# CHAPTER 5

# ALGORITHMS FOR MULTI-TASKING SERVICES IN MULTIUSER OFDM SYSTEMS

In order to cater the increasing demands for high capacity and multi-tasking services, wireless communication systems are required to support multiple users each of whom would have multiple, heterogeneous and simultaneous traffic queues. One such application is a mobile device, used to make a phone call, watch the Internet TV, and download files from a website at the same time.

The network capacity region is defined as a set of all packet arrival rate vectors for which it is possible to keep every queue length finite. For bursty input traffic, it is generally difficult to estimate the packet-arrival rates. Thus, resource allocation solely based on CSI is unable to update rate allocation properly according to the dynamics of the input traffic. As a result, even for a packet arrival rate vector within the network capacity region, some users' queue backlogs may become unacceptably large, causing long queuing delay as well as buffer overflow. Certain schedulers assume infinite queue backlog and simply consider CSI only. On the other hand in the packet switching systems, the channel can be described by the circuit connection, so the concept of CSI is irrelevant. The above two schedulers are called `channel-aware scheduler' and `queue-aware-scheduler', respectively, to distinguish them from the queue-channel-aware scheduler. The network capacity region may not be achievable with channel-aware scheduler that ignores queuing dynamics. In addition, each user's queuing delay, which is an important QoS parameter, is uncontrollable without considering queue sizes in scheduling. Therefore, it is essential to design an intelligent scheduler that considers both CSI and QSI.

In this chapter, utility functions with respect to average delays is used for designing both channel and queue aware scheduling, which is highly advantageous to data transmission with a low latency requirement. Orthogonal Frequency Division Multiplexing (OFDM) provides fine granularity for resource allocation since they are capable of dynamically assigning sub-carriers to multiple users and adaptively allocating transmit power.

The background and motivations of this chapter are introduced in section 5.1. In section 5.2, extending scheduling rules existing in single-carrier systems into OFDM multicarrier scheduling algorithms is discussed. In section 5.3, two types of algorithms under Cross-layer scheduling viz., QoS Proportional Fair (QPF) scheduling and Adaptive Cross-layer Packet scheduling (ACP) are explained and the comparison is performed for several multicarrier scheduling algorithms using simulation. The simulation results are summarized in the last section.

## 5.1 INTRODUCTION

The design of an efficient packet scheduler for a wireless system is a difficult task that typically involves a large number of conflicting requirements, which must be analysed and weighted before a balanced solution can be implemented. On one hand, a scheduler must be efficient in utilizing the radio resources since the wireless spectrum is the most precious resource in wireless communication systems. On the other hand, the services should be fairly scheduled in order to guarantee a certain level of service to users with low average channel conditions. Among the fairness criteria, the

Proportional Fair (PF) scheduling is widely considered as a good solution because it provides an attractive trade-off between the maximum average throughput and the user fairness. Unfortunately, there are some problems that have not been well tackled if we want to adopt PF in OFDM systems.

For instance, the PF scheduling method in Jalali et al (2001) can only be used in a single carrier (SC) situation, and only one user is allowed to transmit at a time. Yoon et al (2004) proposed the PF scheduler for the OFDM systems based on the PF criterion derived from the SC system without considering the PF criteria for a multicarrier (MC) system. Kim and Han (2005) studied the optimal criteria for PF scheduling in MC systems. It extends the PF scheduling proposed for the high data rate (HDR) system to multicarrier transmission systems. Generally, the PF allocation results in the maximization of the sum of logarithmic average user rates. They proposed a PF scheduling that assigns users to each carrier while maximizing the sum of logarithmic average user rates. But its practical implementation is very complex. Thus there is a pressing need in developing an alternative PF scheduling algorithm for practical implementation in multiuser OFDM systems. A typical PF scheduler only considers the performances of average system throughput and fairness. Such a PF scheduler usually does not consider other QoS performance metrics, such as Packet Dropping Probability (PDP) and packet delay. Thus, in this context, a PF scheduling is needed for improving the packet level QoS.

Song (2005a) investigated multiuser downlink data scheduling with QoS provisioning over multiple shared fading channels, which for a network point of view, provides fine flexibility and granularity for resource allocation. They employed a user centric metric, a utility function with respect to mean waiting time, which is able to maintain fairness among users while providing delay QoS to individual users. But these research efforts are based on the assumption of infinite queue length without considering realistic queuing conditions that make this algorithm as impractical in real life wireless packet transmission system. Hence, in order to meet the packet level QoS performance specifications, the data link layer queuing performance should also be analyzed, in addition to the channel conditions, to allocate the radio resources in a cross-layer manner.

Hossain et al (2005) explored optimal and suboptimal power and bit loading algorithms for a multicarrier system. Specifically they studied the trade-offs between the total transmit power of an OFDM system and the buffering delay of the packets in a transmission buffer. The loading framework is formulated as a Markov decision process and an optimal loading policy which minimizes the transmit power while meeting a target delay constraint is obtained via equivalent linear programming methodology. Huang and Niu (2007) proposed a practical packet scheduling algorithm for wireless packet switched OFDM systems called Buffer-Aware and Traffic-Dependent (BATD) scheduler consisting of two parts: PHY-layer AMC technology on each subcarrier and MAC-layer opportunistic scheduling. They scheduled only one packet from one user at any time slot.

In this chapter, the joint channel- and queue-aware scheduling algorithms in OFDM based multicarrier networks are designed. The scheduling design for multicarrier networks is not just a simple extension of existing scheduling approaches in single-carrier networks. First, multicarrier networks have nice granularity for resource allocation, since the whole bandwidth is divided into many sub channels. Second, multicarrier scheduling actually works in a parallel fashion. Unlike in single-carrier networks, multiple users can be served simultaneously in multicarrier networks; thus, from a queuing point of view, there are multiple servers in multicarrier scheduling.

In this scheduling, several packets from different users can simultaneously be scheduled so as to exploit the multiuser diversity. In this part of the thesis, an efficient cross-layer approach to allocate the subcarrier and schedule the packets in the downlink of multiuser OFDM systems is focused. Specifically it is considered to achieve PF while improving the individual user's packet level QoS performance.

For supporting multi-tasking services, the existing QPF algorithm is not sufficient to accommodate multiple users with multiple heterogeneous traffic queues simultaneously. Therefore, a novel Adaptive Cross-layer Packet scheduling (ACP) is developed to maximize the weighted sum capacity of the downlink multiuser multi-tasking OFDM systems.

## 5.2 EXTENDING SCHEDULING RULES IN SINGLE CARRIER NETWORKS INTO MULTICARRIER OFDM NETWORKS

In this section, dynamic subcarrier allocation is considered besides fixed power allocation. Some existing scheduling algorithms exploiting multiuser diversity in single carrier networks can be directly extended to multicarrier networks. In dynamically assigning subcarriers, it is required to solve the optimization problem expressed as follows:

$$\max_{D_i^{(n)}, i \in A^n} \sum_{i \in A^n} w_i[n] r_i[n] \tag{5.1}$$

subject to $\bigcup_{i \in A^n} D_i^{(n)} \subseteq k$, (5.2)

$$D_i^{(n)} \bigcap D_j^{(n)} \neq \phi, \ i \neq j \ \forall i, j \in A^n \tag{5.3}$$

where $A^n = \{i : Q_i[n] > 0\}$ is the set in which each queue is not empty at time slot $n$ and the optimization objective is to maximize the sum weighted data rate with the weights $w_1[n], w_2[n], \cdots, w_M[n]$. In Kibeom Seong (2008), the optimal assignment for dynamic subcarrier allocation problem is derived as

$$m(k,n) = \arg\max_{i \in A^n} \{w_i[n] c_i[k,n]\}, \tag{5.4}$$

where $m(k,n)\ (m(k,n) \in A^n)$ represents subcarrier $k$ to be assigned to user $m(k,n)$ at time $n$. This result is very useful to design scheduling approaches or to extend some scheduling rules in the single carrier case to the OFDM scenario.

### 5.2.1 Max-Sum-Capacity (MSC) Rule

The MSC rule is a channel-aware scheduling algorithm that maximizes the total throughput in the system. The optimization problem can be expressed by Equations (5.1) - (5.3) with $w_i[n]=1$, for all $i$. Clearly, the MSC rule is given by

$$m(k,n) = \arg\max_{i \in A^n} \{c_i[k,n]\} \tag{5.5}$$

Although the MSC rule makes the most efficient use of the bandwidth, it can lead to unfairness and instability, especially for non symmetrical channel conditions and non-uniform traffic patterns.

### 5.2.2 Proportional Fair (PF) Scheduling

The PF scheduling is a channel-aware scheduling rule aiming to maximize $\sum_i \ln\left(\bar{r}_i[n]\right)$, where $\bar{r}_i[n]$ is the average data rate for user $i$. The scheduling rule in multicarrier networks obtained in Song et al (2005a) is

$$m(k,n) = \arg\max_{i \in A^n} \left\{ \frac{c_i[k,n]}{\bar{r}_i[n]} \right\} \tag{5.6}$$

Although this DSA algorithm guarantees the proportional fairness, it is not throughput-optimal as explained by Andrews et al (2001). The PF scheduling is suitable to best-effort traffic having no specific QoS requirements.

### 5.2.3 Modified-Largest Weighted Delay First (M-LWDF) Rule

Andrews et al (2000) proposed the M-LWDF algorithm for single-carrier CDMA networks with a shared single downlink channel. From an optimization point of view, the M-LWDF intends to maximize $\sum_i \frac{T_{HOL,i}[n]}{\overline{r}_i[n]} r_i[n]$, where $T_{HOL,i}$ is the delay of the head-of-line (HOL) packet of user *i*. For multi carrier OFDM networks, the multichannel version of M-LWDF can be obtained by using Equation (5.4) as

$$m(k,n) = \arg\max_{i \in A^n} \left\{ \frac{c_i[k,n]}{\overline{r}_i[n]} T_{HOL,i}[n] \right\} \tag{5.7}$$

### 5.2.4 Exponential (EXP) Rule

Shakkottai and Stolyar (2002) designed the EXP scheduling rule for single-carrier CDMA networks with a shared downlink channel. The structure of the EXP rule is very similar to the M-LWDF, but with different weights. The multichannel version of EXP rule can be expressed as

$$m(k,n) = \arg\max_{i \in A^n} \left\{ \frac{c_i[k,n]}{\overline{r}_i[n]} \exp\left( \frac{T_{HOL,i}[n]}{1 + \sqrt{\overline{T}_{HOL}[n]}} \right) \right\} \tag{5.8}$$

where $\overline{T}_{HOL}[n] = \frac{1}{|A^n|} \sum_{i \in A^n} T_{HOL,i}[n]$.

The M-LWDF and EXP rules have been proven to be throughput-optimal in single carrier networks. With a few modifications, the rules are valid in OFDM networks too. Both scheduling rules are proposed for delay-sensitive traffic.

### 5.2.5 Max-Delay-Utility (MDU) Scheduling

Similar to M-LWDF, Song (2005) proposed a queue based MDU scheduling algorithm that maximizes the utility functions with respect to the delay. The average waiting time over the time window at time $nT_s$ is defined as

$$W_i[n] = \frac{\overline{Q_i}[n]}{\lambda_i} \tag{5.9}$$

where $Q_i$ is the average queue length and $\lambda_i$ is the average arrival rate of queue *i* calculated in current slot.

Given the arrival processes, the average waiting time is actually determined by the service rate during time slot $r_i[n]$. The optimization objective is to maximize the total utility with respect to the predicted average waiting times at each time slot in the network, i.e.,

$$\max \sum_{i=1}^{M} \frac{\left|U_i^{'}(W_i[n])\right|}{\lambda_i} r_i[n], \tag{5.10}$$

### 5.2.6 Packet Dependent Scheduling

The Packet Dependent (PD) scheduling proposed by Nan Zhou et al (2010), a joint queue and channel aware scheduling algorithm, assigns different weights to different packets contained in the same queue. The packet weights are determined based on the delay, packet size, and QoS priority level of the packets. Therefore it is more flexible than the other queue based scheduling. The PD scheduling algorithm assigns a higher weight to the packets in a queue which have a higher QoS priority level, a larger data amount and a fewer time left to become urgent. Then these packets will be scheduled first. The weight $W_k$ for the current slot is given by

$$W_k = \sum_{i=1}^{I_k} \sum_{l} W_{k,i,l}$$

$$= \sum_{i=1}^{I_k} \beta_{k,i} \left[ \sum_{l \in L_{k,i}^{U}} D_{k,i,l} + \sum_{l \in L_{k,i}^{\bar{U}}} \frac{D_{k,i,l}}{C_{k,i,l} + 1} \right] \tag{5.11}$$

Let $W_{k,i,l}$ be the weight of the packets to-be-served, $D_{k,i,l}$ be the packet size (in bits) that belong to queue $i$ of user $k$, $C_{k,i,l}$ be the time left for the packet that becomes urgent and $\beta_{k,i} \in [1, \infty)$ be the QoS priority level for queue $i$ of user $k$.

To make a good tradeoff between performance and complexity, only the first $\Lambda_{k,i}$ to-be-served packets (most urgent packets) in queue $i$ of user $k$ for weight calculation of the MWSC based cross-layer design is selected. The complexity decreases with a small-valued $\Lambda_{k,i}$, while less QoS information of the packets to-be-served is obtained from the MAC layer for resource allocation. As a result, the performance of the cross-layer design is degraded. In this case, the urgent packets represented by the first term in Equation (5.11) play a more important role in weight calculation. On the other hand, with a large valued $\Lambda_{k,i}$, the complexity increases, while more QoS information of the packets to-be-served is reflected in the weights, which leads to an improved system performance. While increasing $\Lambda_{k,i}$, the implementation of PD algorithm becomes a complex one and the performance degradation also occurs.

## 5.3 CROSS-LAYER SCHEDULING ALGORITHMS

To reduce the complexity and to improve the performance of the cross-layer scheduler, the joint channel- and queue-aware scheduling algorithms for multicarrier OFDM based networks are designed. Two algorithms under Cross-layer scheduling algorithms, namely QoS Proportional Fair (QPF) scheduling and Adaptive Cross-layer Packet (ACP) scheduling are proposed.

### 5.3.1 QoS Proportional Fair (QPF) Scheduling

The OFDM frames are transmitted through the physical layers between the base station and mobile user. The general Cross-layer Scheduling model is same as discussed in section 4.3. The data link layer maintains the queue buffers that contain the packets delivered from the higher application layer. The packet scheduler collects the cross-layer information (CSI and QSI) and selects different users' packets for transmission according to the information obtained. In this model, a total of $K$ mobile stations (MS) in the OFDM system with $N_F$ subcarriers and the total bandwidth of $W$ are assumed. In the BS, the incoming packets of each user arriving from some higher layers are buffered in its own First-In-First-out (FIFO) queue with a finite space of $B$ packets waiting to be scheduled. The packet length is assumed to be $N$ bits / packet. At the beginning of each time slot, the packet scheduler selects some packets in the queues for transmission according to CSI and QSI so as to meet the QoS requirements. The selected packets are then forwarded to the OFDM transmitter and they are adaptively modulated at the corresponding mode related to the CSI and distributed on different subcarriers. After inverse fast Fourier transform (IFFT) and guard interval (GI) insertion, the scheduled packets from the different users form an OFDM symbol and are then sent to the MSs via the down-link channels. Those $S$ numbers of OFDM symbols are grouped into one OFDM frame, and one frame transmission time is assumed to be $T_0$ seconds, which can be expressed as

$$T_0 = S \text{ (length of one OFDM symbol)}$$

$$= S\left[\frac{N_F}{W}\left(1 + 0.25\right)\right] \tag{5.12}$$

where $S\left(N_F / W\right)$ is the length of data, and $0.25\,S\left(N_F / W\right)$ is the length of GI. Thus each OFDM frame is segmented in terms of time slots and $T_0$ is the length of one time slot and it also corresponds to one scheduling time interval.

The proposed QPF algorithm consists of two steps. In the first step, the greedy PF method is used to achieve proportional fairness and in the second step, a subcarrier reassignment procedure is utilized to improve the packet-level QoS performances according to the queue analysis. The proportional fairness criterion for a single carrier system is not suitable for the multiple users OFDM systems with multiple carriers. Then based on the PF definition by Kim et al (2005) for MC systems, the PF scheduling in the OFDM systems can be formulated as

$$P = \arg\max \prod \left(1 + \frac{\sum_{m \in C_k} r_{k,m}(t)}{(t_c - 1)\overline{R_k(t)}}\right) \tag{5.13}$$

subject to

$$(I) \bigcup_{k \in U} C_k \subseteq C \qquad (II) C_i \cap C_j = \phi, \qquad i \neq j \quad \forall i, j \in U$$

where $C = \{1,2,\ldots,N_F\}$ and $U = \{1,2,\ldots,K\}$ denote the subcarrier index set and the user index set respectively. $C_k$ is the set of subcarriers that are allocated to the $k^{th}$ user. Let $\overline{R_k(t)}$ is the average date rate at time slot t of user

i. Here, the constraint (I) means that each user selects its subcarrier from $C$, and the constraint (II) shows that each subcarrier can only be assigned to one user. $\overline{R_k(t)}$ can be approximated by a moving average value with average window size $t_c$ time slots

$$\overline{R_k(t+1)} = \left(1 - \frac{1}{t_c}\right)\overline{R_k(t)} + \frac{1}{t_c} r_k(t) \tag{5.14}$$

When introducing the channel assignment status $x_{k,m}(t)$, the preceding Multi carrier PF (MCPF) problem can be reformulated by

$$\max_{x} \prod_{k=1}^{K} \left( 1 + \frac{\sum_{m=1}^{M} x_{k,m}(t) r_{k,m}(t)}{(t_c - 1)\overline{R_k(t)}} \right) \tag{5.15}$$

subject to $\sum_{k=1}^{K} x_{k,m}(t) = 1,\ \ x_{k,m} = \{0, 1\}$

where $x = [x_{1,1}, x_{1,2}, \cdots, x_{K,M}]^T$. The optimization in Equation (5.15) is a nonlinear integer programming problem and the algorithm complexity for the optimal solution is prohibitively high. Specifically, for a system with $K$ users and $N_F$ subcarriers, in the worst case, a total number of ${N_F}^K$ times of iterations are needed to find the solution. Such an algorithm will consume much time and memory space, which is not suitable for practical systems. To make the scheduling method amenable for practical implementation, a simplification for the foregoing generic MCPF scheduler is needed. Thus a fast and efficient greedy method to solve this problem is presented.

The objective function in Equation (5.15) is a multiple-product function. To simplify the algorithm implementation, the product in Equation (5.15) is converted into a summation by taking a logarithm function on the objective. Then, the equivalent problem can be written as

$$\max_{x} \prod_{k=1}^{K} \log\left(1+\frac{\sum_{m=1}^{N_F} x_{k,m}(t) r_{k,m}(t)}{(t_c-1)\overline{R_k(t)}}\right) \tag{5.16}$$

subject to $\sum_{k=1}^{K} x_{k,m}(t)=1,\ \ x_{k,m}=\{0,1\}.$

To be specific, the system PF value PF($t$) at time slot $t$ is defined as

$$PF(t)=\sum_{k=1}^{K}\log\left(1+\frac{\sum_{m=1}^{N_F} x_{k,m}(t)\ r_{k,m}(t)}{(t_c-1)\ \overline{R_k(t)}}\right)$$

$$=\sum_{k=1}^{K}\log\left(1+\frac{r_k(t)}{(t_c-1)\overline{R_k(t)}}\right) \tag{5.17}$$

Notice that if one subcarrier is assigned to a different user, then the resulting PF value will be different. For example, if before the assignment of the $m^{th}$ subcarrier the user rate for every user $k \in U$ is $r_k(t)$, then when this subcarrier is allocated to the $k'^{th}$ user, the new PF value is given by

$$PF(t,k')=\log\left(1+\frac{r_k(t)+r_{k',m}(t)}{(t_c-1)\overline{R_{k'}(t)}}\right)+\sum_{k\neq k'}\log\left(1+\frac{r_k(t)}{(t_c-1)\overline{R_k(t)}}\right) \tag{5.18}$$

Thus for the $m^{th}$ subcarrier, if the largest system PF value PF($t$,$k$) is obtained when the $k^{th}$ user gets it, then it should be assigned to this $k^{th}$ user. With this strategy, the subcarrier can be allocated so as to get the highest PF value. Consequently, all of the subcarriers can be assigned one by one in this greedy manner and the resulting PF value will be the maximal value when all the subcarriers are allocated. Moreover, since the scheduler needs to compare $K$ users to allocate a subcarrier, the total computational complexity is $K\ N_F$, and this is efficient compared to the number of comparisons $N_F{}^K$ for the original problem in Equation (5.16).

#### 5.3.1.1 PF Scheduling with Packet-Level QoS Improvement

The PF scheduler presented earlier did not consider the queue state coupled with the assumption of infinite incoming packets. When considering the queue state and bursty traffic, the scheduler should not serve any empty queue as it will waste radio resources. In addition, if a packet delay exceeds its delay limit, then this packet should be dropped from its queue. The packet-level QoS performance metrics such as packet delay and PDP should also be considered.

To analyse the packet-level QoS performance, a queuing service is modelled. Assuming at the beginning of time slot *t* the queue length is $Q_k(t)$, the scheduler serves the $k^{th}$ user at rate $r_k(t)$ according to CSI and QSI. Furthermore, there are $\pi_k(t)$ packets to be dropped because their packet delays exceed the limit. Then, if there are $\vartheta_k(t)$ packets arriving during this time slot and regardless of the queue limit, the queue length at the end of this time slot can be expressed as

$$U_k(t+1) = Q_k(t) - \frac{r_k(t)T_0}{N}\vartheta_k(t) - \pi_k(t) \qquad (5.19)$$

Since the queue has a capacity limit of *B* packets, the actual queue length $Q_k(t+1)$ at the beginning of time slot *t+1* can be verified by

$$\begin{aligned} Q_k(t+1) &= \min\{B, U_k(t+1)\} \\ &= \min\left\{B, Q_k(t) - \frac{r_k(t)T_0}{N} + \vartheta_k(t) - \pi_k(t)\right\} \qquad (5.20) \end{aligned}$$

The number of dropped packets due to the overflow at the end of time slot $t$ can then be evaluated as

$$D_k(t+1)=\max\{0, U_k(t+1)-B\}$$
$$=\max\left\{0, Q_k(t)-\frac{r_k(t)T_0}{N}+\vartheta_k(t)-\pi_k(t)-B\right\} \tag{5.21}$$

And, to avoid serving the empty queues, the scheduler should control the service rate so that

$$r_k(t)\leq\frac{Q_k(t)N}{T_0} \tag{5.22}$$

The average packet delay and the average PDP can now be analysed based on this model. Similar to the average data rate definition in Equation (5.14), let the average queue length $\overline{U_k(t)}$ over the average window size $t_c$, regardless of the queue limit be

$$\overline{U_k(t+1)}=\left(1-\frac{1}{t_c}\right)\overline{U_k(t)}+\frac{1}{t_c}U_k(t+1) \tag{5.23}$$

Subsequently, at the beginning of time slot $t$, for the given $\overline{U_k(t)}$, the actual queue length $Q_k(t)$, the dropped packets $\pi_k(t)$ due to deadline missed, and the average packet incoming rate $E\{\vartheta_k(t)\}$, the predicted average queue length over the average window size at the end of time slot $t$ can be expressed as

$$\hat{U}_k(t+1)=E_{\vartheta_k(t)}\left\{\overline{U_k(t+1)}\right\}$$
$$=\left(1-\frac{1}{t_c}\right)\overline{U_k(t)}+\frac{Q_k(t)-\frac{r_k(t)T_0}{N}+E\{\vartheta_k(t)\}-\pi_k(t)}{t_c} \tag{5.24}$$
$$=H(r_k(t))$$

where $E_{\vartheta_k(t)}\{\cdot\}$ is the expectation for the expression {.} with respect to $\vartheta_k(t)$, which can be obtained according to the incoming traffic characteristics. Now, if the $k^{th}$ user is scheduled with the data rate $r_k(t)$, then the predicted average number of dropped packets at the end of time slot *t* due to overflow can be expressed as

$$\hat{D}_k(t+1) = \max\{0, \hat{U}_k(t+1) - B\}$$
$$= \max\{0, H(r_k(t)) - B\} = F(r_k(t)) \quad (5.25)$$

Letting $\mu_k(t+1)$ be the estimate of the $k^{th}$ user's average PDP at the end of time slot *t*, then

$$\mu_k(t+1) = \frac{\hat{D}_k(t+1) + \pi_k(t)}{E\{\vartheta_k(t)\}} \quad (5.26)$$

Thus, at the beginning of time slot *t*, the required data rate $r_k(t)$ is obtained if the PDP requirement $\mu_k$ is met when $\hat{U}_k(t), \pi_k(t), Q_k(t)$ and queue size *B* are given. Since

$$\mu_k(t+1) = \frac{\hat{D}_k(t+1) + \pi_k(t)}{E\{\vartheta_k(t)\}} \leq \mu_k, \quad (5.27)$$

$$F(r_k(t)) \leq E\{\vartheta_k(t)\}\mu_k - \pi_k(t) \quad (5.28)$$

Because $F(r_k(t)) - \pi_k(t)$ is a non-increasing function related to $r_k(t)$, the required rate can be expressed by

$$r_k(t) \geq F^{-1}(E\{\vartheta_k(t)\}\mu_k - \pi_k(t)) = \alpha_k(t) \quad (5.29)$$

The estimate of average packet delay for the $k^{th}$ user can be expressed by Little's law as

$$\hat{d}_k(t+1) = \frac{\hat{Q}_k(t+1)}{E\{\vartheta_k(t)\}} \tag{5.30}$$

where $\hat{Q}_k(t+1)$ is the estimated actual queue length at the end of time slot *t+1*,

and $$\hat{Q}_k(t+1) = \min\{B, \hat{U}_k(t+1)\}$$

$$= \min\{B, H(r_k(t))\} = G(r_k(t)) \tag{5.31}$$

Because $G(r_k(t))$ is also a non-increasing function related to $r_k(t)$, to meet the packet delay requirement $d_k$,

$$\hat{d}_k(t+1) = \frac{\hat{Q}_k(t+1)}{E\{\vartheta_k(t)\}} = \frac{G(r_k(t))}{E\{\vartheta_k(t)\}} \le d_k \tag{5.32}$$

and

$$r_k(t) \ge G^{-1}(E\{\vartheta_k(t)\} d_k) = \beta_k(t).$$

By combining (5.29) and (5.32), one can get

$$r_k(t) \ge \max\{\alpha_k(t), \beta_k(t)\}. \tag{5.33}$$

Then the average PDP and the average packet delay performance could be linked together in the system.

### 5.3.1.2 Problem Formulation

With the above packet-level QoS analysis, the MCPF scheduling problem can be formulated as an MCQPF scheduling problem

$$\max \sum_{k=1}^{K} \log\left(1+\frac{\sum_{m=1}^{N_F} x_{k,m}(t) r_{k,m}(t)}{(t_c-1)\overline{R_k(t)}}\right) \tag{5.34}$$

subject to

$$(I)\ \sum_{k=1}^{K} x_{k,m}(t)=1, \qquad x_{k,m}\ \{0,1\}$$

$$(II)\ r_k(t) \le \frac{Q_k(t)N}{T_0}$$

$$(III) r_k(t) \ge \max\{\alpha_k(t), \beta_k(t)\}$$

where constraint (I) and (II) ensure that the scheduler does not serve the empty queue and constraint (III) is useful for improving the average packet delay and average PDP performance metrics.

### 5.3.1.3 Cross-Layer Design for QPF Scheduling in OFDM Systems

To address the QPF scheduling problem, a two step cross-layer QPF algorithm is presented. First the greedy PF scheduling algorithm described in section 5.3.1 is modified to ensure that constraint (II) in Equation (5.34) is satisfied. Then, the subcarrier reassignment procedure is implemented to improve the QoS performance according to constraint (III).

The modified greedy PF scheduling algorithm is described as follows. When the $m^{th}$ subcarrier is allocated to $k^{th}$ user with data rate $r_{k,m}(t)$ to get the largest PF value, then the user's queue length is updated by

$Q_k(t+1) = Q_k(t) - \left(\frac{r_{k,m}(t)T_0}{N}\right)$. If $Q_k(t) \leq 0$, then the queue is empty, and consequently, in the following scheduling steps, this user will not be served to avoid wasting resources.

In the subcarrier reassignment process, the set of subcarriers allocated to the $k^{th}$ user $C_k$ at time slot $t$ can be obtained after performing the previously modified greedy PF scheduling algorithm with the corresponding $r_k(t)$ and $r_{k,m}(t)$,. Let $\Psi$ be the user set in which each user's packet-level QoS performance constraint (III) is satisfied, i.e.,

$$\Psi = \{k : r_k(t) \geq \max\{\alpha_k(t), \beta_k(t)\}\} \subseteq U \tag{5.35}$$

and let $\Psi^c = U - \Psi$ be the user set where each user's QoS performance requirement is not satisfied.

If $\Psi = U$ and if each user's QoS requirement is satisfied, then the allocation is already optimal. Thus the packets can be transmitted at the allocated subcarriers with an assigned data rate. Otherwise, the subcarrier reassignment procedure is needed. Defining $S_\Psi = \{m : m \in C_k, k \in \Psi\}$ as the subcarriers allocated to the users whose rate limits are satisfied, some subcarriers in $S_\Psi$ should successively be reassigned to the users in $\Psi^c$. To control the subcarrier reassignment while maintaining good PF performance, a new swap value PF($i$,$j$,$m$) is defined where the $m^{th}$ subcarrier previously belonging to the $i^{th}$ user $(i \in \Psi)$ is assigned to the $j^{th}$ user $(j \in \Psi^c)$, i.e.,

$$\begin{aligned} PF(i, j, m) = \log\left(1 + \frac{r_i(t) - r_{i,m}(t)}{(t_c - 1)\overline{R_l(t)}}\right) + \log\left(1 + \frac{r_j(t) + r_{j,m}(t)}{(t_c - 1)R_j(t)}\right) \\ + \sum_{k \neq i,j} \log\left(1 + \frac{r_k(t)}{(t_c - 1)\overline{R_k(t)}}\right) \end{aligned} \tag{5.36}$$

If the $i^{th}$ user can still satisfy the QoS constraint (III) and the corresponding PF($i$,$j$,$m$) is the largest among all of the reassignment attempts, then this $m^{th}$ subcarrier can be reassigned to the $j^{th}$ user. The reallocation process continues until its rate requirement is satisfied or there are not enough subcarriers that could be assigned to this user if the users in the system are experiencing deep fading.

#### 5.3.1.4 Qualitative Comparison

Based on the preceding analysis, the QPF algorithm can be compared with the modified greedy MCPF scheduling algorithm, i.e., the first step of QPF algorithm excluding the subcarrier reassignment step. The QPF algorithm is the only approach to take efficiency, fairness, and QoS into consideration together. For a system with the $K$ users and $M$ subcarriers, the complexity and other parameters are given in the table 5.1. As shown in table 5.1, the QPF method is the only approach to take efficiency, fairness and QoS into consideration together. The total computational complexity of the QPF is at most $2KN_{F,}$ which is twice that of Greedy MCPF and Conventional PF (CPF) scheduling algorithms. But it is also acceptable even when adding the additional QoS improvement.

**Table 5.1 Qualitative Comparison of the Scheduling Algorithms**

| **Method** | **Efficiency** | **Fairness** | **QoS** | **Complexity** |
|---|---|---|---|---|
| QPF | Yes | Yes | Yes | $2KN_F$ |
| Greedy MCPF | Yes | Yes | No | $KN_F$ |
| CPF | Yes | Yes | No | $N_F^k$ |

#### 5.3.1.5 Simulation Results

The following parameters are assumed to analyze the performance of Quality of Service (QoS) aware Proportional Fair (QPF) scheduling algorithm. The MultiCarrier Proportional Fair scheduling (MCPF) and

Conventional PF scheduling (CPF) algorithms are compared with the proposed QPF algorithm. The parameters and the corresponding values used for simulation are listed in the Table 5.2.

**Table 5.2 Simulation Parameters for QPF Algorithm Implementation**

| Parameter | Value |
|---|---|
| System Bandwidth | 20 MHz |
| Number of Subcarriers (M) | 256 |
| Number of Users (K) | 40 |
| BER requirement | $10^{-5}$ |
| Packet Length | 100 bit/packet |
| Queue size | 200 |
| Average window size | 80 |

The average system throughput is defined as the average transmitted bit per second in the system. As can be seen from Figure 5.1, the QPF algorithm achieves the highest average system throughput compared to the CPF and greedy MCPF algorithms.

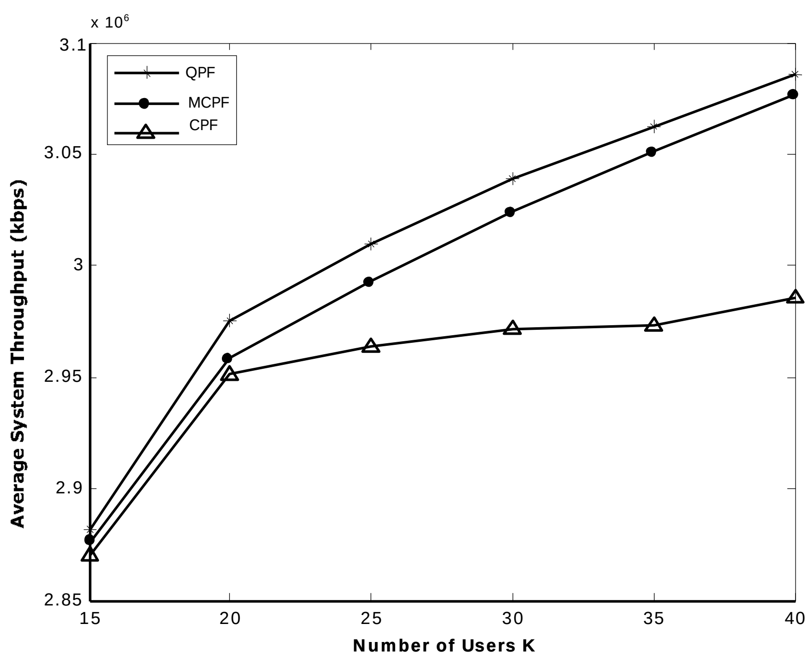


**Figure 5.1 Average System Throughput Versus Number of Users**

When the numbers of users increase, the dropped and delayed packets also increase if there is no QoS control. Because CPF and MCPF algorithms do not consider the QoS, the number of dropped packets and delayed packets are higher than that of QPF and hence there is a decrease in average system throughput of CPF and MCPF algorithms compared to the proposed QPF algorithm. When the traffic density is high, e.g., $K > 20$, the throughput variances significantly increase with the number of users due to the resource competition among users. When the traffic density is low, e.g., $K \leq 20$, the throughput variances among the users for both MCPF and QPF algorithms are similar to each other, because under these conditions, an individual user may obtain enough resources to satisfy its throughput and QoS requirements.

The fairness performance is indicated by the fairness index. The fairness index can range between zero and one and the value closer to one means fairer. From Figure 5.2, it is found that the fairness performance results of MCPF and QPF are much better due to their capabilities to balance efficiency and fairness for resource scheduling. But for CPF, which is not having such capabilities, the fairness performance is poor. Due to the process of QoS control in QPF algorithm, some transmission chances are switched from the QoS satisfied users to some other unsatisfied users. Hence the packets that would likely be dropped due to the deadlines missed or overflow when using MCPF without QoS consideration are successfully selected for transmission by the QPF. Thus, the radio resource is allocated in a fairer manner among users by using the QPF algorithm.

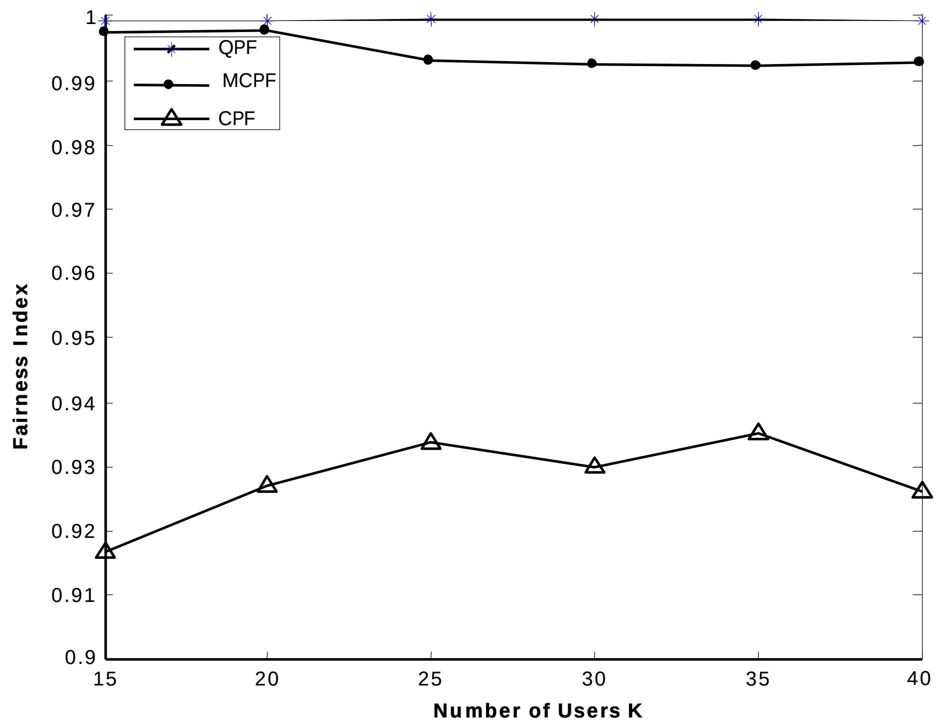


**Figure 5.2 Fairness Index Versus Number of Users**

Packet dropping probability (PDP) is measured in terms of the PDP violation probability. PDP violation occurs when the calculated average PDP at the end of one scheduling slot for a particular user is not satisfied with the predefined requirement. PDP violation probability is defined as the ratio of the number of occurred PDP violations over all the scheduling time slots for all the users to the total number of calculated PDP over these time slots for these users.

A PDP violation occurs when the calculated average PDP at the end of one scheduling slot for a particular user is not satisfied. From the Figure 5.3, it is observed that the QPF algorithm considerably improves the packet dropping performance than CPF and MCPF, particularly when the traffic density (i.e.) the number of users is high (K > 20) due to the consideration of QoS control.

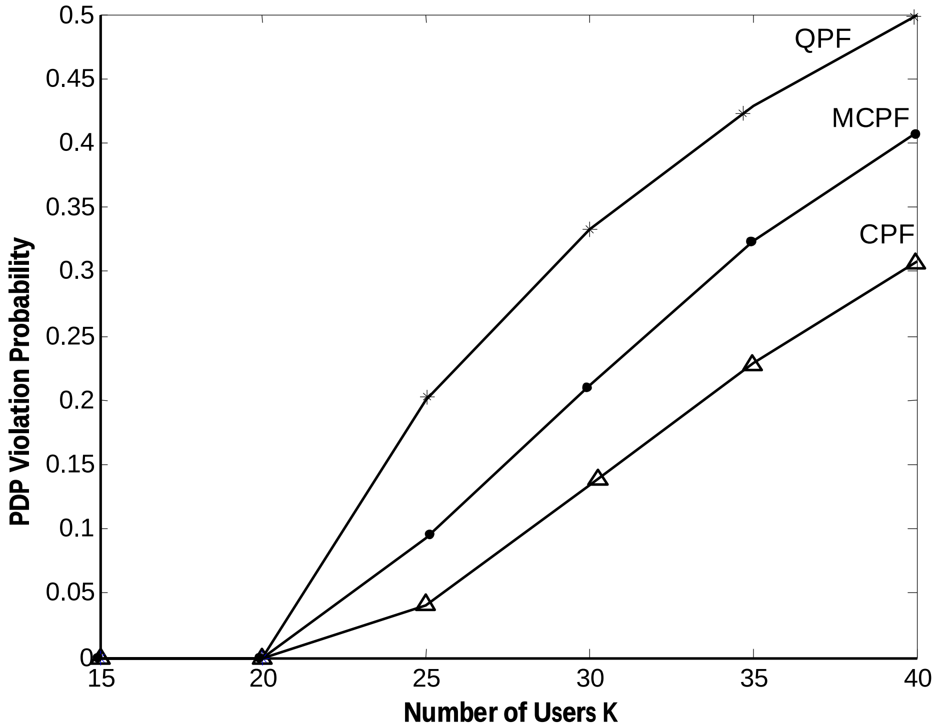


**Figure 5.3 PDP Violation Probability Versus Number of Users**

The average packet delay is the average time that a packet must wait when trying to access a communication network and is given by

$$\text{Average Packet delay} = \frac{N_{wl}}{\lambda(1 - p_d)} \tag{5.37}$$

where $N_{wl}$ is the number of packets in wireless links, $\lambda$ is the arrival rate and $p_d$ is the packet dropping probability.

As seen from the Figure 5.4, the average packet delay of QPF algorithm is the smallest compared to the MCPF and CPF algorithms because it can control the packet level QoS and schedule the packets according to the CSI as well as QSI. In MCPF algorithm, due to high capacity, the delay can be relaxed to combat the channel fading for OFDM systems. But in CPF algorithm, the delay requirement is related to the maximum time for which a user can be starved during poor channel conditions. When the number of users increases, e.g., K > 25, the packet delay deadline also increases, in turn it creates more number of packet losses. So CPF algorithm takes more delay to complete its transmissions.

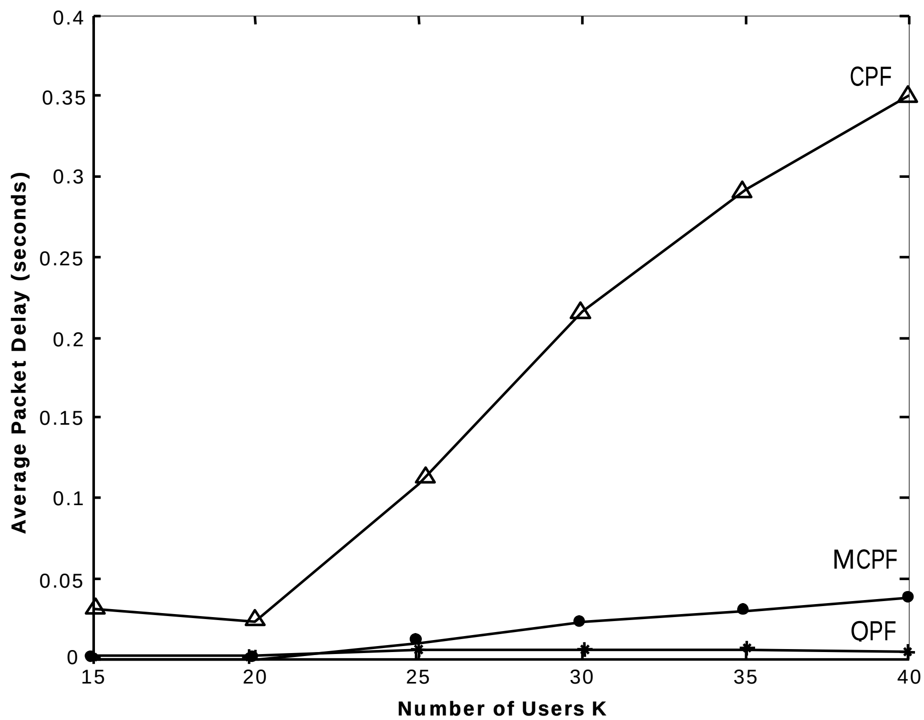


**Figure 5.4 Average Packet Delay Versus Number of Users**

### 5.3.2 Adaptive Cross-Layer Packet (ACP) Scheduling

With the increasing demands for high capacity and multi-tasking services, wireless communication systems are expected to support multiple users, each may have multiple heterogeneous traffic queues simultaneously. For example, a user can use a mobile device to make a phone call, playing the internet TV and download files from a website, at the same time. To support such multiuser multi-tasking services, an Adaptive Cross-layer Packet (ACP) scheduling algorithm is proposed in this section.

#### 5.3.2.1 System Model

A general downlink Time Division Duplexing (TDD) OFDM system is considered. The base station can acquire the Channel State Information (CSI) through the uplink dedicated pilots from all mobile stations at the beginning of each time slot and can use it for resource allocation and scheduling. The subcarrier and power controller at the PHY layer performs subcarrier and power allocation, and the traffic controller at the MAC layer performs data scheduling. With a cross-layer design, the QoS information obtained by the traffic controller is transferred to the subcarrier and power controller for resource allocation and the resource allocation results are fed back to the traffic controller in the base station for scheduling of the data to be sent out in each slot. The cross-layer scheduling architecture for multitasking OFDM system is depicted in Figure 5.5. Assume a total bandwidth of *B* which is divided into independent subcarriers and shared by *K* users. The total transmit power from the base station is assumed to be $P_t$.

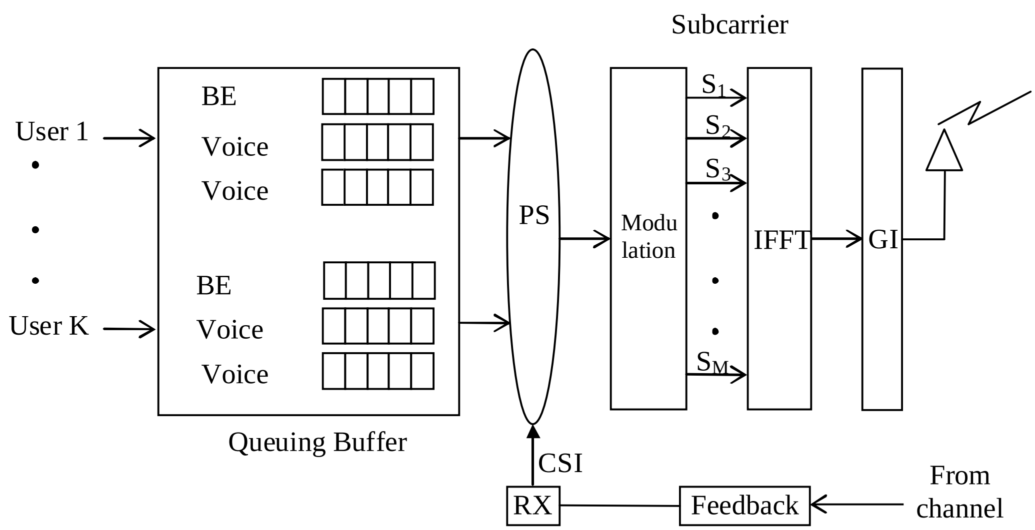


**Figure 5.5 Cross-Layer Scheduling Architecture for Multi-tasking OFDM System**

The OFDM signalling is time-slotted and each time slot is of length $T_{slot}$. A quasi-static fading channel is assumed where the channel gain is constant during each slot and is independent of the channel for other slots.

Each user has $I_k$ traffic queues with heterogeneous delay constraints for the queues. $\Omega_k$ is defined as the index set of subcarriers allocated to user $k$. Without loss of generality and for simplicity, one can assume that each subcarrier is allocated to only one user. Let $p_{k,n}$ denote the power allocated to user $k$ on subcarrier $n$, $h_{k,n}$ the corresponding channel gain, and $N_0$ the single-sided power spectral density of Additive White Gaussian Noise (AWGN). Assuming perfect channel estimation, the achievable instantaneous data rate of user $k$ on subcarrier $n$ is expressed as

$$R_{k,n} = \frac{B}{N}\log_2\left(1 + p_{k,n}.\gamma_{k,n}\right) \tag{5.38}$$

where $\gamma_{k,n} = \left|h_{k,n}\right|^2 / N_0\ B / N$ is the channel-to-noise power ratio for user $k$ on subcarrier $n$. The total achievable instantaneous data rate of user $k$ is given by

$$R_k = \sum_{n \in \Omega_k} R_{k,n} \tag{5.39}$$

#### 5.3.2.2 Problem Formulation

A general maximum weighted sum capacity (MWSC) based cross-layer design is employed to maximize the weighted sum of all users instantaneous capacities as in Cover and Thomas (1991). If $W_k$ and $R_k$ denote the weight containing the QoS information and the achievable instantaneous data rate for user $k$, respectively. $W_k$ can be determined by scheduling at the MAC layer and the cost function to be maximized is given by

$$J = \sum_{k=1}^{K} W_k R_k \tag{5.40}$$

subject to the constraints :

(C1) $p_{k,n} \geq 0$,

(C2) $\sum_{k=1}^{K} \sum_{n \in \Omega_k} p_{k,n} \leq p_T$ ,

(C3) $R_k T_{slot} \leq Q_k$ ,

(C4) $\Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_k \subseteq \{1, 2, ..., N\}$,

(C5) $\Omega_k \cap \Omega_j = \phi (k \neq j)$

where $Q_k$ is the queue length of user *k*. The constraint (C3) guarantees that no more resource is allocated to the user *k* if the user already obtains sufficient resources to send all data out in current slot, to avoid waste of resources.

#### 5.3.2.3 Maximum Weighted Sum Capacity Based Resource Allocation

To make a good trade off between complexity and performance, a suboptimal resource allocation algorithm is proposed that performs subcarrier allocation and power allocation separately. Subcarrier allocation is performed first by assuming equal power across all subcarriers and then the power allocation. With equal power on each subcarrier, the subcarrier allocation should assign subcarrier *n* to user *k* rather than user *j*, if $W_k R_{k,n} > W_j R_{j,n}$. After subcarrier allocation, power allocation is performed by using Karush-Kuhn-Tucker (KKT) conditions. Let Φ denote the user index set. The steps involved in the dynamic algorithm to implement the suboptimal MWSC based subcarrier and power allocation is given below:

1) Initialize the set for all users and for all subcarriers

$$\Phi = \{1, 2, \cdots, k\}, R_k = 0 \,,\; \Omega_k = \phi \,,\; p_{k,n} = \frac{p_T}{N}$$

2) For all the subcarriers, find

$$k^*(n) = \arg \max_{k \in \phi} \{W_k R_k\}$$

3) Assign subcarrier $n$ to user $k^*(n)$ and then update $k^*(n)$ carrier index and data rate.

4) Allocate the power to subcarrier by

$$p_{k,n} = \begin{cases} \dfrac{W_k \left( \sum\limits_{m=1}^{K} \sum q = \Omega_m \dfrac{11}{\gamma_{m,q}} \right)}{\sum\limits_{m=1}^{K} W_m \left|\Omega_m\right|} - \dfrac{1}{\gamma_{k,n}} & k = k^*(n) \\ 0 & k \neq k^*(n) \end{cases}$$

5) For all subcarriers, if power is less than zero for a particular user, then there is no enough power for subcarrier $n$ which has small weight and/or signal-to- noise ratio, thus subcarrier $n$ will be ignored for power allocation.

Repeat the steps 3 and 4 until the power is greater than zero.

The user based cross-layer design for multi-tasking can be easily extended to the queue based cross-layer design, by replacing the user index $k$ with the queue index $i$. Assuming that all users have $w$ queues each, *i.e.*, $I_k = w$, there are total number of $w$K queues in the system. The queue based MWSC cross-layer design is to maximize the cost function

$$J = \sum_{i=1}^{wK} W_i R_i \tag{5.41}$$

subject to the constraints :

(C1) $p_{i,n} \geq 0$,

(C2) $\sum_{i=1}^{K} \sum_{n \in \Omega i} p_{i,n} \leq p_T$ ,

(C3) $R_i T_{slot} \leq Q_i$ ,

(C4) $\Omega_1 \bigcup \Omega_2 \bigcup \cdots \bigcup \Omega_{wk} \subseteq \{1, 2, ..., N\}$,

(C5) $\Omega_{k \backslash i} \cap \Omega_j = \phi (i \neq j)$

where $W_i$, $R_i$ and $Q_i$ denote the weight, instantaneous data rate and queue length of queue *i* and $\Omega_i$ is the index set of subcarriers allocated to queue *i*.

Since the number of users is usually smaller than the number of queues (when $w > 1$), with given weights, the user based MWSC cross-layer design requires a lower complexity than the conventional queue based cross-layer design.

The previous cost functions defined in Johnsson et al (2005) focus on either minimising packet delay or maximising system throughput. In general, satisfying one measure scarifies the other. There are algorithms that attempt to satisfy both measures. However, they tend to perform worse for one or both measures compared with algorithms that focus on only one.

In this section, a new packet scheduling is introduced, which minimizes a prescribed cost function, given the current channel qualities in terms of SNR and delay states of the packets in the queue. Simulations show that it outperforms existing schedulers both in terms of packet delay and system throughput. The weight of the MWSC based cross-layer design contains the QoS information and can be obtained from scheduling at the MAC layer.

The name Adaptive Cross-layer Packet (ACP) scheduling is derived from the fact that the algorithm adapts the scheduling order to changes in two variables across layers, packet delay deadlines on the link layer and channel qualities on the physical layer. Depending on the measurement and processing capabilities of the system, the ACP algorithm may use instantaneous or average channel quality estimates. Thus, the algorithm may track small scale or large scale channel variations. More frequent and accurate estimates result in better ACP performance. Since this algorithm schedules packets according to both delay deadlines and channel qualities, it performs very well with respect to packet delay and system throughput.

The ACP algorithm schedules packets in the order that minimizes the cost function *J*. In particular, the ACP algorithm compares the total estimated cost *J* of all users' packet, then schedules the packets according to the scheduling permutation that minimizes *J*. Scheduling permutations and cost estimates are only valid as long as the current channel and queue conditions persist. Due to user mobility and the flux of users entering and leaving the system, channel estimates and scheduling orders may be invalid by the next transmission opportunity.

The cost function is defined by

$$J = \arg\min_{k} \left( \arg\min_{i} \sum_{i=1}^{M} (1-\eta) \frac{\hat{d}_i}{r_i} + \eta \max\left\{ 0, \frac{\hat{d}_i}{r_i} - 1 \right\} \right) \tag{5.42}$$

where

$$\hat{d}_l = S_{k,i,l} + \sum_{j=1}^{l} \frac{D_{k,i,j}}{R_i} \tag{5.43}$$

The quantity $\hat{d}_l$ is the delay estimate, $S_{k,i,l}$ is the current delay of the $l$th packet. The delay estimate for $l^{th}$ packet, $\hat{d}_l$, is the sum of its current delay and the total estimated transmit time of all packets before and including it. The estimated transmit time of packet $l$ is given by $D_{k,i,l} / R_i$, where $D_{k,i,l}$ and $R_i$ are the packet size in bits and the instantaneous data rate of queue $i$ to which the packet belongs respectively and $M$ is the total number of packets in the queue at a given scheduling event. The coefficient $0 \le \eta \le 1$ is a weighting parameter between the estimated normalized packet delays and missed deadline penalties. The ACP scheduling algorithm runs whenever a new packet enters the queue or a queued packet's channel quality changes by some absolute value. Let $r_l$ be the delay requirement of $l^{th}$ packet and it is given by (Johnsson et al 2005)

$$r_l = r(c, D_{k,i,l}) = \begin{cases} \frac{(50 * D_{k,i,l})}{1024} & \mathrm{c = BE, D_{k,i,l} \le 1024} \\ 50 + \frac{25(D_{k,i,l} - 1024)}{8192 - 1024} & \mathrm{c = BE, D_{k,i,l} > 1024} \\ \frac{(0.5 * D_{k,i,l})}{1024} & \mathrm{c = Voice, D_{k,i,l} \le 1024} \\ 0.5 + \frac{1.5(D_{k,i,l} - 1024)}{8192 - 1024} & \mathrm{c = Voice, D_{k,i,l} > 1024} \\ \frac{(5 * D_{k,i,l})}{1024} & \mathrm{c = Video, D_{k,i,l} \le 1024} \\ 5 + \frac{10(D_{k,i,l} - 1024)}{8192 - 1024} & \mathrm{c = Video, D_{k,i,l} > 1024} \end{cases} \tag{5.44}$$

where $c$ is the packet's delay class like best effort, voice and video.

#### 5.3.2.4 Simulation Results and Performance Comparison

In this section, the performances of different scheduling algorithms for a multicarrier OFDM network are compared. Simulation results are shown to demonstrate performance of the various resource scheduling algorithms for a system with the simulation parameters given in Table 5.3.

**Table 5.3 Simulation Parameters for Various Scheduling Algorithms in Multicarrier OFDM Networks**

| Parameters | Values |
|---|---|
| Total system bandwidth (B ) | 5MHz |
| Slot duration ($T_{slot}$ ) | 2 msec |
| Total system transmission power ($P_T$ ) | 1 W |
| Number of subcarriers | 512 |
| Delay tolerance for queues ($U_{k,i}$) | 100, 400 and 1000 msec |
| Guard interval ($G_{k,i}$) | 100 msec for all traffic types |
| Packet inter-arrival rate | 1 msec |
| Packet arrival rates of voice and BE traffic | 64 kbps and 500 kbps respectively |
| Truncated exponential distribution of Video | 420 kbps (max), 120 kbps (min) and 239 Kbps (mean) |
| Average packet size (bits) | 64 (voice), 239 (video) and 500 (BE) |
| QoS priority levels ($\beta_{k,i}$) | 1024 (voice), 512 (video) and 1 (BE) |
| Delay spread | 0.5 μsec |

The duration of each state of truncated exponential distribution of video follows an exponential distribution with mean 160 msec. The channel is modeled to be of six independent Rayleigh fading paths with an exponential delay profile. The parameters taken for comparison are average system throughput and average packet delay.

The Figure 5.6 shows the impact of the number of users on the average system throughput for various scheduling algorithms discussed in this chapter. Very high SNR represents the good quality of the channel. Similarly low SNR represents the poor quality of the channel. Generally, when the channel is with high SNR or good quality, it is possible to increase the number of users accommodating the channel and it is true for vice-versa (Goldsmith 2005). The ACP scheduling algorithm demonstrates significant performance advantages over other scheduling algorithms except MSC since it depends on the instantaneous data rate and schedules the packet only when the channel has high SNR or if less number of users are sharing the channel. The system throughput achieved by ACP increases with the increase in number of users, due to enhanced multiuser diversity. With given resources, when the number of users increases, there is a higher degree of freedom for resource allocation, resulting in enhanced multiuser diversity.

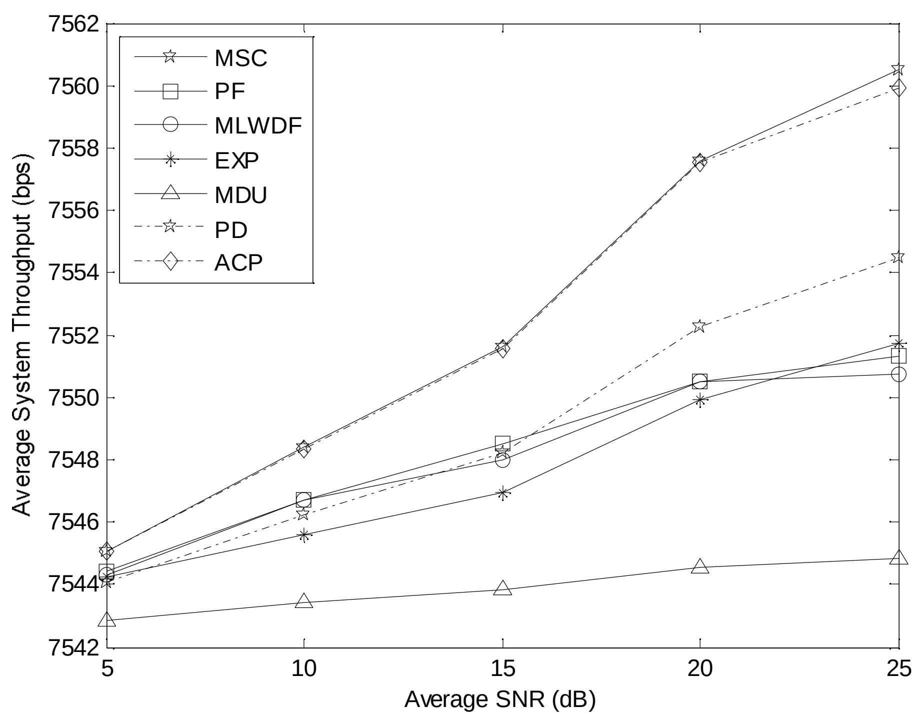


**Figure 5.6 Impact of Number of Users on Average System Throughput**

With ACP, the packets with larger weights are served first on packet basis whichever queues they belong to, whereas with MDU, M-LWDF and EXP all packets in a queue which has a larger weight are served first on queue basis. Therefore average system throughputs of PD, MDU, M-LWDF and EXP benefit less from multiuser diversity. With a large number of users, the average system throughput achieved by ACP is much higher than that achieved by MDU, M-LWDF and EXP scheduling algorithms. With PD, the packet weights are determined based on the packet size, delay and QoS priority levels. So the performance is better compared with some other algorithms.

Figure 5.7 shows the impact of the number of users on the average packet delay. The EXP scheduling is poor compared to the MLWDF and the MDU algorithms. This is due to the mechanism of the EXP rule. If one user has a larger delay than others, the weight of this user becomes very large because of the exponential function used in the weight and then this user may occupy all of the subcarriers with high probability.

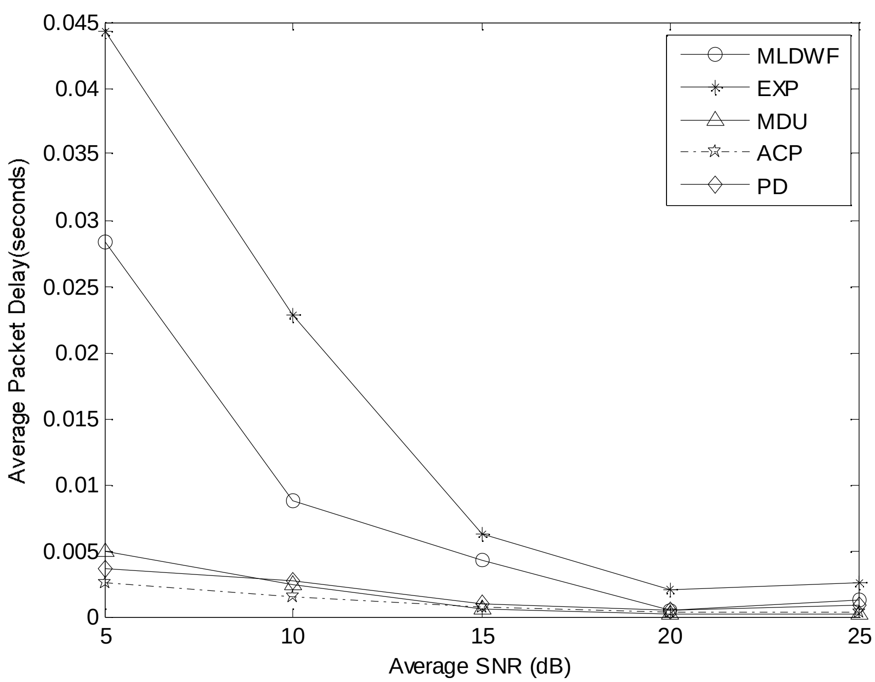


**Figure 5.7 Impact of Number of Users on Average Packet Delay**

Due to the presence of frequency-selective fading, assigning the whole bandwidth to one user is less efficient. As can be seen, the MLWDF works well compared to EXP. However, since the MDU policy uses the average queue lengths (delays) as the weights, which are a more moderate way, the MDU policy can allocate resources more efficiently compared to MLWDF.

The ACP achieves a much lower delay than PD, MLWDF, MDU and EXP since it determines the packet transmission order by assigning different weights to different packets, and therefore is more efficient than the conventional queue dependent scheduling. Since MSC and PF algorithms are channel aware only scheduling algorithms and do not consider the queues, comparison of their delay performances with the other queue only aware and joint channel-and queue-aware scheduling algorithms is not fair. Hence, they are not shown in Figure 5.7.

Thus the Adaptive Cross-layer Packet scheduling is found to perform better compared to other queue based resource scheduling algorithms such as PD, M-LWDF, MDU, and EXP in terms of system throughput and average packet delay.

## 5.4 SUMMARY

The joint channel- and queue- aware multicarrier scheduling in OFDM networks are investigated in view of several important aspects. Based on different users' channel conditions as well as their queue states, a cross-layer scheduling algorithm called QoS aware Proportional Fair (QPF) is proposed for the multimedia services in a multiuser OFDM system. The design of QPF is to achieve PF in the system while improving the different user's packet level QoS performances. With the consideration that the traditional single carrier PF (SCPF) method is not suitable for the OFDM systems, an efficient greedy method is proposed based on the MCPF criterion with a relatively low computational complexity to allocate subcarriers to different users in order to achieve PF. Then, based on the analysis of the packet level QoS performance, a subcarrier reassignment procedure has been proposed to improve the QoS performance. Through simulation, it is found that that the benefits of introducing QoS control into PF scheduling for multicarrier OFDM transmission are threefold. First, the occurrences of packet overflow and deadlines missed significantly decrease and hence the delay and PDP performances are improved. Secondly, with the decrease of failed packets, the throughput performance also correspondingly increases. Third, the radio resource can be allocated to users in a fairer manner when more users become satisfied with their QoS performances. Therefore, PF scheduling method with low complexity can be used in the OFDM system to achieve good performances in terms of throughput, average PDP, packet delay and fairness.

An Adaptive Cross-layer Packet (ACP) scheduling for the downlink multiuser multitasking OFDM systems with heterogeneous traffic is also presented in this chapter that proves significant performance advantages over the queue based MLWDF, EXP, MDU and PD scheduling algorithms in terms of the system throughput and QoS traffic delays. Simulation results show that ACP scheduling has better throughput and delay performance than other scheduling algorithms and that the combination of scheduling and power allocation can significantly improve the performance further. Cross-layer based QPF and ACP scheduling algorithms require much lower complexity than conventional queue based designs.