \xi)$  is the cost function and  $\xi$  is the price for a resource unit.  $\tau$  is obtained according to the processing and arrival service rates. In a distributed self-regarding fashion, each player (i.e., QoS scheduler) is independently interested in the sole goal of maximizing his utility function as follows:

$$\max_x U(x), \quad \text{where } c(x, \xi) = \left(\frac{x\xi}{\bar{x}_A\xi}\right)^q, \quad (2)$$

where  $x$  is the allocated resource in its own QoS scheduler,  $\bar{x}_A$  is the average resource amount of all QoS schedulers, and  $q$  is a cost parameter for the cost function  $c(x, \xi)$ . The cost function is defined as the ratio of its own obtained resource to the average resource amount of all the QoS schedulers. Therefore, other players' decisions are returned to each player. This iterative feedback procedure continues under IoT system dynamics. In this study, QoS schedulers can modify their actions in an effort to maximize their  $U(x)$  in a distributed manner. This approach can significantly reduce the computational complexity and control overheads. Therefore, it is practical and suitable for real world system implementation.

**2.2. Markov Decision Process for QoS Control Problems.** In this work, we study the method that a player (i.e., QoS scheduler) in a dynamic IoT system uses to learn an uncertain network situation and arrives at a control decision by considering the online feedback mechanism. With an iterative learning process, the players' decision-making mechanism is developed as a Markov game model, which is an effective method for the players' decision mechanism. If players change their strategies, the system state may change. Based on the immediate payoff ( $U(S_0, a_i(0))$ ) of the current state  $S_0$  and action  $a_i(0)$ , players must consider the future payoffs. With the current payoff, player  $i$ 's long-term expected payoff ( $V_i(S_0, a_i(0))$ ) is given by [5]

$$V_i(S_0, a_i(0)) = \max_{a_i(t), 0 \leq t \leq \infty} \left[ U_i(S_0, a_i(0)) + \sum_{t=1}^{\infty} (\beta^t U_i(S_t, a_i(t))) \right], \quad (3)$$

s.t.,  $a_i(t) \in \mathbb{A}_i$ ,

where  $a_i(t)$  and  $U_i(S_t, a_i(t))$  are player  $i$ 's action and expected payoff at time  $t$ , respectively.  $\beta$  is a discount factor for the future state. During game operations, each combination of starting state, action choice, and next state has an associated transition probability. Based on the transition probability, (3) can be rewritten by the recursive Bellman equation form given in [5]

$$V_i(S) = \max_{a_S} \left[ U_i(S a_S) + \gamma \sum_{S' \in \mathcal{S}} \{P^i(S' | S, a_S) V_i(S')\} \right], \quad (4)$$

s.t.,  $a_S \in \mathbb{A}_i$ ,

where  $S'$  represents all possible next states of  $S$  and  $\gamma$  can be regarded as the probability that the player remains at the selected strategy.  $P^i(S' | S, a_S)$  is the state transition probability from state  $S$  to the state  $S'$ ;  $S$  and  $S'$  are elements of system state set  $\mathcal{S}$ . In this study,  $N$  is the number of QoS schedulers, and  $m$  is the number of possible strategies for each scheduler. Therefore, there are total  $m^N$  system states.

$P(S' | S, a_S)$  is a distributed multiplayer probability decision problem. Using the multiplayer-learning algorithm, each player independently learns the current IoT system situation to dynamically determine  $P(S' | S, a_S)$ . This approach can effectively control a Markov game process with unknown transition probabilities and payoffs. In the proposed algorithm, each player is assumed to be interconnected by allowing them to play in a repeated game with the same environment. Assume there is a finite set of strategies  $\mathbb{A}_{1 \leq k \leq N}(t) = \{a_1^k(t), \dots, a_m^k(t)\}$  chosen by player  $k$  at game iteration  $t$ ;  $m$  is the number of possible strategies. Correspondingly,  $\mathbf{U}^k(t) = (u_1^k(t), \dots, u_m^k(t))$  is a vector of specified payoffs for player  $k$ . If player  $k$  plays action  $a_{i,1 \leq i \leq m}^k$ , he earns a payoff  $u_{i,1 \leq i \leq m}^k$  with probability  $p_i^k$ .  $\mathbf{P}^k(t) = \{p_1^k(t), \dots, p_m^k(t)\}$  is defined as player  $k$ 's probability distribution.

Actions chosen by the players are input to the environment and the environmental response to these actions serves as input to each player. Therefore, multiple players are connected in a feedback loop with the environment. When a player selects an action with his respective probability distribution  $\mathbf{P}(\cdot)$ , the environment produces a payoff  $U(\cdot)$  according to (1). Therefore,  $\mathbf{P}(\cdot)$  must be adjusted adaptively to contend with the payoff fluctuation. At every game round, all players update their probability distributions based on

the online responses of the environment. If player  $k$  chooses  $a_i^k$  at time  $t$ , this player updates  $\mathbf{P}^k(t+1)$  as follows:

$$p_j^k(t+1) = \begin{cases} f \left( p_j^k(t) + \psi \left[ \frac{u_j^k(t) - u_j^k(t-1)}{u_j^k(t-1)} \right] \right), & \text{if } j = l \\ \varphi p_j^k(t), & \text{if } j \neq l, \end{cases} \quad (5)$$

s.t.,  $\begin{cases} f(\chi) = 0, & \text{if } \chi < 0 \\ f(\chi) = \chi, & \text{if } 0 < \chi < 1 \\ f(\chi) = 1, & \text{if } \chi > 1, \end{cases}$

where  $\varphi$  is a discount factor and  $\psi$  is a parameter to control the learning size from  $p(t)$  to  $p(t+1)$ . In general, small values of  $\psi$  correspond to slower rates of convergence, and vice versa. According to (5),  $P^k(S' | S, a_S)$  is defined based on the Boltzmann distribution:

$$P^k(S' | S, a_S) = \frac{\exp((1/\lambda) p_{a_S}^k(t))}{\sum_{j \in \mathbb{A}_k} \exp((1/\lambda) p_j^k(t))}, \quad (6)$$

s.t.,  $a_S \in \mathbb{A}_k(t) = \{a_1^k(t), \dots, a_m^k(t)\}$ ,

where  $\lambda$  is a control parameter. Strategies are chosen in proportion to their payoffs; however, their relative probability is adjusted by  $\lambda$ . A value of  $\lambda$  close to zero allows minimal randomization and a large value of  $\lambda$  results in complete randomization.

**2.3. The Main Steps of Proposed Scheme.** To allow optimal movement in multischeduler systems, we consider the consequences of using the Markov game model by implementing the adaptive learning algorithm that attempts to learn an optimal action based on past actions and environmental feedback. Although there are learning algorithms to construct a game model, minimal research has been conducted on integrating learning algorithms with the decision-making process where players are uncertain regarding the real world and the influence of their decisions on each other.

In the proposed learning-based Markov decision process, a single QoS scheduler interacts with an environment defined by a probabilistic transition function. From the result of the individual learning experiences, each scheduler can learn how to effectively play under the dynamic network situations. As the proposed learning algorithm proceeds and the various

TABLE I: System parameters used in the simulation experiments.

Traffic class	Message application	Bandwidth requirement	Connection duration average/sec
I	Delay-critical emergency applications	32 Kbps	30 sec (0.5 min)
II	Event-related applications	32 Kbps 64 Kbps	120 sec (2 min) 180 sec (3 min)
III	General applications	128 Kbps 256 Kbps	120 sec (2 min) 180 sec (3 min)
IV	Multimedia applications	384 Kbps 512 Kbps	300 sec (5 min) 120 sec (2 min)
Parameter	Value	Description	
$\omega$	1.2	The player's willingness to pay for his perceived service worth	
$\epsilon$	-0.2	The control parameter between throughput and delay	
$q$	1.1	The estimation parameters of the cost function	
$\gamma$	0.3	A probability that the user keeps staying at the selected strategy	
$\Delta$	1	Predefined minimum bound for stable status	
$\xi$	1	The price for resource unit in the cost function	
$m$	3	The number of strategies for QoS schedulers	
$\psi$	1	A parameter to control the learning size	
$\varphi$	0.8	A discount factor for the respective probability distribution	
$\lambda$	1	A control parameter on the Boltzmann distribution	

```

Init ( )
{
  1:  $p(\cdot) = 1/m$ 
  2: Control parameter values ( $\omega, \epsilon, q, \gamma, p, \Delta, \psi, \varphi$  and  $\lambda$ ) are given from Table 1.
}
Main_QoS control ( )
{
  Start: Init ( );
  For (;) {
    3:  $U(x)$  is obtained from (1) and (2);
    4:  $p(\cdot)$  is adjusted by using (5);
    5:  $P(S' | S, a_S)$  is defined by using (6);
    6:  $a(t)$  is selected to maximize  $V(\cdot)$  based on (4).
    7: IF ( $\|V^{(t+1)}(\cdot) - V^{(t)}(\cdot)\| < \Delta$ ), Temp ( );
      ELSE continue;
    }
  }
  Temp ( )
  { 8: For (;) { IF ( $\|V^{(t+1)}(\cdot) - V^{(t)}(\cdot)\| < \Delta$ ), continue; ELSE break; }
  }
}

```

PSEUDOCODE 1: IoT system QoS control procedure.

actions are tested, the QoS scheduler acquires increasingly more information. That is, the payoff estimation at each game iteration can be used to update  $P(S' | S, a_S)$  in such a manner that those actions with a large payoff are more likely to be chosen again in the next iteration. To maximize their expected payoffs, QoS schedulers adaptively modify their current strategies. This adjustment process is sequentially repeated until the change of expected payoff ( $V(\cdot)$ ) is within a predefined minimum bound ( $\Delta$ ). When no further strategy modifications are made by all the QoS schedulers, the IoT system has attained a stable status. The proposed algorithm

for this approach is described by Pseudocode 1 and the following steps.

*Step 1.* To begin,  $p(\cdot)$  is set to be equally distributed ( $p(\cdot) = 1/m$ , where  $m$  is the number of strategies). This starting guess guarantees that each strategy enjoys the same selection probability at the start of the game.

*Step 2.* Control parameters  $\omega, \epsilon, q, \gamma, p, \Delta, \psi, \varphi$ , and  $\lambda$  are provided to each QoS scheduler from the simulation scenario (refer to Table 1).

*Step 3.* Based on the current IoT situation, each QoS scheduler estimates his utility function ( $U(x)$ ) according to (1) and (2).

*Step 4.* Using (5), each QoS scheduler periodically adjusts the  $p(\cdot)$  values.

*Step 5.* Based on the probability distribution  $P(\cdot)$ , each  $P(S' | S, a_s)$  is defined using the Boltzmann distribution.

*Step 6.* Iteratively, each QoS scheduler selects a strategy ( $a(t)$ ) to maximize his long-term expected payoff ( $V(\cdot)$ ). This sequential learning process is repeatedly executed in a distributed manner.

*Step 7.* If a QoS scheduler attains a stable status (i.e.,  $\|V^{(t+1)}(\cdot) - V^{(t)}(\cdot)\| < \Delta$ ), this scheduler is assumed to have obtained an equilibrium strategy. When all QoS schedulers achieve a stable status, the game process is temporarily stopped.

*Step 8.* Each QoS scheduler continuously self-monitors the current IoT situation and proceeds to Step 3 for the next iteration.

### 3. Performance Evaluation

In this section, we compare the performance of the proposed scheme with other existing schemes [4, 5, 13] and confirm the performance superiority of the proposed approach using a simulation model. Our simulation model is a representation of an IoT system that includes system entities and the behavior and interactions of these entities. To facilitate the development and implementation of our simulator, Table 1 lists the system parameters.

Our simulation results were achieved using MATLAB, which is widely used in academic and research institutions in addition to industrial enterprises. To emulate a real world scenario, the assumptions of our simulation environment were as follows.

- (i) The simulated system consisted of four QoS schedulers for the IoT system.
- (ii) In each scheduler coverage area, a new service request was Poisson with rate  $\rho$  (services/s) and the range of the offered service load was varied from 0 to 3.0.
- (iii) There were three strategies ( $m$ ) for the QoS schedulers and each strategy ( $a_{i,1 \leq i \leq m}$ ) was  $a_i \in \{25 \text{ Mbps}, 30 \text{ Mbps}, 35 \text{ Mbps}\}$ . Therefore, there were total  $m^N$ , that is,  $3^4$ , system states like  $\mathbf{S} = \{(25 \text{ Mbps}, 25 \text{ Mbps}, 25 \text{ Mbps}, 25 \text{ Mbps}), \dots, (35 \text{ Mbps}, 35 \text{ Mbps}, 35 \text{ Mbps}, 35 \text{ Mbps})\}$ .
- (iv) The resources of the IoT system, bandwidth (bps), and total resource amount were 140 Mbps.
- (v) Network performance measures obtained based on 50 simulation runs were plotted as a function of the offered traffic load.

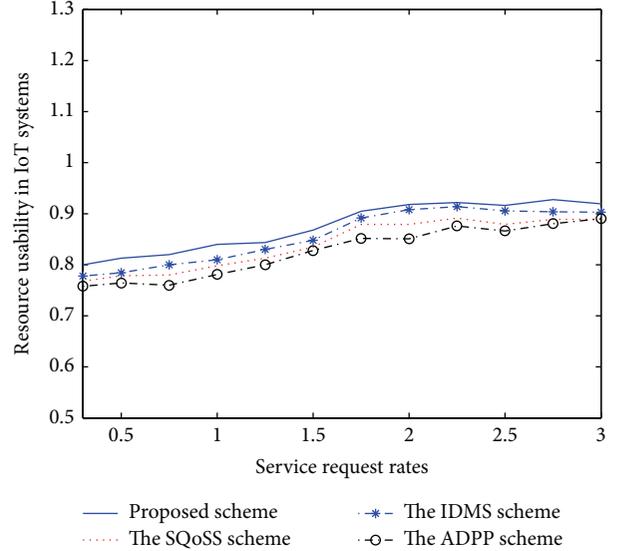


FIGURE 1: Resource usability in IoT systems.

- (vi) The message size of each application was exponentially distributed with different means for different message applications.
- (vii) For simplicity, we assumed the absence of physical obstacles in the experiments.
- (viii) The performance criteria obtained through simulation were resource usability, service availability, and normalized service delay.
- (ix) Resource usability was defined as the percentage of the actually used resource.
- (x) Service availability was the success ratio of the service requests.
- (xi) The normalized service delay was a normalized service delay measured from real network operations.

In this paper, we compared the performance of the proposed scheme with existing schemes: SQoSS [5], IDMS [4], and ADPP [13]. These existing schemes were recently developed as effective IoT management algorithms.

Figure 1 presents the performance comparison of each scheme in terms of resource usability in the IoT systems. In this study, resource usability is a measure of how system resources are used. Traditionally, monitoring how resources are used is one of the most critical aspects of IoT management. During the system operations, all schemes produced similar resource usability. However, the proposed scheme adaptively allocates resources to the IoT system in an incremental manner while ensuring different requirements. Therefore, the resource usability produced by the proposed scheme was higher than the other schemes from low to heavy service load intensities.

Figure 2 represents the service availability of each IoT control scheme. In this study, service availability is defined as the success ratio of the service requests. In general, excellent service availability is a highly desirable property for real

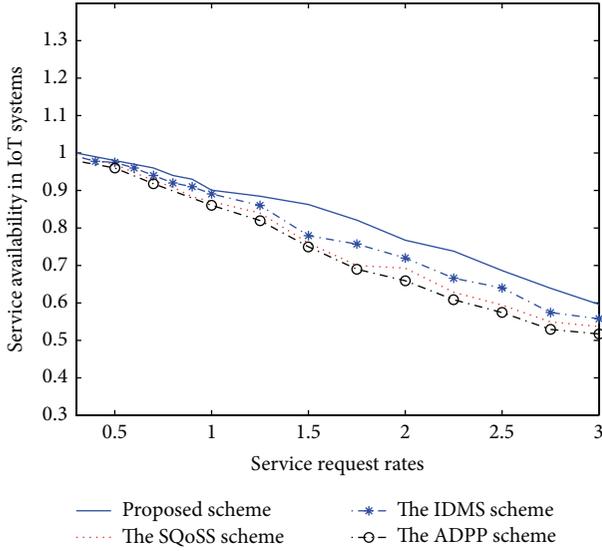


FIGURE 2: Service availability in IoT systems.

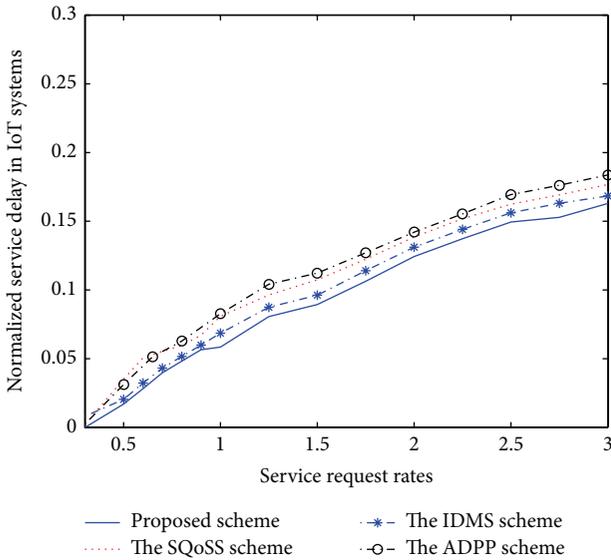


FIGURE 3: Normalized service delay in IoT systems.

world IoT operations. As indicated in the results, it is clear that performance trends are similar. As the service request rate increases, it can saturate or exceed the system capacity. Therefore, excessive service requests may lead to system congestion, decreasing the service availability. This is intuitively correct. Under various application service requests, the proposed game-based approach can provide a higher traffic service than the other schemes. From the above results, we conclude that the proposed scheme can provide a higher service availability in IoT systems.

The curves in Figure 3 illustrate the normalized service delay for IoT services under different service loads. Typically, service delay is an important QoS metric and can reveal the fitness or unfitness of system protocols for different delay-sensitive applications. Owing to the feedback-based Markov

game approach, the proposed scheme can dynamically adapt the current situation and has significantly lower service delay than the other schemes. From the results, we can observe that the proposed approach can support delay-sensitive applications and ensure a latency reduction in IoT services.

The simulation results presented in Figures 1–3 demonstrate the performance of the proposed and other existing schemes and verify that the proposed Markov game-based scheme can provide attractive network performance. The main features of the proposed scheme are as follows: (i) a new Markov game model based on a distributed learning approach is established, (ii) each QoS scheduler learns the uncertain system state according to local information, (iii) schedulers make decisions to maximize their own expected payoff by considering network dynamics, and (iv) when selecting a strategy, schedulers consider not only the immediate payoff but also the subsequent decisions. The proposed scheme constantly monitors the current network conditions for an adaptive IoT system management and successfully exhibits excellent performance to approximate the optimized performance. As expected, the performance enhancements provided by the proposed scheme outperformed the existing schemes [4, 5, 13].

#### 4. Summary and Conclusions

Today, IoT-based services and applications are becoming an integral part of our everyday life. It is foreseeable that the IoT will be a part of the future Internet where “things” can be wirelessly organized as a global network that can provide dynamic services for applications and users. Therefore, IoT technology can bridge the gap between the virtual network and the “real things” world. Innovative uses of IoT techniques on the Internet will not only provide benefits to users to access wide ranges of data sources but also generate challenges in accessing heterogeneous application data, especially in the dynamic environment of real-time IoT systems.

This paper addressed a QoS control algorithm for IoT systems. Using the learning-based Markov game model, QoS schedulers iteratively observed the current situation and repeatedly modified their strategies to effectively manage system resources. Using a step-by-step feedback process, the proposed scheme effectively approximated the optimized system performance in an entirely distributed manner. The most important novelties of the proposed scheme are its adaptability and responsiveness to current system conditions. Compared with the existing schemes, the simulation results confirmed that the proposed game-based approach could improve the performance under dynamically changing IoT system environments whereas other existing schemes could not offer such an attractive performance. Resource usability, service availability in IoT systems, normalized service delay, and accuracy were improved by approximately 5%, 10%, 10%, and 5%, respectively, compared to the existing schemes.

Furthermore, our study opens the door to several interesting extensions. In the future, we plan to design new reinforcement-learning models and develop adaptive online feedback algorithms. This is a potential direction and possible

extension to this study and can further improve the performance of IoT systems. Moreover, it would be interesting to extend the Markov game model to various decision-theoretic frameworks. Under uncertain system environments, this would be an interesting topic for future research.

## Conflict of Interests

The author, Sungwook Kim, declares that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) Support Program (IITP-2015-H8501-15-1018) supervised by the IITP (Institute for Information & Communications Technology Promotion) and by the Sogang University Research Grant of 2014 (201410020.01).

## References

- [1] G. Sallai, "Chapters of future internet research," in *Proceedings of the 4th IEEE International Conference on Cognitive Infocommunications (CogInfoCom '13)*, pp. 161–166, IEEE, Budapest, Hungary, December 2013.
- [2] K. Ashton, "That 'Internet of Things' thing in the real world, things matter more than ideas," *RFID Journal*, 2009, <http://www.rfidjournal.com/article/print/4986>.
- [3] Q. Wu, G. Ding, Y. Xu et al., "Cognitive internet of things: a new paradigm beyond connection," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 129–143, 2014.
- [4] Q. Zhang and D. Peng, "Intelligent decision-making service framework based on QoS model in the internet of things," in *Proceedings of the 11th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES '12)*, pp. 103–107, Guilin, China, October 2012.
- [5] L. Li, S. Li, and S. Zhao, "QoS-Aware scheduling of services-oriented internet of things," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1497–1507, 2014.
- [6] S. Kim, "Adaptive ad-hoc network routing scheme by using incentive-based model," *Ad Hoc & Sensor Wireless Networks*, vol. 15, no. 2, pp. 107–125, 2012.
- [7] G. Pujolle, "Metamorphic networks," *Journal of Computing Science and Engineering*, vol. 7, no. 3, pp. 198–203, 2013.
- [8] I. Jang, D. Pyeon, S. Kim, and H. Yoon, "A survey on communication protocols for wireless sensor networks," *Journal of Computing Science and Engineering*, vol. 7, no. 4, pp. 231–241, 2013.
- [9] J. van der Wal, "Discounted Markov games: successive approximation and stopping times," *International Journal of Game Theory*, vol. 6, no. 1, pp. 11–22, 1977.
- [10] P. Vrancx, K. Verbeeck, and A. Nowe, "Decentralized learning in Markov games," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 4, pp. 976–981, 2008.
- [11] K. Edemacu and T. Bulega, "Resource sharing between M2M and H2H traffic under time-controlled scheduling scheme in LTE networks," in *Proceedings of the 8th International Conference on Telecommunication Systems Services and Applications (TSSA '14)*, pp. 1–6, Kuta, Indonesia, October 2014.
- [12] X. Jin, S. Chun, J. Jung, and K.-H. Lee, "IoT service selection based on physical service model and absolute dominance relationship," in *Proceedings of the 7th IEEE International Conference on Service-Oriented Computing and Applications (SOCA '14)*, pp. 65–72, Matsue, Japan, November 2014.
- [13] X. Luo, H. Luo, and X. Chang, "Online optimization of collaborative web service QoS prediction based on approximate dynamic programming," in *Proceedings of the International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI '14)*, pp. 80–83, IEEE, Beijing, China, October 2014.
- [14] A. Imran, M. Bennis, and L. Giupponi, "Use of learning, game theory and optimization as biomimetic approaches for Self-Organization in macro-femtocell coexistence," in *Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW '12)*, pp. 103–108, IEEE, Paris, France, April 2012.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

