Resource Allocation in CDMA Packet Systems with Maximum Rate Constraint

C. Manikandan and Srikrishna Bhashyam
Department of Electrical Engineering,
Indian Institute of Technology, Madras, Chennai, India 600036
mani2004c@gmail.com, skrishna@ee.iitm.ac.in

***Abstract*— Scheduling algorithms in CDMA packet systems need to allocate codes and power based on channel state information and queue information. For finite queue sizes, the allocated rate should be less than or equal to the maximum rate required to clear the whole queue. This ensures that resources are not under-utilized. Although queue size and head-of-the-line packet delays have been considered in existing scheduling methods, this maximum rate constraint has not been explicitly enforced. In this paper, we propose a new scheduling algorithm incorporating the maximum rate constraint. The proposed algorithm performs better in terms of delay distribution and percentage of dropped packets compared to existing algorithms that do not directly enforce the maximum rate constraint.**

I. Introduction

Resource allocation algorithms that schedule a single user in any given time slot were initially proposed in [1], [2] for CDMA packet systems like CDMA2000 HDR. In these algorithms, throughput is maximized while achieving some fairness objective. Proportional fairness is achieved in [1]. In [2], head-of-the-line packet delay information is used along with channel state information to derive a throughput optimal single-user scheduling policy named Modified Largest Weighted Delay First (MLWDF) rule. However, single-user scheduling algorithms have been shown to be sub-optimal in [3], [4]. Multi-user scheduling is considered in [3], [4], [5] in order to maximize the weighted sum of rates. This weighted sum of rates can capture a large class of fairness measures including proportional fairness and MLWDF fairness.

When users have an infinite backlog of traffic, the optimal code and power allocation is proposed in [4]. However, for a finite queue length case with random packet arrivals, the scheduling algorithm in [4] may allocate more than the maximum rate required to clear the queue of a scheduled user. Since this maximum rate constraint is not explicitly enforced, resources are under-utilized. Queue information can be used in the algorithm in [4] by choosing the weights in the objective function to be the weights from the MLWDF [2], [5] algorithm. However, this will still not bridge the gap between the allocated and the actual rate as queue information is not fully used in defining the MLWDF weights.

In this paper, we propose a new multi-user scheduling algorithm which takes queue information into account effectively by incorporating the maximum rate constraint in the objective function and not in the form of weights. Therefore, the allocated rate to any user is always less than or equal to the actual maximum rate required to clear the user's queue. Incorporating the maximum rate constraint explicitly is shown to be better than using the queue information in the weights in terms of the packet delay distributions and the packet dropping probability. The proposed algorithm is obtained by modifying the code allocation procedure in the algorithm in [4]. The other steps from the algorithm in [4] are not modified.

II. System model

Assume that there are N users in the cell and each user is associated with a queue. The arrival distribution of packets in the queue is assumed to be Bernoulli. Time is divided into slots and in each slot multi-user scheduling decisions are taken. The channel is assumed to be constant for the duration of the slot. We assume a finite queue size. Therefore, packets are dropped if the queue is full. In our model, we maximize the weighted sum of rates subject to power, code and maximum rate constraints.

III. Problem formulation

Our problem formulation is similar to that in [4] except for the maximum rate constraint enforced by Q_i.

$$\text{maximize} \sum_i w_i \min(R_i, Q_i) \tag{1}$$

subject to

$$\sum_i p_i \leq P, \quad \sum_i n_i \leq N, \quad n_i \leq N_i \tag{2}$$

$$\check{s}_i(n_i) \leq \frac{p_i e_i}{n_i} \leq s_i(n_i), \tag{3}$$

where $R_i = n_i \log_e \left(1 + \frac{p_i e_i}{n_i}\right)$,
w_i is the weight associated with the i^{th} user,
p_i is the power allocated to the i^{th} user,
n_i is thenumber of codes allocated to the i^{th} user,
P is the total power,
N is the total number of codes,
N_i is the per user code constraint,
R_i is the rate allocated to the i^{th} user,
Q_i is the maximum rate constraint associated with the i^{th} user,
e_i is the channel SNR of the i^{th} user,
$\check{s}_i(n_i)$ is the lower bound for SINR, and
$s_i(n_i)$ is the upper bound for SINR.

If the n_i's are allowed to be real numbers, this is a convex optimization problem and can be readily solved using the primal-dual method [7]. Also, since Slater's condition [6] is satisfied, there will be no duality gap. The Lagrangian is formulated as

$$L(\lambda, \mu, n, p) = \sum_i w_i \min(R_i, Q_i) + \lambda(P - \sum_i p_i) + \mu(N - \sum_i n_i).$$

Although the maximum rate constraint has been mentioned in [4] as a special case of the constraint on SINR, the algorithm proposed in [4] assumes that the SINR upper and lower bounds are independent of n_i, i.e., they are identical irrespective of the code allocation. Therefore, maximum rate constraints based on queue size are not possible.

IV. Existing Algorithm

The algorithm in [4] can be summarized as follows.

1) For a given value of the Lagrange multiplier λ, find the optimal code allocation.
2) For a given code allocation, calculate the optimal power allocation.
3) Using the results of steps 1 and 2, calculate the dual solution $L(\lambda, \mu^*, n^*, p^*)$.
4) Update λ using golden search, and repeat steps 1 to 3 until the dual solution does not decrease significantly.

V. Proposed Algorithm

In this paper, we modify only the code allocation in step 1 to incorporate the maximum rate constraint as explained below. For simplicity, we discuss only the case of $\check{s}_i(n_i) = 0$ and $s_i(n_i) = \infty$. The algorithm can be easily generalized for arbitrary $\check{s}_i(n_i)$ and $s_i(n_i)$.

As mentioned earlier, if the n_i's are allowed to be real numbers, then we have a convex optimization problem that can be solved using the primal-dual method. The above problem can be restated as

$$\max_{n_i} \left(\max_{p_i(n_i)} \sum_i w_i \min(R_i, Q_i) \right), \tag{4}$$

where the power allocation is optimized for a given code allocation in the inner maximization.

A. Finding the optimal power for a given optimal code allocation

The optimal power allocation for a given code allocation is given by [4]

$$p_i^* = \frac{n_i}{e_i} \max\Big(\frac{w_i e_i}{\lambda} - 1, 0\Big). \tag{5}$$

From the complementary slackness condition, and assuming that λ is not equal to zero, we should have

$$\sum_i p_i^* = P. \tag{6}$$

Substituing the value of p_i^* from equation (5), we can find the optimal value of λ.

B. Optimizing the code allocation

Without the maximum rate constraint, substituting the expression for p_i^* in terms of n_i, we get

$$L(\lambda, \mu, n, p^*) = \sum_i (w_i n_i h(w_i e_i, s_i(n_i), \check{s}_i(n_i), \lambda) - \mu n_i) + \lambda P + \mu N, \tag{7}$$

where

$$h(w_i e_i, s_i(n_i), \check{s}_i(n_i), \lambda) = \frac{\lambda}{w_i e_i} - 1 - log_e \frac{\lambda}{w_i e_i}, \quad 0 \le \lambda < w_i e_i \tag{8}$$

Now, maximizing the lagrangian over the set of codes is equivalent to maximizing the objective function on a code by code basis resulting in allocation of the code to the best user. In [4], the objective function grows linearly with n_i. For a given i, the objective function increases by x_i for each allocated code, where x_i is defined as

$$x_i = \begin{cases} w_i \left(\frac{\lambda}{w_i e_i} - 1 - \log_e \frac{\lambda}{w_i e_i}\right) & 0 \le \lambda < w_i e_i \\ 0 \quad \text{else.} \end{cases} \tag{9}$$

Therefore, to allocate codes, the x_i's are arranged in descending order and codes are allocated in that order according to the per-user code constraint for each i till the codes are exhausted.

Now, we describe how the maximum rate constraint is incorporated. If n_i^* can be a real number, we would choose n_i^* such that

$$n_i^* \le \frac{Q_i}{\log_e\left(\frac{w_i e_i}{\lambda}\right)} \quad 0 \le \lambda < w_i e_i, \tag{10}$$

We incorporate the integer constraint as follows.

$$\text{Let } N_{i,max} = \left\lfloor \left(\frac{Q_i}{\log_e\left(\frac{w_i e_i}{\lambda}\right)} \right) \right\rfloor + 1, \quad 0 \le \lambda < w_i e_i. \tag{11}$$

The number of codes allocated to user i should definitely be lesser than or equal to $N_{i,max}$. Therefore, we can modify the per-user code constraint based on the queue information using N_i^* instead of N_i as:

$$N_i^* = \min(N_i, N_{i,max}), \tag{12}$$

and

$$n_i^* \le N_i^*. \tag{13}$$

Now, if $N_i^* = N_i$ for all i, i.e., there is enough backlog in the queues, then the code allocation in [4] can be used without any modification. If $N_i^* = N_{i,max}$, the code allocation procedure has to be modifed.

With the maximum rate constraint, the objective function grows linearly only upto $n_i = N_{i,max}$. For the first $N_{i,max} - 1$ codes, the increase in the objective function is x_i as defined before. For the last code, the increase in the objective function is y_i, where

$$y_i = \begin{cases} \frac{\lambda}{e_i} - w_i + w_i \left(Q_i + N_i^* \log_e \left(\frac{\lambda}{w_i e_i} \right) \right) & 0 \leq \lambda < w_i e_i \\ 0 \qquad \text{else.} \end{cases} \tag{14}$$

Note that y_i is always less than or equal to x_i. In order to determine whether to use $N_{i,max}-1$ codes for user i or $N_{i,max}$ codes for user i, we treat each user as two virtual users with code constraints $N_{i,max}-1$ and 1. Then, we arrange the x_i's and y_i's together in descending order and allocate in that order using the new code constraint, until all the codes are exhausted.

The algorithm is summarized as below:

1) Calculate N_i^* and x_i for all the users.
2) If $N_i^* = N_i$, set $y_i = 0$, else calculate y_i using equation (14).
3) Sort the x_i's and y_i's of all the users together in descending order.
4) Keep only the values that are ≥ 0.
5) Let N_{rem} be the remaining number of codes.
6) While $N_{rem} > 0$, do
   - If the next entry in the truncated sorted list is one of the x_i's and $N_i^* = N_{i,max}$, allocate $\min(N_{i,max}-1, N_{rem})$ codes. If $N_i^* = N_i$, then allocate $\min(N_i, N_{rem})$ codes as in [4].
   - If the next entry in the truncated sorted list is one of the y_i's, allocate one code.

   End.

C. Optimizing over λ

$$\lambda^* = arg \min_{\lambda} L(\lambda, \mu^*, n^*, p^*) \tag{15}$$

The optimal λ is found using the golden search algorithm. The starting interval is given in [4]. The golden search algorithm is used to find the global minimum of the convex function $L(\lambda, \mu^*, n^*, p^*)$ as a function of λ. The golden search algorithm is summarized as follows:

1) Start with two extreme points, say a and b, and two intermediate points, say a_1 and b_1, between a and b.
2) Compute the function at a_1 and b_1.
3) If the functional value at a_1 is less than the function value at b_1, then set $b = b_1$ and repeat the algorithm, else, set $a = a_1$ and repeat the algorithm from step 1.
4) Stop when the functional value at the intermediate points are fairly close.
5) At each iteration of the algorithm, we are narrowing down the interval to get the global minimum.

VI. Simulation Results

A downlink of a CDMA packet data system like HSDPA is assumed with the following simulation parameters. The number of users is 7. The arrival traffic for the users are assumed to be independent and identically distributed bernoulli packet generation processes with probability of packet generation in each timeslot λ. The packet sizes and the queue buffer size are 10 kbits and 100 packets respectively. The channel is assumed to be constant in each timeslot (scheduling interval). The scheduling interval is 2ms. Each user has an average SNR of -3 dB with the standard Jakes fading channel model with a vehicle speed of 25 kmph. The total transmit power in the system is 11.9 W and the total number of codes in the system is 15. Each user can be allocated a maximum of 5 codes. The simulations are for a duration of 10,000 slots (20 sec).

A. Delay Distribution

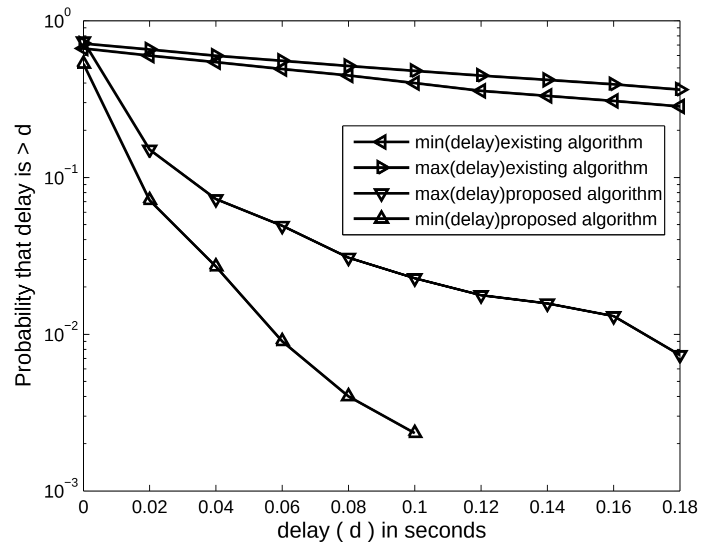

Fig. 1. Delay Distribution with equal weights, $\lambda = 0.3$

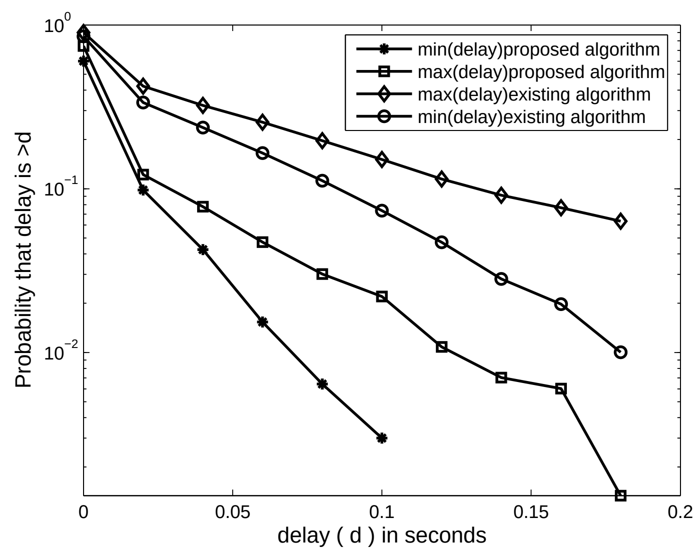

Fig. 2. Delay Distribution with MLWDF weights, $\lambda = 0.3$

The delay of all the transmitted packets for all the users is calculated and the best and worst case delay distributions are plotted for the algorithm in [4] and the proposed algorithm.

The result with equal weights and MLWDF weights have been shown in the Fig 1 and Fig 2 respectively. We observe that the proposed algorithm gives significantly better delay performance than the existing algorithm [4] in both cases.

B. Percentage of Dropped Packets

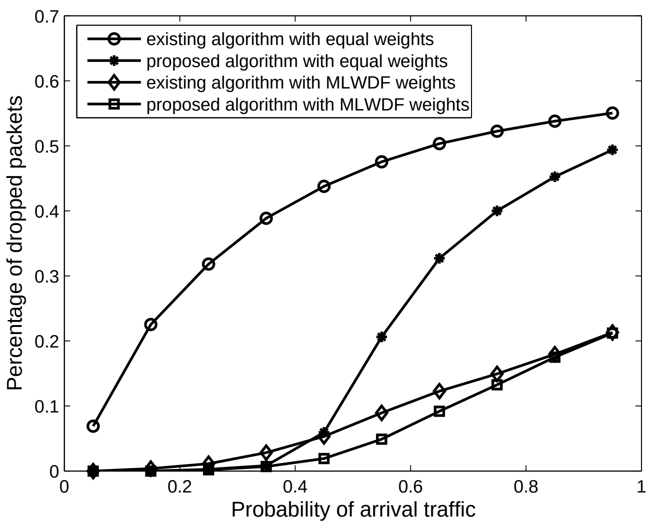

Fig. 3. Percentage of dropped packets

The percentage of dropped packets for a given arrival traffic distribution is calculated for all users and has been plotted in Figure 3 for the user with the maximum packet drop rate. Again, the proposed algorithm is compared with the algorithm in [4] for the case of equal weights and for MLWDF weights. The percentage of dropped packets is lesser in the existing algorithm when the MLWDF weights are used. However, the proposed algorithm provides further improvement by completely eliminating under-utilization of allocated resources. Therefore, the proposed algorithm can support larger stable traffic.

C. CDF of Rates

The CDF of rates for a given arrival traffic distribution is calculated and plotted in Figure 4. Both the actual rate and the allocated rate for the proposed algorithm are compared with the algorithm in [4] for the case of equal weights and for MLWDF weights. Resources are not wasted in the proposed algorithm as the allocated rate is close to the actual rate.

VII. CONCLUSION

In this paper, we have proposed a scheduling algorithm optimal for finite queue lengths. The queue information is used in the objective function as a maximum rate constraint. The proposed algorithm gives better delay performance and supports larger arrival traffic with no dropped packets. This work can also be extended to scheduling in multi-user Orthogonal Frequency Division Multiplexing systems.

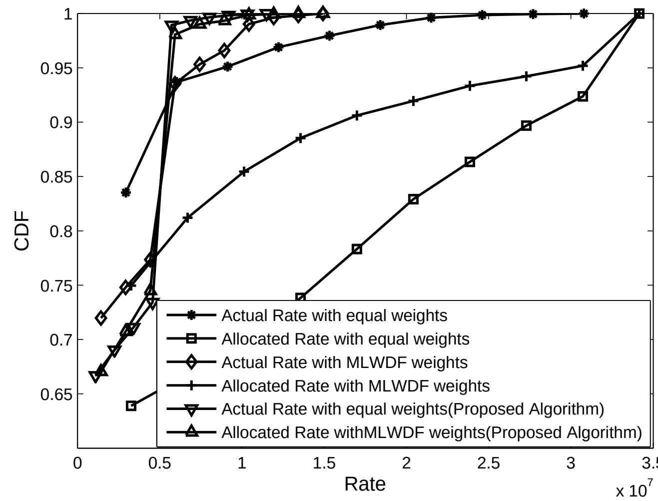

Fig. 4. CDF of Rates, $\lambda = 0.3$

REFERENCES

[1] E. F. Chaponniere, P. Black, J. M. Holtzman, D. Tse, "Transmitter directed multiple receiver system using path diversity to equitably maximize throughput," *U. S. Patent No. 6449490*, Sep 2002.

[2] M. Andrews, K. Kumaran, K. Ramanan, A. L. Stolyar, R. Vijayakumar, P. Whiting, "Providing quality of service over a shared wireless link," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 150-154, Feb 2001.

[3] K. Kumaran, H. Viswanathan, "Joint power and bandwidth allocation in downlink transmission," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1008-1016, May 2005.

[4] R. Agarwal, V. Subramanian, R. Berry, "Joint scheduling and resource allocation in CDMA systems," 2^{nd} *workshop on modeling and optimization in mobile, adhoc and wireless networks (WiOPT '04)*, Cambridge, UK, Mar 2004.

[5] V. Sandeep, S. Bhashyam, "Multiuser scheduling and power sharing for CDMA packet data systems," in *Proc. of NCC '07*, IIT-Kanpur, pp. 301-305, Jan 2007.

[6] S. Boyd, L. Vandenberghe,*Convex Optimization*. Cambridge University Press, 2004.

[7] Z-Q. Luo, W. Yu, "An Introduction to Convex Optimization for Communications and Signal Processing," *IEEE journal on selected areas in communications*, vol. 24, no. 8, pp. 1426-1428, Aug 2006.