A Near-Optimal Time Slot Allocation Algorithm for Wireless Networks that Support Multiple Classes of Traffic

Frank Yeong-Sung Lin
Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.
yslin@im.ntu.edu.tw

Li-Yi Lin
Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.
r94041@im.ntu.edu.tw

Cheng-Ta Lee
Department of Information Management
National Taiwan University,
Lan Yang Institute of Technology
Taipei, Taiwan, R.O.C.
d90001@im.ntu.edu.tw

Abstract

Wireless communication networks provide convenience, however, also challenges to multimedia services due to typically limited bandwidth and various QoS (Quality-of-Service) requirements. For a wireless communication network service provider/administrator, it is then essential to develop an effective resource allocation policy so as to fully satisfy possibly different QoS requirements by different classes of traffic; while in the meantime, for example, the overall long-term system revenue rate can be maximized.

In this paper, we consider the problem of time slot allocation for multiple classes of traffic in wireless networks under throughput and delay constraints. To solve the problem, we propose an algorithm that is a novel combination of the Markovian decision process (MDP) and Lagrangean relaxation (LR). Another primal heuristic based on the policy enhancement algorithm is also developed for comparison purposes. Our experiment results show that the proposed approach can find a near optimal time slot allocation policy to maximize long-term system revenue under QoS requirements.

Keywords: Lagrangean relaxation, Markovian decision process, optimization, time slot allocation, wireless network

A Near-Optimal Time Slot Allocation Algorithm for Wireless Networks that Support Multiple Classes of Traffic

Abstract

Wireless communication networks provide convenience, however, also challenges to multimedia services due to typically limited bandwidth and various QoS (Quality-of-Service) requirements. For a wireless communication network service provider/administrator, it is then essential to develop an effective resource allocation policy so as to fully satisfy possibly different QoS requirements by different classes of traffic; while in the meantime, for example, the overall long-term system revenue rate can be maximized.

In this paper, we consider the problem of time slot allocation for multiple classes of traffic in wireless networks under throughput and delay constraints. To solve the problem, we propose an algorithm that is a novel combination of the Markovian decision process (MDP) and Lagrangean relaxation (LR). Another primal heuristic based on the policy enhancement algorithm is also developed for comparison purposes. Our experiment results show that the proposed approach can find a near optimal time slot allocation policy to maximize long-term system revenue under QoS requirements.

Keywords: delay, Lagrangean relaxation, Markovian decision process, optimization, time slot allocation, throughput, wireless networks

I. INTRODUCTION

In the last decade, the Internet has become increasingly important in our daily lives. With the growth in the number of users, many applications have been developed to provide more convenient services as well as entertainment. The demand for data transmission applications for web browsing and multimedia services has also increased dramatically.

New wireless networks technologies, such as the third-generation (3G) cellular system and IEEE 802.16 (Worldwide Interoperability for Microwave Access, WiMax), are designed to provide higher capacity for data services [10, 19]. Moreover, with the increasing demand for multimedia and other real-time transmission services, Quality of Service (QoS) has become a key issue that must be considered in the design of new wireless networks [3, 20, 22]. However, although the standards define the QoS architecture, the scheduling algorithm for a system with QoS-guaranteed transmission is not specified [2, 22].

Because of restricted bandwidth and QoS requirements, proper resource allocation is much more important in wireless networks than in wired networks. Better resource allocation policies allow wireless networks to achieve higher capacity utilization under the QoS requirements of each service class. Thus, a number of mechanisms, such as time-division multiplexing, frequency division multiplexing, time division multiple access, frequency division multiple access, and code division multiple access, have been developed to improve the utilization of channel capacity, [5, 7, 8, 11, 14, 15, 16].

In this paper, we discuss how a time slot system for resource allocation at wireless base stations (BS) can optimize the utilization of the capacity of wireless networks and also satisfy QoS requirements.

System revenue is considered in [7, 8, 15], but [15] does not address the QoS issue; while [7] and [8] only focus on the call blocking rate as a QoS requirement. However, to provide multimedia and other real-time services in wireless networks, it is not sufficient to consider the call blocking rate alone. Hence, we use delay and throughput requirements as QoS criteria. Since some systems have difficulty estimating delay, we use some approximations to estimate the delay of each class of service.

Quality of service, a key issue in multimedia transmission and other real-time services on the Internet, can be evaluated from a number of perspectives, such as throughput, delay, jitter, and reliability. The latest wireless networks are designed to support QoS-guaranteed transmission. For example, in 3G and IEEE 802.16, data packets are classified into several classes of service, each of which has different QoS requirements, [20, 21, 22]. In [5], two modes of bandwidth allocation for IEEE 802.16, namely, complete partitioning and complete sharing, are considered. With complete partitioning, a fixed amount of bandwidth is statically assigned to UGS (unsolicited grant service), while the remaining bandwidth is allocated to PS (polling service) and BE (best effort) services. In the case of complete sharing, when the bandwidth requirement for UGS traffic is less than the available bandwidth, the latter is allocated to PS. In [22], traffic priority is one of the QoS parameters considered. Given two service flows with identical QoS parameters except the priority, the higher priority service flow should be allocated a shorter delay and a higher buffering preference.

Many techniques can be used to improve QoS. For instance, "traffic shaping" smoothes traffic on the server side and can also be used for traffic policing to monitor traffic flow; "resource reservation" reserves resources, including bandwidth and buffer space, to ensure they are available for transmitting packets; and "admission control" allows a base station to decide whether to admit or reject the incoming traffic flow based on its own capacity and how many commitments it has already made to other flows [1]. Because of the importance of QoS, many resource allocation methods have been proposed in recent years [5, 7, 8, 11, 16].

In applications that provide QoS-guaranteed data transmission, delay and throughput requirements are normally used as QoS criteria. Many resource allocation methods had been proposed to maximize the utilization of the capacity and satisfy QoS requirements. Some approaches use a deadline, which is the acceptable delay, of each packet to allocate time slots [11, 16], while others base the allocation on the current queuing situation in the system [5]. Admission control is also used to control incoming traffic flows to ensure that the system can fully satisfy the QoS requirements of new flows as well as those already admitted [5, 11]. In cellular systems, the call blocking rate is an important criterion for evaluating QoS, [7, 8]. To summarize, the common purpose of the above approaches is to maximize the utilization of a system's capacity under given QoS requirements.

The remainder of this paper is organized as follows. Section II contains the problem description. In Section III, we formulate the problem as an analytical model. In Section IV, we propose a solution approach to the problem. Section 5, contains the experiment results. We also compare the proposed solution with a simple algorithm. Then, in Section VI, we present our conclusions.

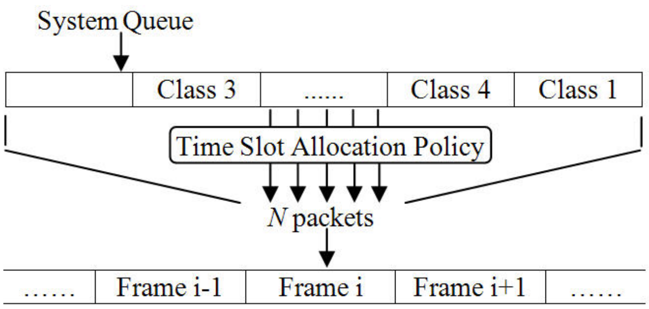

Fig.1. A system queue

II. PROBLEM DESCRIPTION

As shown in Figure 1, we consider a queuing system for a wireless base station (BS), where packets are classified into four service classes. The problem is to determine the best time slot allocation policy for each wireless base station in order to maximize the total system revenue under the delay and throughput constraints of each service class. When a BS has data to transmit to subscriber stations, the data packets must wait in a queue. The four service classes have different transmission priorities; hence, if there is not enough space in the queue for newly arrived packets, the packet with highest priority will join the queue and the packet with the lowest priority will be dropped, even it was already in the queue.

In the proposed model, the system state is defined by the number of packets of each service class in the queue. For example, state (4, 3, 2, 1) means the number packets in service classes 1, 2, 3, and 4 in the queue is 4, 3, 2, and 1 respectively. In each state, the system can transmit at most N packets in one frame. The different combinations of the four service class packets that can be transmitted in one frame are called "alternatives." For instance, suppose the current state of a system with four service classes is (1, 2, 0, 3) and the maximum number of packets that can be transmitted in a frame is 3; then, the alternatives of this state will be (1, 2, 0, 0), (1, 1, 0, 1), (1, 0, 0, 2), (0, 2, 0, 1), (0, 1, 0, 2), and (0, 0, 0, 3). The revenue derived will depend on the number and type of packets serviced, as packets in different service classes may yield different rewards.

III. PROBLEM FORMULATION

We assume that the arrival processes of the four service classes follow a Poisson distribution with different arrival rates and are mutually independent. Hence, the probability that x packets of a service class c will arrive in a particular frame can be calculated by the following Poisson distribution:

$$P_c(x)=\frac{e^{-\lambda_c}\lambda_c^x}{x!},\ c\in M,\ x=1,2,3,...$$

As the arrival processes of the four service classes are known, the state transition probability can be calculated by the function p_{ij}^k, which means a system currently occupying state i will occupy state j after its next transition given that the decision is k.

$$P\{\text{ state } i\ (q_1,q_2,...,q_m) \to \text{ state } j\ (q_1^{'},q_2^{'},...,q_m^{'})\ |$$

$$\text{when choosing the alternative } k,\ (n_1,n_2,...,n_m)\}$$

$$= P_{ij}^k$$

$$= \begin{cases} 0,\ \text{if } \{ \sum_{c\in M} q_c^{'} < B \text{ and } (q_c - q_c^{'} > n_c,\ \exists\ c \in M)\} \\ \quad \text{or } \{ \sum_{c=1}^{t} q_c^{'} = B \text{ and } q_t^{'} \geq 1 \text{ and} \\ \quad\quad (q_r - q_r^{'} > n_r,\ \exists\ r = \{1,...,t-1\}),\ t \in M - \{1\}\ \} & .....(1) \\ \prod_{c\in M} P_c(x = q_c^{'} - q_c + n_c),\ \text{if } \sum_{c\in M} q_c^{'} < B & .....(2) \\ P_1(x \geq B - q_1 + n_1),\ \text{ if } q_1^{'} = B & .....(3) \\ \{\prod_{c=1}^{t-1} P_c(x = q_c^{'} - q_c + n_c)\} \cdot P_t(x \geq \max(0,\ q_t^{'} - q_t + n_t)), \\ \quad \text{if } q_t^{'} \geq 1 \text{ and } \sum_{c=1}^{t} q_c^{'} = B,\ \text{for } t \in M - \{1\} & .....(4) \end{cases}$$

Approximation of the Queuing Delay

Because the queuing problem is very complex, it is difficult to estimate the queuing delay of each service class. In [4], Little's formulas are used to relate the steady state's mean system size to the average packet waiting time as follows. The queuing delay, W_q, can be calculated by

$$W_q = \frac{L_q}{\lambda},$$

where L_q is the average number of packets in the queue and λ is the arrival rate of the packets. One of the conditions of Little's formulas is that the system must be conservative, which means no packet in the queue will be dropped [4].

Therefore, we approximate the queuing delay for a service class, c, the approximation is derived by dividing the average number of packets belonging to c in the queue by the average number of transmitted packets of c in each time frame. When the drop rates of each service class are very low, the approximation is highly accurate. The error rate between the real delay and the approximated queuing delay will deteriorate if the drop rate of each service class increases.

The notations used to model the problem are listed below:

Given Parameters	
Notation	Description
M	The set of service classes
m	The number of service classes
λ_c	The arrival rate of service class c, c∈M
R_c	System revenue when servicing one class c packet, c∈M
N	The maximum number of packets that can be transmitted in a frame
B	The queue size of the system (in packets), where B≥N
S	The set of all states
K	The set of all alternatives
q_i^c	The number of packets belonging to service class c in state i, c∈M, i∈S
D_c	The delay requirement of service class c, c∈M
T_c	The throughput requirement of service class c, c ∈M
r_{ij}^k	The revenue required to change from state i to state j given decision k, $r_{ij}^k = \sum_{c \in M} n_i^c(k) R_c$
r_i^k	The expected system revenue of state i given decision k
$n_i^c(k)$	The number of packets belonging to service class c transmitted in state i if the decision of state i is k, $\sum_{c \in M} n_i^c(k) \le N$, $\forall\, c \in M$, $i \in S$
P_{ij}^k	When an alternative k has been chosen for state i, the probability from state i to state j after one state transition is P_{ij}^k

Decision Variables	
Notation	Description
d_i^k	Conditional probability of choosing alternative k given that the system is in state i
π_i	The limiting state probability of state i, which is independent of the initial state

The original problem can be reformulated as a Markovian decision problem with additional QoS constraints.

The Markovian decision process (MDP) is a dynamic programming application used to solve a stochastic decision process that can be described by a finite number of states. The transition probabilities between the states are described by a Markov chain. The reward structure of the process is also described by a matrix whose individual elements represent the revenue can be obtained by moving from one state to another. Both the transition and revenue matrices depend on the decision alternatives available to the decision-maker. The objective is to determine the optimal policy that maximizes the expected revenue of the process over a finite or infinite number of stages [9, 18].

We can formulate the Markovian decision process as a linear programming problem [6]. The objective function of the original problem is shown as follows:

Objective function:

$$Z_{\text{LP 1}} = \max \sum_{i \in S} \sum_{k \in K} \pi_i d_i^k r_i^k \qquad \text{(LP 1)}$$

Then, we reformulate the objective function of (LP 1) into a minimum form, which will not affect the original result, and the formulation is listed in the following:

Objective function:

$$Z_{\text{LP 2}} = \min \{ -\sum_{i \in S} \sum_{k \in K} \pi_i d_i^k r_i^k \} \qquad \text{(LP 2)}$$

Subject to:

$$\pi_j = \sum_{i \in S} \sum_{k \in K} \pi_i d_i^k p_{ij}^k \qquad \forall\ j \in S \qquad (1)$$

$$\pi_i \ge 0 \qquad \forall\ i \in S \qquad (2)$$

$$\sum_{i \in S} \sum_{k \in K} \pi_i d_i^k = 1 \qquad (3)$$

$$\sum_{j \in S} p_{ij}^k = 1 \qquad \forall\ i \in S,\ k \in K \qquad (4)$$

$$p_{ij}^k \ge 0 \qquad \forall\ i, j \in S,\ k \in K \qquad (5)$$

$$d_i^k \ge 0 \qquad \forall\ i \in S,\ k \in K \qquad (6)$$

$$\sum_{k \in K} d_i^k = 1 \qquad \forall\ i \in S \qquad (7)$$

$$r_i^k = \sum_{j \in S} p_{ij}^k r_{ij}^k \qquad \forall\ i \in S,\ k \in K \qquad (8)$$

$$r_i^k \ge 0 \qquad \forall\ i \in S,\ k \in K \qquad (9)$$

$$\frac{\sum_{i \in S} \sum_{k \in K} \pi_i d_i^k q_i^c}{\sum_{i \in S} \sum_{k \in K} \pi_i d_i^k n_i^c(k)} \le D_c \qquad \forall\ c \in M \qquad (10)$$

$$\sum_{i \in S} \sum_{k \in K} \pi_i d_i^k n_i^c(k) \ge T_c \qquad \forall\ c \in M\,. \qquad (11)$$

Explanation of the objective function:

The objective function of (LP) is to maximize the long-term system revenue when the system is stationary.

Explanation of constraints:

[1] Steady State Constraints:

Constraints (1), (2), and (3) are the steady state constraints of the system. Constraint (1) is $\pi = \pi P$, where $\pi = (\pi_0,\ \pi_1, \ldots)$ represents the limiting probability vector of the system state and P is the state transition probability matrix. Constraint (2) restricts the probability to a value larger or equal to zero Constraint (3) requires that the sum of all the limiting probabilities must be equal to 1. Constraints (2) and (3) jointly restrict the value of each π_i to between 0 and 1.

[2] State Transition Probability Constraints:
Constraints (4) and (5) relate to the state transition probability. Constraint (4) represents that when a state transits from i to all states with decision k, the sum of all the transition probabilities must be equal to 1. Constraint (5) restricts the transition probability P_{ij}^{k} to a value larger or equal to zero.

[3] Decision Making Constraints:
Constraints (6) and (7) are related to the decision variable d_i^k. In state i, the system chooses alternatives with different probabilities, and the sum of the probabilities is equal to 1. Constraint (6) restricts the probability d_i^k to a value larger or equal to zero.

[4] Revenue Constraints:
Constraints (8) and (9) relate to revenue calculation. Constraint (8) is used to calculate the expected revenue of transiting from state i if the corresponding decision is k, while Constraint (9) represents the restriction of r_i^k.

[5] QoS Constraints:
Constraints (10) and (11) stipulate the delay and throughput requirements of the four service classes. Note that the numerator of the queuing delay is $\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k q_i^c$ instead of $\sum_{i\in S}\pi_i q_i^c$. In the next section, we use $\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k q_i^c$ to reformulate the objective function because it simplifies the process.

IV. SOLUTION APPROACH

The Lagrangean relaxation (LR) method was first used to solve large-scale integer programming problems in the 1970s [12]. It can be used to solve complicated mathematical problems more efficiently and provide excellent solutions for such problems. Hence, the LR method has become one of the best tools for solving optimization problems, such as integer programming, linear programming with a combinatorial objective function, and non-linear programming problems [13, 17].

The primal problem (LP 2) is transformed into a LR problem in which Constraints (10) and (11) are relaxed. To relax Constraint (10), we multiply the both sides by the denominator on the left-hand side.

Objective function:

$$\begin{aligned}
&Z_D(\mu^D,\ \mu^T)\\
&=\min\{-[\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k r_i^k]\\
&+\sum_{c\in M}\mu_c^D(\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k q_i^c - D_c\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k n_i^c(k))\\
&+\sum_{c\in M}\mu_c^T(T_c-\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k n_i^c(k))\}\\
&=\min\{-\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k[r_i^k+\sum_{c\in M}(\mu_c^T n_i^c(k)\\
&+\mu_c^D(n_i^c(k)D_c-q_i^c))]+\sum_{c\in M}\mu_c^T T_c\}
\end{aligned} \qquad \text{(LR 1)}$$

Subject to Constraints (1) ~ (9)

The subproblem of this LR problem is exactly only one that is (SUB 1) subjected to constraints (1)~(9), which can be solved by Policy Iteration method, [6, 18]. The objective function of the subproblem is defined as follows.

Objective function:

$$\min\{-\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k[r_i^k+\sum_{c\in M}(\mu_c^T n_i^c(k)+\mu_c^D(n_i^c(k)D_c-q_i^c))]\} \qquad \text{(SUB 1)}$$

Subject to constraints (1) ~ (9)

Getting Primal Feasible Solutions:

After the subproblem has been solved optimally, we can obtain a set of decision variables and use them to develop a LR-based heuristic algorithm to find a near optimal feasible solution. The primal feasible solution is an upper bound (*UB*) of the problem (LP 2), and the Lagrangean elaxation dual problem is a lower bound (*LB*) of the problem (LP 2). The duality gap between *UB* and *LB*, computed by $|(UB - LB) / UB|*100\%$, indicates the optimality of the solution. Next, we consider the proposed heuristic.

The violation status of some QoS requirements may change during the decision adjustment procedure - a phenomenon known as "oscillation". In addition, if oscillations occur during the decision adjustment procedure, we may have changed too many decisions at each iteration. To reduce the probability that oscillations will occur, we can modify *decision_change_limit* by applying the adjustment rule.

TABLE 1. Heuristic-Phase Ⅰ: Feasible Solution

Heuristic-Phase Ⅰ: Feasible Solution
Step 1: Sort the steady state probabilities of each state from large to small. Step 2: Calculate the violation factor $violation\_c=\max(\frac{\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k q_i^c}{\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k n_i^c(k)}-D_c,\ T_c-\sum_{i\in S}\sum_{k\in K}\pi_i d_i^k n_i^c(k)),$ for $c\in M$. Note that violation_c≧0 means the relative constraint has been violated. Step 3: Stage 1: Perform the decision adjustment procedure beginning with the state that has the largest steady state probability. Stage 2: Randomly select states for decision adjustment. Decision Adjustment: For each selected state, move one time slot from the state that has the best performance to the state with the worst performance if possible. Then, select the next state for decision adjustment. Step 4: Calculate the delay and throughput performance of each service class.

IF (all constraints have been satisfied) {
Stop the procedure of Phase 1: Feasible_Solution;
Go to Phase 2: Objective_Value_Improvement;}
ELSE {
Adjust decision_change_limit according to the rule for adjusting decision_change_limit.
Calculate the steady state probabilities of each state;
IF (oscillation_limit has been reached)
Go to Step 2 and then take stage 2 of step 3.
ELSE
Go to Step 1 and then take stage 1 of step 3;
}

TABLE 2. Heuristic-Phase Ⅱ: Objective Value Improvement

Heuristic-Phase Ⅱ: Objective Value Improvement

Step 1:
Sort the steady state probabilities of each state.

Step 2:
while (is_feasible) {
/*start from the state that has the lowest steady state probability*/
Move one slot from the service class with the lower reward to the class with the highest reward;
IF (the new UB is better than the original UB)
Update the value of the UB;
Check feasibility;
/*if not feasible, then is_feasible = false
the final feasible solution is the primal solution to the problem*/
Next state;
}

TABLE 3. The Rule for Adjusting *decision_change_limit*

The Rule for Adjusting *decision_change_limit*

IF (no oscillation occurs) {//Y is a small number
IF ((*violation_Cn* $< Y$ for all $n \in M$) &&
(*decision_change_limit* $>$ *threshold_A*))
Set *decision_change_limit* to *threshold_A*;
ELSE
decision_change_limit remains the same;
}
ELSE {
IF (*threshold _B* $<$ *decision_change_limit* $<$ *threshold _A*)
Reduce *decision_change_limit* by one unit;
ELSE IF ((*violation_Cn* $< Y$ for all $n \in M$) &&
(*decision_change_limit* $>$ *threshold _A*))
Set *decision_change_limit* to *threshold_ A*;
ELSE
decision_change_limit = *decision_change_limit* / 2;
}

V. Computational Experiments

Simple Algorithm:

We compare our proposed iteration-based algorithm with a non-iteration-based algorithm that uses the "weight" to allocate slots to each service class. The "weight" of each service class considers the number of packets in queue of that class, as well as the throughput and delay requirements.

In each state, we first assign one slot to the service class with the highest weight, and then divide the corresponding weight by two, and repeat the assignment process until all slots have been assigned.

Scenario:

We use the following scenarios to evaluate the performance of our proposed algorithm under different parameter settings.

1. Different queue sizes under different revenue matrixes.
2. The performance under different QoS requirements.
3. The impact under different adjustments of *decision_change_limit*.

If the algorithm can not find a feasible solution, the objective value of the experiment will be set to zero.

The experiment results in Figures 2 to 5 show that the objective values depend on the throughput performance. Therefore, if the throughput requirements are relaxed, both LR and SA can find a feasible solution to the problem easily. However, when the throughput requirements are strict, it is much harder for SA to find a feasible solution.

The adjustment rule for *decision_change_limit* has three parameters: *threshold_A*, *threshold_B,* and the initial value of *decision_change_limit,* which can be modified to suit cases with a different total number of states. The experiment results show that different parameter settings only affect the objective values slightly, but they have a strong effect on the total number of iterations required to find a feasible solution

If *threshold_A* and *threshold_B* are increased, the total number of iterations will increase accordingly because there may be more oscillation in the decision adjustment procedure before feasible solutions can be found. In addition, if the initial value of the *decision_change_limit* is too small, more iterations may be needed to find feasible solutions because the improvement at each iteration is relatively small.

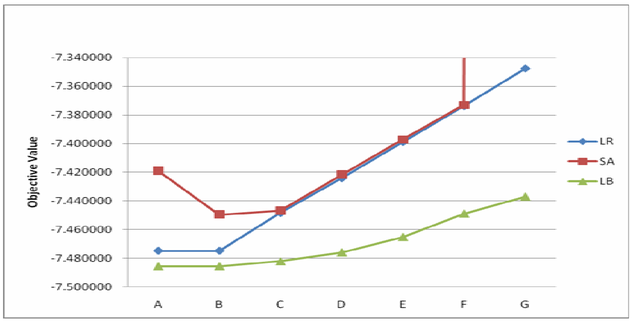

Fig. 2. Objective Values under Different QoS Requirements

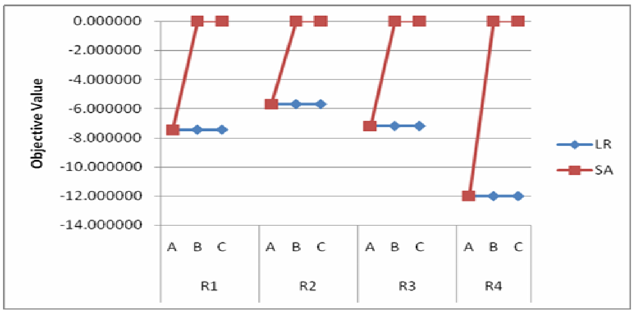

Fig. 3. Objective Values under different queue sizes

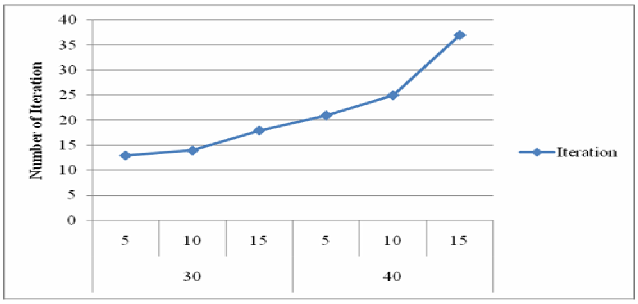

Fig. 4. The Number of Iterations for Different Adjustments of *decision_change_limit* (The Initial Value of *decision_change_limit* is 80)

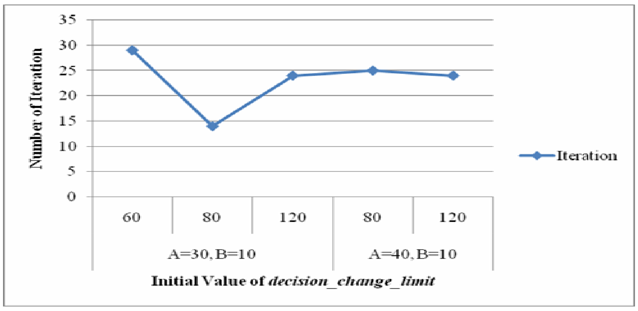

Fig. 5. The Number of Iterations for Different Initial Values of *decision_change_limit*

VI. CONCLUSIONS

We have formulated the problem of time slot allocation in wireless networks as a linear programming problem, where the objective function is to maximize long-term system revenue. In Section 3, we propose a Lagrangean Relaxation-based heuristic combined with the Markovian decision process to solve the problem. The total number of the policies increases dramatically as the queue size, number of service classes, and number of packets that can be transmitted in a frame become larger. For example, there are more than $9.5*10^{1469}$ different policies in the problem, such that the queue size, service classes, and maximum number of transmitted packets in one frame are 12, 4, and 6, respectively. Although the complexity of the problem is very high, our proposed approach can still find a near optimal feasible solution. The experiment results show that the proposed algorithm outperforms a simple algorithm in terms of finding a near optimal feasible solution to the problem. Moreover, the duality gaps of our proposed solution are smaller than 2%.

Since we know that the arrival rate dominates the throughput performance because of the occupancy priority, we can modify the occupancy rule for the queue space. For example, we can divide the queue into two parts. The packets in one part will not be dropped, even if a packet with higher occupancy priority wants to join the queue. We also consider the situation where packets of some service classes will not be dropped if they are already in the queue. The system discussed in this paper only has one queue. In our future work, we will try to extend the concept to a multiple-queue system for multiple communication channels in order to accommodate different types of networks.

REFERENCES

[1] Andrew S. Tanenbaum, "QUALITY OF SERVICE," *Computer Network*, 4th edition, pp. 397-417, Pearson Education, Inc. Publishing as Prentice Hall PTR, 2003.

[2] Carl Eklund, Roger B. Marks, Kenneth L. Stanwood, and Stanley Wang, "IEEE Standard 802.16: A Technical Overview of the WirelessMAN™ Air Interface for Broadband Wireless Access," *IEEE Communications Magazine*, pp. 98-107, June, 2002.

[3] Claudio Cicconetti, Luciano Lenzini, Enzo Mingozzi and Carl Eklund, "Quality of Service Support in IEEE 802.16 Networks," *IEEE Network*, March/April, 2006.

[4] Donald Gross Carl M. Harris "Fundamentals of Queueing Theory," 3rd edition, pp. 10-13 and 141-143, Wiley-Interscience Publication, 1998.

[5] Dusit Niyato and Ekram Hossain, "Queue-Aware Uplink Bandwidth Allocation and Rate Control for Polling Service in IEEE 802.16 Broad band Wireless Networks," *IEEE TRANSACTIONS ON MOBILE COMPUTING*, VOL. 5, NO. 6, JUNE, 2006.

[6] Hamdy A. Taha, "Markovian Decision Process," *Operations Research: An Introduction*, 5th edition, pp. 682-712, Macmillan Publishing Company, 1992.

[7] Hsu-Kuan Hung, ADVISER: Yeong-Sung Lin, "Optimization of GPRS Time Slot Allocation," June, 2001.

[8] Hui-Ting Chuang, ADVISER: Yeong-Sung Lin, "Optimization of GPRS Time Slot Allocation Considering Call Blocking Probability Constraints," June, 2002.

[9] K. Karen Yin, Hu Liu, and Neil E. Johnson, "Markovian inventory policy with application to the paper industry," *Computers and Chemical Engineering*, Vol. 26, NO. 10, pp. 1399-1413(15), October, 2002.

[10] K. W. Richardson, "UMTS overview," *ELECTRONICS & COMMUNICATION ENGINEERING JOURNAL*, June, 2000.

[11] Kitti Wongthavarawat and Aura Ganz, "Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems," *INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS*, 2003.

[12] Marshall L. Fisher, "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science*, Vol. 27, NO. 1, pp. 1-18, January 1981.

[13] Marshall L. Fisher, "An application Oriented Guide to Lagrangian Relaxation," *Interfaces*, Vol. 15, NO. 2, pp. 10-21, April 1985.
[14] Min Cao, Wenchao Ma, Qian Zhang, Xiaodong Wang, and Wenwu Zhu, "Modelling and Performance Analysis of the Distributed Scheduler in IEEE802.16 Mesh Mode," *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing MobiHoc '05*, May, 2005.
[15] Peter Marbach and Randall Berry, "Downlink Resource Allocation and Pricing for Wireless Networks," *IEEE INFOCOM*, 2002.
[16] Qingwen Liu, Xin Wang, and Georgios B. Giannakis, "A Cross-Layer Scheduling Algorithm With QoS Support in Wireless Networks," *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, VOL. 55, NO. 3, MAY 2006.
[17] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, "Lagrangian Relaxation and Network Optimization," *Network Flows: Theory, Algorithms, and Applications*, pp. 598-638. Prentice-Hall, Inc. 1993.
[18] Ronald A. Howard, "Dynamic Programming and Markov Processes," Wiley, 1970.
[19] Steven J. Vaughan-Nichols, "Achieving Wireless Broadband with WiMax," *Computer*, Vol. 37, Issue 6, June, 2004.
[20] Vijay K. Garg and Oliver T.W. Yu, "Integrated QoS Support in 3G UMTS Networks," *Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE*, Vol. 3, pp. 1187-1192, 2000.
[21] "IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems," *IEEE Computer Society and the IEEE Microwave Theory and Techniques Society*, 2004.
[22] http://www.umtsworld.com/technology/qos.htm