Service Level Agreement (SLA) Based Scheduling Algorithms for Wireless Networks

Mehdi Alasti, Farrokh R. Farrokhi, Masoud Olfat, and K. J. Ray Liu
Electrical and Computer Eng. Department , University of Maryland, College Park, MD 20742

Abstract— **The objective of scheduling algorithms in wireless networks is to provide Quality of Service (QoS) for mobile users in a shared environment and at the same time utilize the system resources efficiently. We have introduced a notion of income maximization to increase throughput for multimedia wireless systems and to maintain the QoS for each user above an agreed level. We have proposed greedy and dynamic programming approaches to solve the optimization problem. The simulation results reveal that our scheduling algorithms provide high network throughput, support QoS even under heavy network loads, and generate high income for service provider.**

I. Introduction

Scheduling algorithms that support Quality of Service (QoS) while maintaining high throughput for wireless networks are crucial to the development of broadband wireless networks. QoS refers to the capability of a network to provide certain services to selected network traffics. The four important attributes of QoS in packet networks are *dedicated bandwidth, controlled jitter/latency, and controlled loss characteristics.* Through the notion of effective bandwidth, it can be shown that a certain QoS level can be translated into a bandwidth guaranteed to a user [1]. Thus, we will represent the QoS by a bandwidth guaranteed to a user in this work.

Although many mature scheduling algorithms are available for wireline networks, they are not directly applicable in wireless networks due to major differences in medium. The time varying nature of wireless channels introduces some discontinuity in the availability of a user when the channel is in a bad condition. The very same nature of wireless channel provides opportunities for the transmission of large amount of information when the channel is in a good condition. On the other hand, if a scheduler operates independent of channel condition, it might allocate bandwidth to users in deep fade, where most of data is lost and bandwidth is wasted, while at the same time deprive users with good channel from taking advantage of their instantaneous large capacity.

Channel State Dependent Packet Scheduling (CSDPS) defers transmission of packets on links experiencing bursty errors [2]. A link status monitor, checks the channel condition for all mobiles, and when a channel is in a bad state, the scheduler does not serve the user associated with that link. Any one of the known wireline scheduling algorithms, e.g., round robin, earliest deadline first, and longest queue first, could be used as the service policies for this scheduling algorithm. However, CSDPS does not have any mechanism for supporting QoS (to guarantee bandwidth) for a mobile user.

Idealized Wireless Fair Queuing (IWFQ) is a modified version of Weighted Fair Queueing (WFQ) scheduling algorithm for wireless networks [4]. It uses an error-free WFQ as a reference system and tries to approximate the real service to the ideal error-free system. This algorithm provides some appealing properties in fairness and QoS guarantees. However, when a user is compensated for its previous lagged service, all other users with good channels will not be served at all.

Service Level Agreement (SLA) is a contract between a user and its service provider. An SLA defines the service (QoS) requested by the user, the price that the user must pay for the service, the penalty if the agreement is violated, and etc. This paper considers SLA as the reference point between the network and the network users. In scheduling algorithms, there is a trade-off between throughput maximization, which relates to the efficiency in utilizing bandwidth, and supporting QoS, which indicates how resources are shared among users. In this work, we introduce a notion of income maximization, by which the scheduler is rewarded when total network throughput is increased, and penalized when SLA for each user is violated. We will show that by properly choosing a penalty, as a function of SLA, and reward, as a function of network throughput, the trade-off is performed efficiently. We will also show that our algorithms meet the QoS and utilize network resources efficiently. We propose a greedy solution and a dynamic programming approach for the problem.

This paper is organized as follows: In Section II system model is introduced. SLA-based scheduling algorithms are proposed in Section III. The simulation results and performance comparison for different algorithms are presented in Section IV, and finally a summary of the contributions of this work is included in Section V.

II. System Model and Background

In this paper a single cell wireless network is investigated. We consider a time-slotted system, where time is the resource to be shared among all mobile users by a central processor. At any given time, only one user can be scheduled to occupy a given channel within a cell. The scheduling algorithm decides which time slot should be assigned to which user. At the down-link packets are queued at the base station; therefore, the scheduler at the base station has the full knowledge of the status of the queues. The block diagram of this system is shown Figure 1.

Let r_n denote the rate reserved by user n ($n = 1, 2, \cdots, N$), which is a fraction of the total available bandwidth ($0 \leq r_n \leq 1$). In fluid model, user n expects to receive

 0-7803-8533-0/04/$20.00 (c) 2004 IEEE

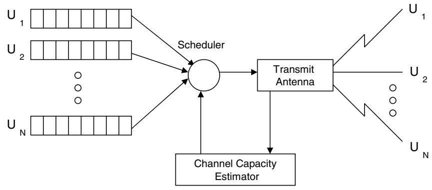

Fig. 1. System block diagram

a fraction of a time slot, r_n. However, in this work, we do not consider the fluid model and assume that a time slot is assigned only to one user. Define $Y_n(t)$ as

$$Y_n(t) = \begin{cases} 1 & \text{if the scheduler selects user } n \text{ at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

Also, assume that the indictor function, $I_n(t)$ is one when the queue of user n is non-empty at time t, and zero otherwise.

We assume that the link between each user and the base station is a wireless fading channel. In a power controlled system, the average power in each link is maintained at a fixed level and the instantaneous power follows a Rayleigh fading distribution. The Signal to Noise Ratio (SNR) for the n^{th} user is a function of the received power P_n, and the noise power N_n. The capacity per unit bandwidth for this user, is given by

$$B_n = \log_2(1 + P_n/N_n). \tag{1}$$

Since thermal noise at each receiver is fixed, the SNR at each user follows a Rayleigh distribution.

We assume that the link capacity is quantized to a limited number of levels. Let us assume that the channel capacity for user n at time t is denoted by $g_n(t)$, which is a fractional number. Therefore, the service received by user n at time t is $g_n(t)Y_n(t)$.

Next, we consider two scheduling algorithms. The first one is proposed to support QoS, and the other one to provide high network throughput.

A. Maximum Credit Scheduling (MCS)

In order to support QoS, a scheduler monitors and allocates resources in such a way that users' effective rates stay within a satisfactory range. A credit based mechanism can be used to measure and control the service provided to each user; the user n is assigned a credit, denoted by $C_n(t)$ ($n = 1, 2, \cdots, N$). A user's credit represents how much service the network owes to the user.

The credit for user n at time t evolves as follows:

$$C_n(t) = C_n(t-1) + I_n(t)r_n - g_n(t)Y_n(t). \tag{2}$$

The second term on the right hand side of the above equation, $I_n(t)r_n$, represents the service reserved by user n. If the n^{th} queue is non-empty, this term is the requested service. The third term represents the service received by user n. Starting from $C_n(-1) = 0$ for all users, by induction, it is straightforward to show that 2 leads to the following non-recursive expression for the credits:

$$C_n(t) = r_n \sum_{s=0}^{t} I_n(s) - \sum_{s=0}^{t} g_n(s)Y_n(s). \tag{3}$$

In the above equation, the first and second terms are the reserved and received service by user n up to time t, respectively. A negative credit means that a user has received a better service than what has been reserved. On the other hand, a positive credit implies that the network owes service to the user. Therefore, credit is a measure of how much service has been delivered or how much the service provider owes to a user. To support QoS, a scheduler must keep credits of all the users as small as possible. In this case, for users with non-empty queues the delivered service is close to their reserved services.

In order to minimize user credits, a Maximum Credit Scheduler (MCS) assigns the available bandwidth to the user with maximum credit [9]; in other words $Y_n(t) = 1$ only for $n = \arg\max_k\{C_k(t)$. Since the scheduling is based on the credit values at time t, and these credits are independent of the channel capacities at time t, the total throughput in this algorithm is equal to the average channel capacity, $\mathbf{E}[g_n(t)]$, when $\mathbf{E}[x]$ means the expectation of the random variable x.

B. Maximum Throughput Scheduling (MTS)

It has been proved that to maximize the network throughput, a scheduler must select the user with the best capacity or with the lowest fading among all the users [5]; , i.e. $Y_n(t) = 1$ only for $n = \arg\max_k\{g_k(t)$. The total throughput in this algorithm is equal to $\mathbf{E}[\max_n g_n(t)]$. However, this algorithm has no mechanism for supporting QoS.

C. A Trade-off: Throughput versus QoS

In MTS, a user that is trapped in a bad channel state, does not receive a service as long as its channel stays at that state. For this user, QoS or SLA is not satisfied. Thus, supporting QoS and maximizing network throughput cannot necessarily be achieved at the same time.

MCS does not face this trade-off in wired networks, since the channel responses of all users are equally good, i.e., $g_n(t) = 1$ for all n, t. Therefore, by selecting the user with the highest credit, the scheduler maintains the credits as small as possible. However, in wireless networks, attempting to support QoS for users with bad channel response may result in reducing network throughput. Another approach is to ignore users with the most eligible QoS that are in deep fade, in favor of users with better channel response with the hope that in the future the capacity of those ignored users would improve.

III. SLA-Based Scheduling Algorithms

In this work, we propose scheduling algorithms that resolves the trade-off between throughput and QoS based on the users SLA. SLA includes QoS, pricing for the service provided and penalty when the agreement is violated. Let us denote $d_n(t)$ to be the income of the service provider from user n at time t. Also, let us assume that for the service received by user n, i.e., $g_n(t)Y_n(t)$, the network charges the user by $\alpha_n g_n(t)Y_n(t)$, where α_n is the rate that the n^{th} user pays for the service. On the other hand, if the user has not received the requested QoS, its credit is increased and the network is penalized by $f_n[C_n(t)]$. We assume that $f_n[.]$ is a real, positive and continuous function with $f_n[x] = 0$ for $x \leq 0$. Both $f_n[.]$ and α_n are defined through the SLA between the service provider and user n. Thus, we obtain:

$$d_n(t) = \alpha_n g_n(t) Y_n(t) - I_n(t) f_n\left[C_n(t)\right]. \qquad (4)$$

If N is the number of users, the total income of the network at time t is given by $D(t) = \sum_{n=1}^{N} d_n(t)$. An SLA-based scheduler selects the user that increases the total income.

The penalty function has a significant role in the performance of a SLA-based algorithm. It is chosen in such a way that a user with negative credit does not penalize the system since this user has received its requested QoS; therefore, $f_n[x] = 0$ for $x \leq 0$. Also, if a user has accumulated a big credit, i.e., has received a poor QoS, it might be beneficial to disconnect this connection and pay the corresponding penalty. Moreover, we will expect that the penalty increases to be more significant for high credits. This means that $f_n[.]$ needs to be convex. We will see that for some special cases, the convexity of $f_n[.]$ is necessary [10]. One special example for $f_n[.]$ is:

$$f_n[x] = \begin{cases} \gamma_n x^2 & \text{if } x > 0, \\ 0 & \text{otherwise,} \end{cases} \qquad (5)$$

where γ_n is a positive number.

A. Maximum Income Greedy Scheduling

The Maximum income Greedy Scheduling (MIGS) algorithm selects the user that maximizes the total system income at each time slot t, $D(t)$.

Without loss of generality, from now on, we assume all users have non-empty queues, i.e., $I_n(t) = 1$. The following lemma summarizes the MIGS algorithm.

Lemma 3.1: The maximum income greedy scheduling algorithm selects the user that maximizes the following quantity over all the users with non-empty queues:

$$\begin{aligned} H_p(t) = \alpha_p g_p(t) + f_p[C_p(t-1) + r_p] \\ - f_p[C_p(t-1) + r_p - g_p(t)]. \end{aligned} \qquad (6)$$

Proof: Assume that user k is selected at time t, i.e., $Y_k(t) = 1$ and $Y_n(t) = 0$, $n \neq k$. In this case, we denote the total income at time t by $D^k(t)$, i.e.:

$$\begin{aligned} D^k(t) = \alpha_k g_k(t) - \sum_{\substack{n=1 \\ n \neq k}}^{N} f_n[C_n(t-1) + r_n] \\ - f_n[C_k(t-1) + r_k - g_k(t)]. \end{aligned} \qquad (7)$$

The MIGS selects the user p that maximizes the total income, i.e. $p = \arg\max_k \{D^k(t)\}$. That is, $Y_p(t) = 1$, and $Y_k(t) = 0, \ k \neq p$ and $D^p(t) \geq D^k(t), \ k \neq p$, where $D^p(t)$ and $D^k(t)$ are defined in (7). After simple manipulations, we obtain

$$H_p(t) \geq H_k(t). \qquad (8)$$

Therefore, MIGS maximizes (8) at time t. ■

Now consider a special case at time t, where a user, say k^{th} user, is in deep fade ($g_k(t) = \epsilon << 1$). Intuitively, we expect the scheduler to ignore such a user, because scheduling this user is equivalent to bandwidth loss at time slot t. Moreover, the small allocated rate ($g_k(t)$) is not large enough to support the QoS for this user. Because of continuity of the function $f_k[.]$, it can be shown that the metric associated with this user would be $\alpha_k \epsilon + \delta$, where δ is a small positive number. This means that if a user is in deep fade, it has a very small metric and is not expected to be selected by scheduler.

In the following, we'll consider two extreme cases. In one case only the system throughput is important, and in the other case, only the QoS matters. We will show that MIGS approaches these two cases, so it addresses the throughput-QoS trade-off.

1) Case I: Maximum Throughput Scheduling (MTS): Let us assume that the system would not be penalized for not supporting QoS, i.e., $f_n[.] = 0$. Also, for simplicity assume that the system charges different users with the same rate, i.e., $\alpha_k = \alpha$. Subsequently, as we discussed in Section II-B, in this case, the system tries to maximize the system throughput, and the scheduler selects the user with the best capacity. With these assumptions, the optimal user is selected as follows:

$$p = \arg\max_k \{\alpha g_k(t)\} \equiv \arg\max_k \{g_k(t)\}. \qquad (9)$$

2) Case II: Minimum Penalty Scheduling (MPS): Here, let us assume that the only goal of the system is to deliver QoS to the users, and the system throughput is not important. In this case $\alpha_n = 0$ for all the users, and therefore, the scheduling process will be:

$$p = \arg\max_k \{f_k[C_k(t-1) + r_k] - f_k[C_k(t-1) + r_k - g_k(t)]\}.$$

We expect this algorithm (we call it Minimum Penalty Scheduling, MPS) to support QoS for wireless networks.

Now, let us assume that $g_n(t) = 1$, as in wireline networks. Then, it can be shown that if $f_n[.]$ is a positive, continues, increasing and convex function, MPS can be simplified as [10]

$$p = \arg\max_k \{C_k(t-1) + r_k\}, \qquad (10)$$

which is similar to the MCS, mentioned in Section II-A.

B. Maximum Income Dynamic Programming Scheduling (MIDPS): Optimal Solution

The algorithms presented in the previous sections, maximize the total income locally. In this section, the objective is to maximize the system income, globally. In order to do so, dynamic programming algorithms are used to predict the future to make the decisions at the present time. In this framework, the optimization can be done within a finite horizon or infinite horizon policy [11]. We focus on the infinite horizon problem, since it provides the steady state policy which is independent of time. Define the expected total income as follows:

$$D = E\left\{\sum_{t=0}^{\infty} \beta^t \left[\sum_{n=1}^{N} D_n(t)\right]\right\}, \qquad (11)$$

where $0 < \beta \leq 1$ is the discount factor to keep the total income bounded.

Let the column vectors $\underline{C}(t), \underline{Y}(t)$, and $\underline{g}(t)$ represent the credits, scheduling decisions, and channel capacities of all N users at time t, respectively. Moreover, the vectors $\underline{r}$, and $\underline{\alpha}$ show the reserved rates, and reward rate of all users. Then,

$$\underline{C}(t+1) = \underline{C}(t) + \underline{r} - \underline{g}(t) \otimes \underline{Y}(t), \qquad (12)$$

where $\otimes$ denotes Hadamard product. Let $\{\underline{Y}\}$ and $\{\underline{g}\}$ denote the set of all $\underline{Y}(t)$ and $\underline{g}(t)$ for all times, then we want to maximize

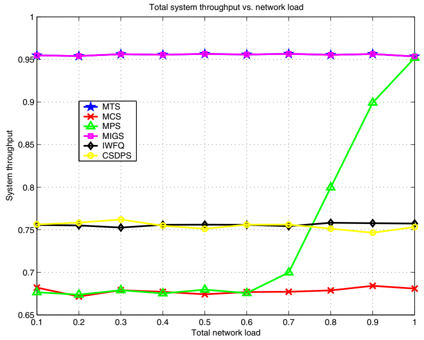

Fig. 2. Throughput vs. network load

$$\max_{\{\underline{Y}\}} \mathbf{E}_{\{\underline{g}\}} \left\{ \sum_{s=0}^{\infty} \beta^s \left[\sum_{n=1}^{N} \{\alpha_n g_n(s) Y_n(s) - f_n\left[C_{n+1}(s)\right]\} \right] \right\}.$$

Now, we define $G(\underline{C}(t), \underline{Y}(t), \underline{g}(t)) \triangleq D(t)$. Let S_t, the state of the system at time t be defined as the augmented vector $\underline{X}(t) = (\underline{C}(t), \underline{g}(t))$. Note that the scheduler knows the channel capacities at the decision time, and therefore, the channel capacities are part of the state vector. However, before time t, the vector $\underline{g}(t))$ is a random vector. At time $t = 0$ the system state is $\underline{X}(0)$. Then we have

$$J^*(\underline{X}(0)) \triangleq \max_{\{\underline{Y}\}} \mathbf{E}_{\{\underline{g}\}} \left[\sum_{t=0}^{\infty} \beta^t G(\underline{C}(t), \underline{Y}(t), \underline{g}(t)) | \underline{X}(0) \right]. \tag{13}$$

We would like to obtain the optimal policy $\underline{Y}^*(t) = \mu^* \left(\underline{C}(t), \underline{g}(t)\right)$ at each time slot t that maximizes (13). We can rewrite the optimal income in the form of Bellman's recursive equation for discounted infinite horizon problem at any time slot t [11], as follows:

$$\begin{aligned} J^*(\underline{X}(t)) = \max_{\underline{Y}(t)} \{ & G(\underline{C}(t), \underline{Y}(t), \underline{g}(t)) \\ & + \beta \mathbf{E}_{\underline{g}(t+1)} \left[J^*(\underline{C}(t+1), \underline{g}(t+1)) \right] \}. \end{aligned}$$

If we denote the probability of $\underline{g}(t+1) = \underline{g}^k$ by $\hat{p}_{\underline{g}^k}$, where $\underline{g}^k$ is the k^{th} level quantized value for the channel capacity, we obtain:

$$\begin{aligned} J^*(\underline{X}(t)) = \max_{\underline{Y}(t)} \{ & G(\underline{C}(t), \underline{Y}(t), \underline{g}(t)) \\ & + \beta \sum_k \hat{p}_{\underline{g}^k} J^* \left(\underline{C}(t+1), \underline{g}^k\right) \}. \end{aligned} \tag{14}$$

Starting from $\underline{X}(t)$, the scheduler selects the user that maximizes the righthand side of (14), in which the first terms represents the current income (as seen in the greedy algorithm, MIGS) and the second term represents the income-to-go. It can be shown that this maximization is equivalent to selecting the user n in the following maximization problem:

$$\begin{aligned} \max_n \{ & \alpha_n g_n(t) + f_n\left[C_n(t) + r_n\right] - f_n\left[C_n(t) + r_n - g_n(t)\right] \\ & + \beta \sum_k \hat{p}_{\underline{g}^k} J^*(\underline{C}(t) + \underline{r} - \begin{bmatrix} 0 \\ . \\ g_n(t) \\ . \\ 0 \end{bmatrix}, \underline{g}^k) \}. \end{aligned}$$

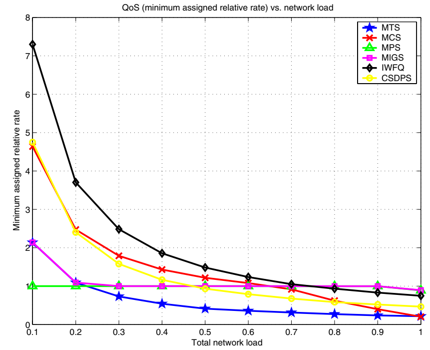

Fig. 3. Minimum assigned relative rate vs. network load

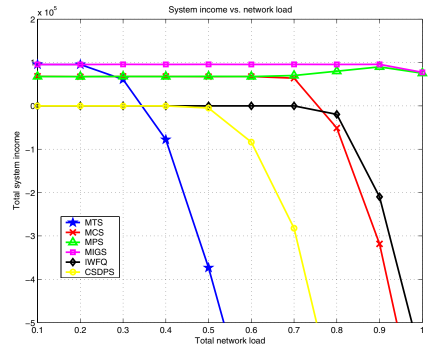

Fig. 4. Total income vs. network load

IV. Simulations Results

In order to evaluate the performance of our algorithms, we have simulated a single-cell wireless system where users are randomly distributed. We assume that path loss and shadow fading are compensated by a power allocation mechanism and the channel follows a Rayleigh fading distribution. By considering the same noise level at all receivers, the received signal power also follows a Rayleigh distribution. Here we have assume that number of quantized levels of channel capacities is $Q = 4$. These levels and their probabilities are $\{1.0, 0.6, 0.4, 0.2\}$ and $\{0.43, 0.24, 0.19, 0.14\}$, respectively.

If R_n is the assigned rate to user n, and r_n is the reserved rate by that user, we define the *minimum assigned relative rate* over all users as $\eta = \min_n\{\frac{R_n}{r_n}\}$. This value can be considered as a measure of QoS; to support QoS for all users, we want $\eta \geq 1$.

First we present the simulation results for MIGS and compare its performance with MTS, MCS, MPS, IWFQ, and CSDPS for a system with four users. The reserved rates of the four users are $\rho[0.1, 0.2, 0.3, 0.4]$, where $0 \leq \rho \leq 1$ is the network load. Also, we assume that $\alpha_n = 1000$ for all users.

Throughput, minimum assigned relative rate, and total income are plotted in Figures 2-4, respectively. The penalty function in the simulations is selected as in (5) with $\gamma_n = 1$. The horizontal axis in all these figures shows the network load. As illustrated in Figure 2, MTS and MIGS achieve the maximum throughput (the expectation of maximum link capacity, $\mathbf{E}\{\max(g) = 0.96\}$). MCS achieves a flat throughput which is equal to the average link capacity ($\mathbf{E}(g) = 0.68$). At low network loads, MPS tries to satisfy each user with

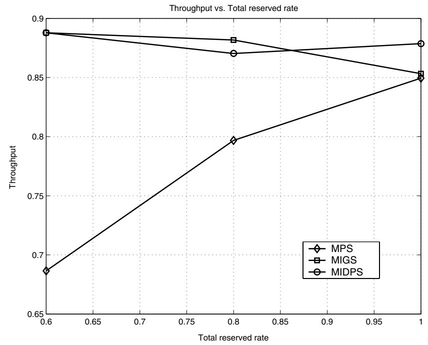

Fig. 5. Throughput of MIDPS, MIGS, and MPS vs. network load

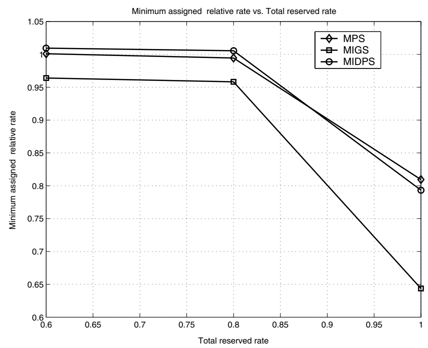

Fig. 6. Minimum assigned relative rate of MIDPS, MIGS, and MPS vs. network load

its requested bandwidth; that is, throughput is minimally allocated to satisfy each user. As network load increases, the system throughput increases and it approaches to that of MTS and MIGS. As illustrated in Figure 3, at low network loads all the algorithms support QoS. However, as network load increases, MPS and MIGS try to maintain QoS for all users, while MTS and CSDPS fail to do so. This result is expected since they are not designed to provide QoS. Since MCS does not utilize bandwidth as efficiently as MIGS, it fails at high network loads due to the lack of available channel bandwidth. As it was mentioned earlier, total income is a combination of system throughput and penalty when QoS requirement is not met. This quantity is shown in Figure 4. MIGS generates the highest income since the throughput and penalty are optimized jointly. The income for MTS, MCS, IWFQ, and CSDPS drop at high network loads, since they fail to meet QoS after certain loads. MTS and CSDPS fail to meet QoS at lower network loads compared to MCS and IWFQ, and thus, their incomes drop faster. Total income for MPS increases as load increases, since it tries to minimize penalty independent of load, while at large loads, throughput grows and increases the income. At high network loads, MPS income approaches that of MIGS, since both achieve similar throughput at high loads.

Next, we evaluate the performance of MIDPS and compare it with performance of MIGS and MPS (See Figures 5, 6 and 7). However, because of complexity issue of DP algorithm, we consider only three users, limit the credits of users to be between -1 and 2, and perform the simulations for three cases where the reserved rates are a:$[0.2, 0.2, 0.2]$, b:$[0.2, 0.2, 0.4]$, and c:$[0.2, 0.2, 0.6]$. The penalty function is described by (5), and $\alpha_1 = 1$, $\alpha_2 = 2$, and $\alpha_3 = 4$. Figures 5 and 6 show that

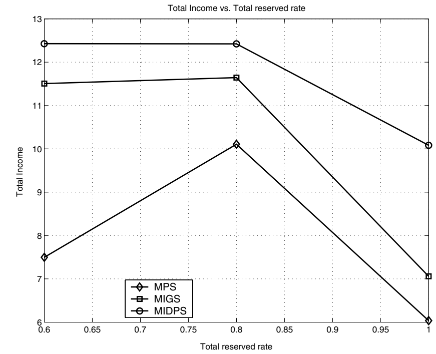

Fig. 7. Income of MIDPS, MIGS, and MPS vs. network load

the system throughput and QoS with MIDPS are as good as the throughput and QoS with MIGS. Therefore, MIDPS can support QoS and provide high system throughput. However, as shown in Figure 7, the total income with MIDPS is better than the sub-optimal MIGS. We have to mention that when we increase the range of credits, sub-optimal solution MIGS performs close to the optimal solution MIDPS.

V. Conclusion

In this work we have proposed Service Level Agreement (SLA) based scheduling schemes. We have introduced a notion of income maximization where throughput is the objective of maximization with the constraint that the scheduler is penalized when the QoS or SLA is violated. We have proposed a greedy approach and a dynamic programming approach to solve the problem. Our results show that the performance of the algorithm is superior to cases where only throughput or QoS is considered in the scheduling process.

References

[1] R. Guerin, H. Ahmadi, and M. Naghsineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE Journal of Selected Areas in Communications,*, Vol. 9, No. 7, September 1991.

[2] P. Bhagwat, A. Krishna, S. Tripathi, "Enhancing Throughput over Wireless LANs Using Channel State Dependent Packet Scheduling," Proc. INFOCOMM96, March 1996, pp. 1133-1140.

[3] C. Fragouli, "Controlled multimedia wireless link sharing via enhanced class-based queuing with channel-state dependent packet scheduling," *Proceedings of INFOCOM'98,* Vol. 2, March 1998, pp. 572-580.

[4] Yaxin Cao, Victor O. K. Li, Zhigang Cao, "Scheduling Delay Sensitive and Best-Effort Traffic in Wireless Networks," ICCC 2003,

[5] M. Grossglauser, D. Tse, "Mobility Increases the Capacity of Wireless Adhoc Networks," IEEE INFOCOMM01, April 2001.

[6] Dapeng Wu, Rohit Negi, "Utilizing Multiuser Diversity for Efficient Support of Quality of Service over a Fading Channel," ICCC 2003,

[7] S. Lu, V. Bharghavan, "Fair Scheduling in Wireless Packet Networks," IEEE/ACM Trans. Networking, Vol. 7, No. 4, pp. 473-489, 1999.

[8] Xin Liu, Edwin K. P. Chong, Ness B. Shroff, "Opportunistic Transmission Scheduling With Resource-Sharing Constraints in Wireless Networks," IEEE Journal of Selected Areas of Communications, Vol. 19, No. 10, pp 2053-2064, Oct. 2001.

[9] A. C. Kam, K. Y. Siu, "Linear Complexity Algorithms for Bandwidth Reservatations and Delay Guarantees in Input Queued Switches with no Speedup," Proc. International Conference on Network Protocols 98, Austin TX, Oct. 1998, pp. 2-11.

[10] M. Olfat, "Spatial Processing, Power Control, and Channel Allocation for OFDM Wireless Communications," phD Thesis, November 2003.

[11] D. P. Bertsekas, "Dynamic Programming And Optimal Control," Athena Scientific, 1995.