Channel Adaptive Fair Queueing for Scheduling Integrated Voice and Data Services in Multicode CDMA Systems

Li Wang, Yu-Kwong Kwok, Wing-Cheong Lau, and Vincent K. N. Lau
Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong

***Abstract*— CDMA (Code Division Multiple Access) systems are critical building blocks of future high performance wireless and mobile computing systems. While CDMA systems are very mature for voice services, their potentials in delivering high quality data services (e.g., multimedia messaging services) are yet to be investigated. One of the most crucial component in an advanced wideband CDMA system is the judicious allocation of bandwidth resources to both voice and high data rate services so as to maximize utilization while satisfying the respective quality of service requirements. Specifically, in a multicode CDMA system, the problem is to intelligently allocate codes to the users' requests. While previous work in the literature has addressed this problem from a capacity point of view, the fairness aspect, which is also important from the users' point of view, is largely ignored. In this paper, we propose a new code allocation approach that is channel adaptive and can guarantee fairness with respect to the users' channel conditions. Simulation results show that our approach is more effective than the proportional fair approach.**

Keywords: wireless, multicode, CDMA, scheduling, rate allocation, fairness, channel adaptive.

I. Introduction

CDMA (Code Division Multiple Access) systems are the major infrastructures for the next wave of innovative wireless information applications. CDMA techniques have already proven themselves to be highly effective (both in cost and service quality) for voice and short messaging applications. Multimedia messaging Services (MMS), which require a much higher data rate (e.g., at least 5 times the basic rate used by a voice channel), are widely envisioned to be the next "killer applications" in the wireless world. However, the question of how to efficiently allocate the packet channels in a wideband CDMA system is still largely unanswered because of several technical difficulties. Firstly, the spreading process in the physical layer limits the permissible data rates in limited wireless spectrum. Nevertheless, from an information theoretic point of view, the efficiency of utilizing the allocated spectrum could be increased in order to support packet data services and, in fact, this is the major motivation behind the wideband CDMA systems. Secondly, law of large number does not hold for the relatively small number of packet data users. Thus, the intrinsic advantage of perfect statistical multiplexing in CDMA systems does not apply to high speed packet data users. In other words, packet data transmissions from data users have to be *coordinated* carefully and to achieve this goal we need to devise an intelligent rate allocation scheme that works under realistic constraints such as considering both the downlink (from the base-station to the mobile device) and uplink (from the mobile device to the base-station), channel adaptation, as well as the soft handoff effects.

Liu et al. [9] proposed a unified bandwidth-on-demand fair-sharing platform together with the Maximum Capacity Power Allocation (MCPA) criterion. The principle of MCPA is that the maximum capacity power allocation is realized when the assigned power to each code among multi-rate mobile terminals is minimized at the same time of fulfilling the target signal-to-interference ratio (SIR) at the base station. From the results of the maximum capacity power allocation, the system capacity (the number of basic rate mobile terminals that the system can accommodate) of a multicode CDMA system is defined to be:

$$S = \frac{\gamma_0 + 1.5G}{\gamma_0} \tag{1}$$

where γ_0 is the target SIR at the base station; G is the spread-spectrum processing gain. If a mobile terminal is assigned to transmit at a higher rate (e.g., m times the basic rate), it will occupy the following amount of the system capacity:

$$S_m = \frac{m \times (\gamma_0 + 1.5G)}{m \times \gamma_0 + 1.5G} \tag{2}$$

Thus, the remaining system capacity S' after this mobile terminal is scheduled is given by:

$$S' = S - S_m \tag{3}$$

After this mobile terminal gets its packet transmitted, it will return this part of system capacity back. Most importantly, for a system with remaining capacity S' after some of the capacity has been assigned, the maximum multi-rate $m_{\max}$ which a mobile terminal can be permitted to transmit at is given by:

$$m_{\max} = \left\lfloor \frac{1.5G\frac{(S'-\mathcal{D})}{\gamma_0}}{\left[\frac{\gamma_0+1.5G}{\gamma_0} - S' + \mathcal{D}\right]} \right\rfloor \tag{4}$$

where $\mathcal{D}$ is the system capacity reservation factor.

While the framework introduced by Liu et al. [9] is rigorously formulated and has stimulated a number of important research work [3], [8]. Their algorithm (and subsequent ones) fails to make use of the channel state information to make more accurate scheduling decisions. Furthermore, fairness is largely ignored in that only round robin sharing is implemented in the code distribution (i.e., bandwidth allocation) algorithm. In this paper, we propose a channel adaptive fair queueing approach which can be incorporated in Liu et al.'s framework for more efficient allocation of bandwidth resources in a fair manner. In Section II, we provide a detailed description of our Channel Adaptive Fair Queueing (CAFQ) algorithm. We then present the performance results of the CAFQ algorithm for three different scenarios: pure voice, pure data, and integrated voice and data systems. We also compare the performance of CAFQ and proportional fairness (PF) [6], which is widely considered to be a good metric and has been implemented in the CDMA HDR services (for the downlink) [2], [5]. The last section concludes the paper.

II. Channel-Adaptive Fair Queueing

We propose a new notion of fairness to be maintained in the short term, called *channel-adaptive fairness* (CAF). Specifically, a scheduler is channel-adaptive fair if in the short term the difference between the normalized throughput (normalized with respect to the channel capacity) of any two backlogged sessions i and j is bounded as follows:

$$\left| \frac{T_i(t_1,t_2)}{r_i f(\Phi_i)} - \frac{T_j(t_1,t_2)}{r_j f(\Phi_j)} \right| < \epsilon \qquad (5)$$

where Φ_i denotes the channel state (e.g., one of the five classes A, B, C, D, and E), and $f(\Phi_i) = M(\Phi_i)^\eta$ in which $M(\Phi_i)$ is the *effective throughput factor* ($0 \le M(\Phi_i) \le 1$). The effective throughput factor is channel state dependent: $M(\Phi_i) = 0.75$ if Φ_i is channel state B, and so on. Note that the values of $M(\Phi_i)$ are determined by a channel adaptive physical layer which consists a variable throughput channel modulator of a variable throughput channel coder (i.e., one which uses different levels of FEC protection—more FEC for a poorer channel condition) and of a variable throughput channel modulator (i.e., one which uses BPSK, QPSK, 16QAM, etc. for different channel conditions). In our study, we use the ABICM (Adaptive Bit-by-Bit Interleaved Coding and Modulation) method suggested in [7]. It should be noted that using ABICM is just for illustration only and other schemes (such as [10]) can also be used with our proposed CAFQ approach. Here, η is the *punish factor*, which can help to decide between to make use of the bandwidth more efficiently and to treat every session more fairly. When a larger value of punish factor is used, we punish the non-perfect channel state session that transmit packets more seriously, and prevent it from wasting too much bandwidth. In effect, the bandwidth is used more efficiently, and the average delay of the total system is decreased and the throughput is increased. But if there is a session that is more unlucky than the others and have a higher probability of having a bad channel state, its average delay and throughput may be very bad, because it is punished seriously and prevented from occupying the bandwidth. When a smaller punish factor is used, this kind of unlucky sessions will be punished only moderately, so the average delay and throughput of these sessions are improved. But as they have more chance to access the bandwidth and hence incur a larger wastage of bandwidth, the total throughput and average delay of the system will be adversely affected. Thus, there is a trade-off and the punish factor can be used to tune the utilization of system resources.

As in existing algorithms, we associate the scheduling system with an error-free system to account for the service lost or gained by a session due to errors. A session is classified as leading or non-leading depending on the difference of the service it received between the error-free system and the real one. A session is leading if it has received more service in the real system than in the error-free one, while it is non-leading if it has received less or the same amount.

We simulate SFQ (Start-Time Fair Queueing) [4] in the error-free system in our study for the reason of simplicity because it is hard to schedule according to the finish times of the packet in the wireless environment. In the SFQ, when packet k of session i arrives, it is stamped with a virtual start time $S(P_{i_k})$, computed as:

$$S(P_{i_k}) \leftarrow \max\{V((A(P_{i_k})), F(P_{i_{k-1}}))\} \qquad (6)$$

$$F(P_{i_k}) \leftarrow S(P_{i_k}) + \frac{l_{i_k}}{r_i} \qquad (7)$$

where P_{i_k} is the k-th packet of session i, $F(P_{i_k})$ is the virtual finish time of packet P_{i_k}, $V(A(P_{i_k}))$ is the virtual clock of the system at the arrival time $A(P_{i_k})$ of the packet, r_i is the pre-allocated service share of session i, and l_{i_k} is the length of the packet. The virtual time of the packets are initialized to zero. In the error-free system, a session i is selected in the increasing order of the sessions virtual starting times among sessions that are backlogged. Since it is possible that the packet of another session instead of session i will be transmitted in the real system, a session's virtual time only keep track of the normalized service received by the session in the error-free system.

Another parameter, Δ, is used to keep track of the difference of the service a session received in the real system and in the error-free one. The Δ of a session is initialized to zero. A session is non-leading if Δ is greater than or equal to zero, while it is leading if Δ is less than zero.

In CAFQ, fairness is maintained in two aspects: in the short term, CAF is maintained among the leading sessions and non-leading sessions separately unless the sessions have the worst channel state (cannot transmit). In the long term, outcome fair is ensured with the help of a virtual compensation session.

We introduce two parameters N and L to implement the channel-adaptive fairness in the short term. N_i keeps track of the normalized amount of services received by session i which is proportional to its channel state function when it is non-leading. When a session i becomes both non-leading and not suffering from the worst channel state, N_i will get initialized as follows:

$$\max\{N_i, \min k \in \Psi\{N_k|\text{lag}_k \geq 0\}\} \tag{8}$$

where Ψ denotes the set of sessions that are backlogged and for a non-leading session chosen to transmit packets in the real system, the N_i is updated as follows:

$$N_i \leftarrow N_i + \frac{l_i}{r_i f(\Phi_i)} \tag{9}$$

and L_i is defined similarly. Here, L_i keeps track of the normalized amount of services received by session i which is proportional to its channel state function when it is leading. When a session i becomes both leading and not suffering from the worst channel state, L_i will get initialized in a way analogous to (8).

In the real system, selection is made among the non-leading ones first. The session with the minimum N_i will be selected, and the packet at the head of the waiting queue of this session will be transmitted and N_i will be updated accordingly. If there is no such kind of session which is non-leading and backlogged, the system will select from the leading ones in the increasing order of the sessions' L_i, and then L_i will be updated accordingly. If all sessions are not backlogged (a very unlikely situation in a mobile computing system with a reasonable number of active users), dummy packets will be sent. If the session j selected in the real system is not the one chosen in the error-free one and it is that is selected in the error-free system, the Δ of i and j will both be updated: $\Delta_i \leftarrow \Delta_i + l_i$, $\Delta_j \leftarrow \Delta_j - l_j$; otherwise, the Δ will not be changed. When a session with a comparatively bad channel state transmits packet, the N_i or L_i will increase more rapidly than a session with a better channel state. As the punish factor changes, we can decide how serious we should punish a session which does not have a perfect channel and transmits packets. The larger the punish factor is, the more serious we punish the unlucky sessions.

Nonetheless, there is still one issue to be considered: although the sessions, which do not have perfect channel states but get packets transmitted, are punished, they are given some chance to transmit, and part of the bandwidth of the system is wasted and can never get compensated. Because the scheduler will not schedule a leading session to transmit if there is a lagging one which is backlogged and is not in the worst channel state (i.e., state E), the scheduler will not save the effort of the system as most of the other scheduling algorithms do. So, we assign a service share to a virtual compensation session to help in the long term. This pre-allocated service share is used to help the lagging ones with perfect channel state, because only when a session has a perfect channel state, it can get compensation most efficiently. When a lagging session exits from non-perfect channel states, its session ID will be queued in the virtual compensation session. Sessions that are queued in the virtual compensation session are in the decreasing order of their Δ. So we give bonus service to the lagging sessions if it has perfect channel state, and the session which lags most will get it first so that it can be helped to catch up, and thus, long term outcome fair can be maintained.

In the error-free system, we select a session i among all the backlogged sessions and the virtual compensation session in the increasing order of the virtual time. If it is the virtual compensation session that is selected and there is session ID waiting in the queue, the session with the ID at the head of the virtual compensation queue will be scheduled to transmit in the real system. The Δ of this session will be decreased as $\Delta_i \leftarrow \Delta_i - l_i$. If it is not the virtual compensation session that is selected or there is no session ID waiting in the queue, the system will select a session to transmit in the real system from the non-leading ones according to N_i, then from the leading ones according to L_i if there is no non-leading one to take the service as we have mentioned above.

In CDMA networks, multiple mobile terminals can transmit at the same time. Using a multicode approach, multiple packets from a certain mobile terminal can transmit at the same time. However, as governed by the power and interference budgets, the number of simultaneous transmissions of packets is limited. We use the multicode CDMA framework proposed by Liu et al. [9] (i.e., equations (17) to (22) in [9]) and the method of calculating the optimal power allocation given the code/rate allocation.

In applying the proposed CAFQ algorithm in a multicode TD-CDMA systems, we refer to resource as the number of codes that the base station can support or the number of channels that the base station can transmit packets at the same time given the target SINR value. Assuming perfect power control, the resource of the cell can be calculated as given by the following equation [9]:

$$N = \frac{G}{\gamma_b} + 1 \tag{10}$$

Packets from the same mobile terminal can be delivered simultaneously to reduce the interference level of the system, but the number of them is limited by the BER requirement of their traffic types. For a certain mobile terminal transmitting packets of service type T, the allowed number of code channels is bounded by $\mathcal{M}_T$. For voice service mobile terminals, only 1 packet need to be transmitted at a time. However, for data service mobile terminals, several packets can be transmitted together, depending upon the channel condition and the interference/power budgets.

At the beginning of a frame, the system resource is initialized as (10), so there is full resource available to schedule. Then the flow with the highest priority according to CAFQ algorithm is selected to receive service, and the number of codes which is allocated to this flow is determined using equation (4). After the scheduling, the remaining resource is updated accordingly. When a flow is selected, the system will compare the effective service share $\hat{r_i}$ of this flow and the remaining resource C. The smaller one is the number of codes assigned to it: $C_i \leftarrow \min(\hat{r_i}, C)$. The lagging, non-lagging definitions, the updating of N_i, L_i, Δ and the virtual compensation flow are manipulated as described earlier. For a lagging flow chosen to transmit packets in the real system, N_i is updated as follows:

$$N_i \leftarrow N_i + \frac{C_i l_i}{r_i f(\Phi_i)} \tag{11}$$

where C_i is the number of codes assigned to flow i. When a flow i becomes both non-lagging and not suffering from the worst channel state, L_i gets initialized in a way analogous to N_i.

When there is remaining system resource, the algorithm works as follows. When it is a normal flow that is selected in the reference system (round robin runs in the reference system when $\mathcal{M}_T \neq 1$, because the system allocates the codes according to service shares in the real system), a backlogged lagging flow with the smallest N_i will be selected in the real system to get service. It will allocate C_i codes to mobile terminal i, and after this, the resource is decreased by C_i. If there is no such kind of lagging flow available, the service chance goes to the non-lagging flow with the least L_i. If all flows are not backlogged, scheduling will be done in the next frame. If the flow i selected in the real system is not the flow j which is selected in the error-free one, the Δ of i and j will both be updated: $\Delta_i \leftarrow \Delta_i - C_i \times l_i$, $\Delta_j \leftarrow \Delta_j + r_j \times l_j$; otherwise, the Δ values will not be changed.

When it is the virtual compensation flow that is selected in the reference system and there is a flow ID waiting in the queue, the flow with the ID at the head of the virtual compensation queue will be scheduled to transmit in the real system. The Δ of this flow i will be decreased as $\Delta_i \leftarrow \Delta_i - C_i \times l_i$. Flow i must be the one which is currently having a perfect channel state and lags most seriously.

III. Performance Results

We simulated the performance of CAFQ and PF with the realistic channel model as described earlier (i.e., an accurate channel model and the ABICM scheme are used in a completely integrated simulation environment in that the physical layer is combined with the MAC and scheduling layers). The parameters used in the simulation scenarios are listed in Table I (the parameters we used are based on those presented in [1]). We run each test case for 1000 seconds of simulated time and average the results of 10 runs to obtain the performance data. The virtual compensation flow has the same service share as the other flows.

TABLE I
Simulation parameters.

Parameter	Value
Path loss exponent	4
log-normal shadowing variance	8 dB
Channel bandwidth	5 MHz
Cell radius	800 m
Chip rate	7.7328 Mcps
Average adjacent cell load	75%
Modulation	QPSK modulation with quadrature spreading
$\mathcal{M}_T$ (voice)	1
$\mathcal{M}_T$ (data)	4
processing gain	30.3
frame length	16 msec
basic transmission rate	165 kbps
channel bandwidth	5 MHz
number of slots in each frame	10
window size of PF	5 sec

A. Test Scenario 1: Data Sources

Assuming perfect power control, the system can support 2 codes transmitting at the same time for data source with SINR of 17.985 dB. In fact, different kinds of power control algorithms can vary the number of packets that the base station can transmit at a time, but in our study a fixed number after the power control algorithm is selected. We simulate data sources using Poisson arrivals. All the six data sources have the same arrival rate of 60 kbps, which is 6 times higher than the basic voice rate.

First, we try to find whether both of the algorithms work fairly to the flows if they have the same channel error mode. We simulate both CAFQ and PF when all the flows have the same error mode 1, and find that CAFQ performs as well as PF. All the flows have almost the same throughput when both of the algorithms run as shown in Figure 1.

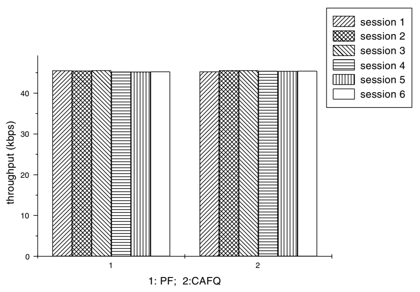

Fig. 1. Both of the algorithms treat the flows with the same channel state fairly.

Then, how do they perform when flows have different error modes? We consider the situation when flow 1 and 2 have error mode 1, flow 3 and 4 encounter error mode 2, and flow 5 and 6 meet error mode 3. That means flows 1 and 2 have the best overall channel state, flow 3 and 4 have poorer one, and flow 5 and 6 meet with bad channel states with the highest probability.

The throughput and the average delay of the system are shown in Figure 2 and Figure 3, respectively. For $x = 0$ on the x-axis, the value on the axis shows the throughput and the average delay when PF works respectively. For $x = 1, 2, 3, 4, 5$ on the x-axis, the value on the y-axis shows the throughput and average delay when CAFQ works and is the value of the punish factor. These two figures show that CAFQ outperforms PF, and as the punish factor increases, the throughput increases and the average delay decreases. This is because CAFQ tries to use the resource more efficiently at the expense of short term fairness with a larger punish factor. With a big punish factor, the system will punish the flows that have bad channel states and use the resource. Flows 5 and 6 have the highest probability to have bad channel states, so they have least chance to access the resource. Thus, the resource is used more efficiently, and the performance is improved.

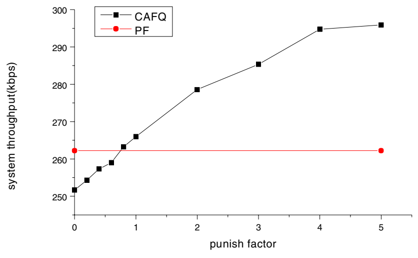

Fig. 2. The system throughput of the data users.

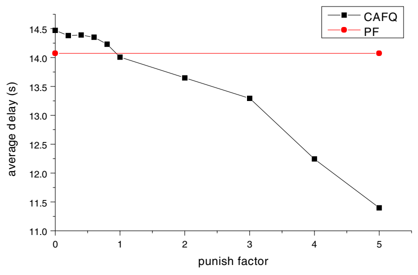

Fig. 3. The average delay of the data users.

However, the improvement of the performance is at the expense of the fairness when the punish factor increases. In the short term, CAF is maintained, but it is more and more unfair in the sense of outcome fairness with a larger punish factor. With a larger punish factor, the channel utilization efficiency is improved, so the performance is better. However, it tends to treat flows with various channel states differently. Thus, the flows with better channels have better performance than those with poor channels. These are indicated in Figure 4 and Figure 5.

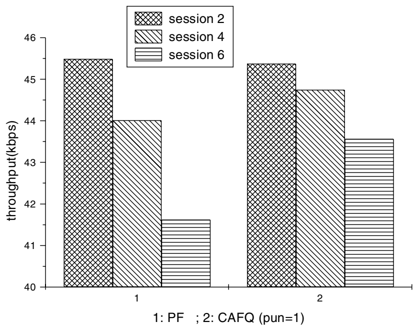

Fig. 4. Throughput comparison.

Figure 4 shows the throughput of flows 2, 4, and 6 of both algorithms when the punish factor is 1 in CAFQ. We do not show the throughput of the other flows because we have shown in Figure 1 that both of the algorithms treat the flows with the same channel error mode equally. CAFQ has better overall performance than PF, and it still ensures better outcome

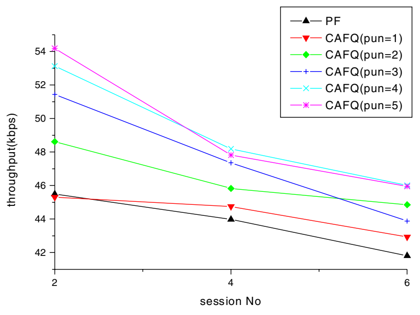

Fig. 5. The fairness property changes with the punish factor.

fairness property than PF. With the same system resource, CAFQ can have both better performance and better outcome fairness property than PF, this shows CAFQ makes use of the resource more intelligently.

Figure 5 also serves to show the fairness properties. The line when the punish factor is equal to 1 is flatter than that of PF. But as the punish factor increases, the lines are less and less flat, which means the long term outcome fair property is poorer and poorer as the punish factor increases. As we mentioned, the desirable fairness notion should have a balanced consideration of fairness and channel utilization efficiency. The punish factor in CAFQ can adjust the weights of fairness and channel utilization efficiency in the trade-off.

B. Test Scenario 2: Voice Sources

We simulate 30 voice sources as constant bit rate of 32 kbps, and a voice packet will be thrown away if it has not be delivered 2 frames after it arrives. For voice source with SINR of 4.75, the system can support 7 codes at the same time. Figure 6 and Figure 7 show that CAFQ outperforms PF again in both throughput and packet loss rate.

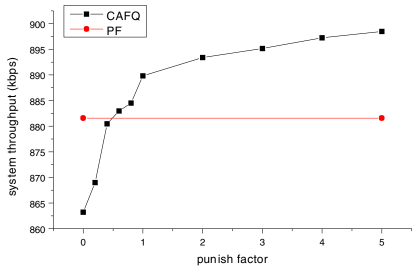

Fig. 6. The system throughput of the voice users.

C. Test Scenario 3: Integrated Voice and Data Services

To investigate the performance of the CAFQ algorithm for integrated voice and data services, we also performed experiments with varying numbers of voice and data users that are active simultaneously in the system. The results are shown in Figures 8 to 11. We can see that as in the cases of homogeneous systems (i.e., voice users only and data users only), the CAFQ algorithm consistently outperforms the PF approach.

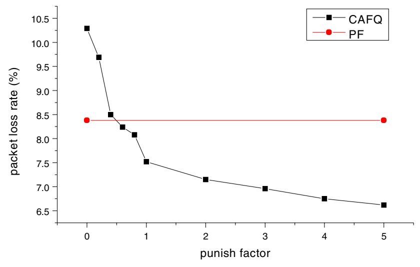

Fig. 7. The packet loss rate of the voice users.

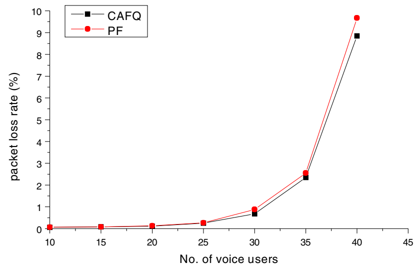

Fig. 8. The packet loss rate of voice users when there are 5 data users.

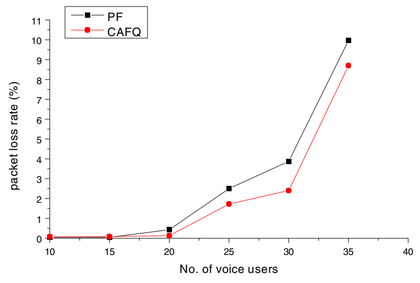

Fig. 9. The packet loss rate of voice users when there are 10 data users.

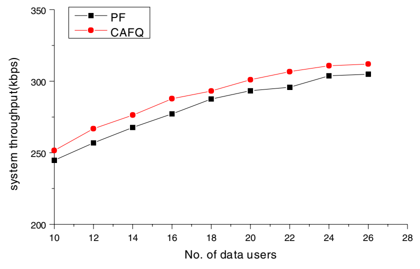

Fig. 10. The system throughput for various number of data users when there are 5 voice users.

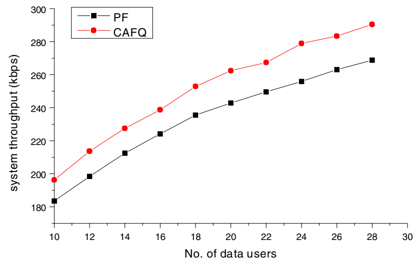

Fig. 11. The system throughput for various number of data users when there are 10 voice users.

IV. Conclusions

In this paper, we describe the CAFQ (Channel Adaptive Fair Queueing) algorithm that can be applied in a multicode TD-CDMA system. To schedule the transmissions in each frame, the CAFQ algorithm keeps on allocating codes until the whole bandwidth resource is used up (subject to the interference budget as controlled by the power allocation). The CAFQ algorithm selects flows and allocates codes to them in a way that short-term fairness is guaranteed and long-term fairness is heuristically achieved. The simulation results show that CAFQ outperforms the proportional fair algorithm in terms of throughput, average delay, and fairness property.

Acknowledgments

This research was supported by the Hong Kong Research Grants Council under project number HKU 7024/00E.

References

[1] I. F. Akyildiz, D. A. Levine, and I. Joe, "A Slotted CDMA Protocol with BER Scheduling for Wireless Multimedia Networks," *IEEE/ACM Transactions on Networking*, vol. 7, no. 2, pp. 146–158, Feb. 1999.

[2] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: A Bandwidth Efficient High-Speed Wireless Data Service for Nomadic Users," *IEEE Communications Magazine*, July 2000, pp. 70–77.

[3] S. Choi and K. G. Shin, "An Uplink CDMA System Architecture with Diverse QoS Guarantees for Heterogeneous Traffic," *IEEE/ACM Transactions on Networking*, vol. 7, no. 5, pp. 616–628, Oct. 1999.

[4] P. Goyal, H. M. Vin, and H. Chen, "Start-time Fair Queueing: A Scheduling Algorithm for Integrated Services," *Proc. ACM SIGCOMM'96*, pp. 157–168, Aug. 1996.

[5] A. Jalali, R. Padovani, and R. Pankaj, "Data Throughput of CDMA-HDR: A High Efficiency High Data Rate Personal Communication Wireless System," *Proc. VTC'2000*.

[6] F. Kelly, "Charging and Rate Control for Elastic Traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[7] V. K. N. Lau, "Performance of Variable Rate Bit-Interleaved Coding for High Bandwidth Efficiency," *Proc. of VTC'2000*, vol. 3, pp. 2054–2058, May 2000.

[8] S. J. Lee, H. W. Lee, and D. K. Sung, "Capacities of Single-Code and Multicode DS-CDMA Systems Accommodating Multiclass Services," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 2, pp. 376–384, Mar. 1999.

[9] Z. Liu, M. J. Karol, M. El Zarki, and K. Y. Eng, "Channel Access and Interference Issues in Multicode DS-CDMA Wireless Packet (ATM) Networks," *Wireless Networks*, vol. 2, pp. 173–193, 1996.

[10] M. Zorzi and R. R. Rao, "The Role of Error Correlations in the Design of Protocols for Packet Switched Services," *Proceedings of the 35th Annual Allerton Conference on Communications, Control, and Computing*, pp. 749–758, Sept. 1997.