TKN Telecommunication Networks Group

Technical University Berlin

Telecommunication Networks Group

Optimal Packet-Centric Down-Link Scheduling in Cellular OFDMA

Mathias Bohge, James Gross, Michael Meyer

{bohge,gross}@tkn.tu-berlin.de, michael.meyer@ericsson.com

Berlin, September 2006

TKN Technical Report TKN-06-006

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

Abstract

The scheduling decisions made by the scheduler in a packet-oriented network have a significant impact on the network performance. By the use of cross-layer optimization techniques, the packet scheduling process can be supported, such that system resources are allocated with respect to the user's current channel state and data requirements. In this paper, we present means to obtain optimal scheduling decisions for a base-station scheduler of a packet-centric wireless OFDMA cell. We study the influence of optimal decisions on the system's throughput performance, while observing the impact on signaling costs and complexity. Moreover, we derive two mechanisms that deliver sub-optimal results based on relaxation and heuristics, and compare their performance to the optimal case. [1]

[1]This work has been supported by the BMBF and Ericsson Research, Germany, in the context of the project ScaleNet.

Contents

1 Introduction **2**

2 System Model **3**
2.1 Physical Layer . . . 3
2.2 Wireless Channel Model . . . 3
2.3 Medium Access Control Layer . . . 4
2.4 Involved Signaling Overhead . . . 4

3 Optimal Packet Centric Scheduling **6**

4 QoS Constraints **8**

5 Suboptimal Scheduling **9**
5.1 Relaxation Approach . . . 9
5.2 Heuristic Approach . . . 10

6 Performance Analysis **14**
6.1 Simulation Model . . . 14
6.2 Throughput and Delay Results . . . 14
6.3 Signaling Results . . . 16
6.4 Complexity Observations . . . 18

7 Conclusions **19**

References **20**

Chapter 1

Introduction

Applying dynamic resource allocation mechanisms to down-link transmissions of cellular OFDMA (Orthogonal Frequency Division Multiple Access) systems has been shown to provide significant capacity increases, simply by utilizing the given bandwidth more efficiently [1–3]. This is mainly due to the fact that different sub-carriers of a broadband wireless system experience different attenuation conditions, i.e. the system provides frequency diversity. In addition, as the attenuation of the (same) sub-carriers for different terminals are statistically independent, multi-user diversity is present. This has lead to the proposal of *dynamic sub-carrier assignment schemes* that adaptively distribute the sub-carriers among all users.

In packet-oriented networks, these gains can be used in order to improve the performance of packet schedulers that are located at the base-station of a wireless cell. Several algorithms have been proposed to advance the scheduling decisions by exploiting the system's diversities. Some well-known among them are based on virtual clocks (VC), utility functions (UF), or general processor sharing (GPS) [4–6]. However, there is a lack of algorithms that decide on a per-packet basis. All algorithms mentioned above schedule bits per frame to adequate terminals, neglecting the sizes of packets available for transmission. Depending on the system model, this might result in significant capacity wastage [7], or fragmentation overhead. To avoid that, the *packet-centric* system-view has been suggested. In a packet-centric system, packets are the smallest allocatable data unit that must not be further divided. Thus, the allocated resources always have to be matched exactly to the packets scheduled for transmission.

In this paper, we present a mathematical optimization framework that allows us to determine optimal scheduling decisions in a packet-centric wireless OFDMA context. This general model can be adjusted in order to comply with different scheduling goals and QoS requirements, determined by the system administrator. However, as obtaining these optimal decisions is a very complex task, we present means to obtain near-optimal decisions at low computational cost. Particularly, we present a relaxation approach that delivers almost optimal decisions while staying in reasonable complexity bounds. Throughput performance analysis, as well as signaling and complexity considerations are provided and conclusions on optimal packet scheduling decisions are drawn. The remainder of this paper is organized as follows: in the next Chapter we present our System Model. In Chapter 3 and 4 we present the optimization framework that yields optimal performance. We introduce the sub-optimal mechanisms in Chapter 5 and compare optimal and sub-optimal performance results in terms of throughput, signaling cost and complexity in Chapter 6. Finally, in Chapter 7 we conclude the paper.

Chapter 2

System Model

We consider a single cell of an OFDM based packet-centric cellular system according to [8] with radius r_{cell} (cf. Figure 2.1). Within this cell, a base station coordinates all down-link data transmissions. Apart from receiving acknowledgements, we do not consider the up-link any further. J terminals are located within the cell. For each terminal, the base station features one FIFO (first-in-first-out) queue for each terminal and QoS (quality-of-service) class. Hence, at any point in time each packet buffered at the base station can be uniquely addressed by the triple $\{j, c, p\}$, where j is its destination terminal, c is its QoS class, and p is its queueing position in the respective QoS queue. The packet's size is given by $\varsigma_{j,c,p}$, its momentary delay by $\tau_{j,c,p}^{(t)}$.

2.1 Physical Layer

The system under consideration uses OFDM as transmission scheme for down-link data transmission. It features a total bandwidth of B [Hz] at center frequency f_{c}. The given bandwidth is split into S sub-carriers (with an equal bandwidth of B/S and an symbol length of T_{s} each). Modulation type and coding rate are selected by the base station's *adaptive modulation and coding* unit. Prior to the transmission of the time domain OFDM symbol, a cyclic prefix of length T_{g} is added as guard interval. The maximum transmit power P_{t} is equally split between the sub-carriers.

2.2 Wireless Channel Model

Terminals are uniformly distributed over the area of the cell. Each terminal's instant SNR $\gamma_{j,s}^{(t)}$ varies permanently due to reflecting and scattering objects within the cell that are moving with a speed up to $v_{\max}$. The SNR is given by:

$$\gamma_{j,s}^{(t)} = \frac{p \cdot (h_{j,s}^{(t)})^2}{\sigma^2}, \tag{2.1}$$

where $p = P_{\text{t}}/S$ denotes the transmission power per sub-carrier, $h_{j,s}^{(t)}$, denotes the adequate channel gain and σ^2 denotes the noise power per sub-carrier.

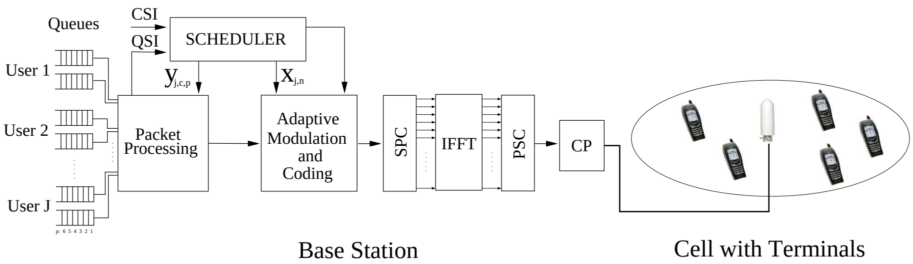

Figure 2.1: The system model consisting of a base station and a single cell.

2.3 Medium Access Control Layer

We consider a time-division-duplex (TDD) system. A consecutive up- and down-link frame pair forms a so called transmission-time-interval (TTI) of duration T_{TTI}. Dynamic OFDMA is applied to the down-link transmissions, i.e. during each TTI the sub-carriers can be arbitrarily allocated to the terminals. However, sub-carriers are grouped into N *chunks* (cf. Figure 2.2) and only chunks can be assigned. In the frequency domain, a chunk consists of a well defined number of adjacent sub-carriers Φ_{sub} such that sub-carrier gain conditions are strongly correlated within a chunk. Thus, the same modulation type is applied to each of them. The number of consecutive OFDM symbols belonging to one chunk Φ_{sym} depends on the TTI ($\Phi_{\mathrm{sym}} = T_{\mathrm{TTI}}/\left[2 * (T_{\mathrm{s}} + T_{\mathrm{g}})\right]$).

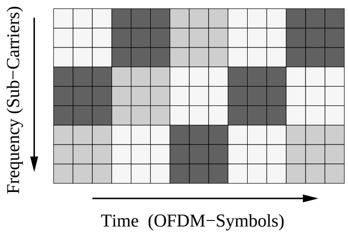

Figure 2.2: Chunks consisting of $3x3 = 9$ OFDM symbols.

The down-link SNR values per chunk are obtained at the base station by tracing the channel states during the preceding up-link frame, exploiting the wireless channel's reciprocity.

2.4 Involved Signaling Overhead

For each TTI terminals have to be informed about the dynamic chunk and modulation type assignments. For this purpose, a contol channel is required. We do not specify the control channel any further, but provide calculations of the load that has to be handled by the control channel.

The signaling information itself consists of the information which chunk has been assigned to which terminal with which modulation type (the triple <CHUNK_ID,TERM_ID,MOD_ID>). As these three parameters have to be addressed by binary identifiers, the signaling overhead obviously depends on the total amount of terminals, chunks and modulation types. However, it is clear that the exact binary load per TTI depends on the chosen representation of the signaling information.

Basically, we assume a rather simple way to represent the signaling information, which is referred to as fixed size signaling field (FSSF) approach (cf. [9]). For each chunk the current assignment is conveyed via the control channel for each TTI. As each chunk is addressed, the chunk identifier itself can be omitted. Instead, the position of the tuple <TERM_ID,MOD_ID>in the signaling bit-stream indicates the chunk it refers to. There are N such assignments, leading to a load given by $N \cdot (\lceil \log_2(J) \rceil + \lceil \log_2(M) \rceil)$ bit per TTI by the FSSF. However, this overhead can be compressed, as is discussed in Chapter 6.3.

Chapter 3

Optimal Packet Centric Scheduling

At the base station, the *Scheduler* is responsible of assigning chunks to the terminals and selecting the packets to be transmitted during the next TTI. According to the *packet centric* perspective, packets need to be scheduled and transmitted in one piece, whereas fragmentation on packet-level is not an option (cf. [8]). Several scheduling policies are known, ranging from very simple (e.g. *round robin*) to advanced adaptive (e.g. *proportional fair*) ones, where the term *adaptive* relates to the fact that the scheduler's decision is based on instant channel-state-(CSI) and/or queueing state information (QSI).

The choice of an adequate scheduling policy is crucial for the system's performance. In order to evaluate different scheduling approaches and find the most appropriate among them, the scheduling problem can be transfered into a mathematical optimization problem that results in optimal scheduling decisions. However, most formulations (e.g. [2, 5, 6]) do not result in actual packet assignments, but deliver capacity allocations. In [7], a new optimization formulation was presented that matches the capacity allocated to each terminal during TTI-frame t to the size of its packets to be delivered, and thus assures optimal capacity allocation in a packet centric context. This is done by *maximizing the sum over the sizes of all scheduled packets*, while bounding the packets scheduled for each terminal j to its allocated chunk capacity:

$$\max \sum_{j,c,p} \left[y_{j,c,p}^{(t)} \cdot \varsigma_{j,c,p}^{(t)} \right] \tag{3.1}$$

$$\sum_{n} \left[x_{j,n}^{(t)} \cdot \chi_{j,n}^{(t)} \right] \geq \sum_{c,p} \left[y_{j,c,p}^{(t)} \cdot \varsigma_{j,c,p}^{(t)} \right] \; \forall j, \tag{3.2}$$

where $\varsigma_{j,c,p}^{(t)}$ is the size of terminal j's packet in queue c at position p, $\chi_{j,n}^{(t)}$ number of information bits available for transmission on chunk n for terminal j, $y_{j,c,p}^{(t)}$ is the packet-assignment variable at time t:

$$y_{j,c,p}^{(t)} = \begin{cases} 0 & \text{if pkt no. } p \text{ in } j\text{'s queue } c \text{ is not scheduled} \\ 1 & \text{if pkt no. } p \text{ in } j\text{'s queue } c \text{ is scheduled,} \end{cases}$$

and $x_{j,n}^{(t)}$ is the chunk-assignment variable at time t:

$$x_{j,n}^{(t)} = \begin{cases} 0 & \text{if chunk } n \text{ is not allocated to terminal } j, \\ 1 & \text{if chunk } n \text{ is allocated to terminal } j. \end{cases}$$

Since we consider each chunk to be assigned to at most one terminal at a time, and assume FIFO queues, we need two more constraints in order to thoroughly transfer our system model into the mathematical representation:

$$\sum_j x_{j,n}^{(t)} \leq 1 \quad \forall j \tag{3.3}$$

$$y_{j,c,p}^{(t)} \geq y_{j,c,p+1}^{(t)} \quad \forall j, c, (p < \#\mathcal{P}) \tag{3.4}$$

The optimization problem formulation consisting of goal (3.1) and its constraints (3.2)– (3.4) belongs to the class of *Mixed Integer Linear Programs* (MILP). Solving the problem leads to the optimal binary packet-scheduling and chunk assignments ($y_{j,c,p}^{(t)}$ and $x_{j,n}^{(t)}$) with respect to the optimization objective (3.1).

Chapter 4

QoS Constraints

Without any further constraints, the assignments obtained in Chapter 3 are optimal with respect to the *maximal cell throughput* in a packet-centric system model. Such a raw throughput maximization usually leads to fairness problems between different terminals. Also, no support for different QoS classes is provided. By introducing additional constraints and priority weights, such better (with respect to fairness and QoS) optimization objectives can be achieved.

Consider two different QoS-classes BE and TD. BE is a non- constrained best effort class, whereas TD is a QoS class with strict delay requirements (timely delivery). In order to assure proportional fair packet delivery among the terminals the objective in (3.1) can be modified to include each terminal's throughput $\Psi_{j,k}^{(t)}$ over the last k TTIs:

$$\max \sum_{j,p} \left[y_{j,c,p}^{(t)} \cdot \frac{\varsigma_{j,c,p}^{(t)}}{\Psi_{j,k}^{(t)}} \right] . \tag{4.1}$$

While the objective in (4.1) features fairness between different terminals, it does not assure timely TD packet delivery. In order to guaranty TD packet timeliness, an additional QoS constraint has to be introduced:

$$\left(1 - y_{j,\mathrm{TD},p}^{(t)}\right) \cdot \left(\tau_{j,\mathrm{TD},p}^{(t)} + T_{\mathrm{TTI}}\right) \leq \mathcal{T}_{\mathrm{TD}} \quad \forall j, p \tag{4.2}$$

where $\tau_{j,\mathrm{TD},p}^{(t)}$ is the momentary delay of j's TD packet at position p and $\mathcal{T}_{\mathrm{TD}}$ is the maximum admitted TD delay. If the delay of each TD packet should not only be bounded, but minimized for each terminal, the TD packet delay $\tau_{j,\mathrm{TD},p}^{(t)}$ has to be included in the optimization goal. In the following formulation, a highly delayed TD packet has a higher weight than a less delayed packet, and is, thus, more likely to be scheduled. This implies that older TD packets are scheduled first:

$$\max \sum_{j,p} \left[y_{j,\mathrm{BE},p}^{(t)} \cdot \frac{\varsigma_{j,\mathrm{BE},p}^{(t)}}{\Psi_{j,k}^{(t)}} + \omega \cdot y_{j,\mathrm{TD},p}^{(t)} \cdot \tau_{j,\mathrm{TD},p}^{(t)} \right] . \tag{4.3}$$

Goal (4.3) maximizes proportional fair BE traffic, while minimizing the TD traffic delay. However, as maximizing the throughput usually conflicts with minimizing the delay, both partial goals in (4.3) need to be decoupled by the scaling factor ω. The higher ω, the higher is the delay minimization impact to the overall goal and vice versa. Note that ω should be related to the maximum BE packet size, as well as to the maximum TD delay in order to lie in a meaningful range.

Chapter 5

Suboptimal Scheduling

Solving the MILPs introduced in Chapter 3 and 4 delivers the optimal binary chunk and packet scheduling assignments according to the objectives (3.1), (4.1) or (4.3). However, solving these problems to *optimality* needs far too much time and computational power, mainly due to the integer constraints on the assignment variables. Instead, sub-optimal scheduling algorithms are required for the application in reality. In this paper, we present two such approaches: one based on *relaxation* as well as a *heuristic*.

5.1 Relaxation Approach

A common approach in integer programming is to intially relax the integer constraints and solve the corresponding linear program (LP). The advantage of the linear program is that it can be solved quite fast to optimality. Thus, we replace $y^{(t)}_{j,c,p}$ and $x^{(t)}_{j,n}$ with $\tilde{y}^{(t)}_{j,c,p}$ and $\tilde{x}^{(t)}_{j,n}$ which can take now any real-valued number in $[0;1]$. However, since we consider exclusive chunk assignments and do not admit fragmentation, the scheduler's output needs to be binary. Thus, we need to derive integer-valued $y^{(t)}_{j,c,p}$ and $x^{(t)}_{j,n}$ from the real-valued optimal LP results $\tilde{y}^{(t)}_{j,c,p}$ and $\tilde{x}^{(t)}_{j,n}$. This task is performed by the Real-To-Binary (RTB) algorithm . The RTB algorithm initially considers all non-zero packet assignments, ordered by QoS classes. These scheduled packets are then matched one by one to chunks, if appropriate chunks are still available. Potentially, only chunks with a real-valued share of 0.5 or larger for a specific terminal can be used for packet transmission to this terminal. The formal description of the algorithm is given in Algorithm 1, where $\mathcal{J}, \mathcal{N}, \mathcal{C}, \mathcal{P}$ are the sets of available terminals, chunks, QoS-classes and queue-positions (determined by the queue sizes) respectively, $\#$ defines their cardinality, $\varsigma^{(t)}_{j_0,c_0,p_0}$ is the size of the considered packet, $\chi^{(t)}_{j_0,n_0}$ the momentary number of bits (capacity) available on chunk n_0 for terminal j_0, $\mathcal{X}^{(t)}_{j_0}$ is the accumulated chunk capacity and $\mathcal{S}^{(t)}_{j_0}$ is the accumulated packet-size of terminal j_0 for the upcoming TTI. After this step, several (but probably not all) packets are matched to chunks, while there might remain chunks which have not been assigned yet. Then, the remaining chunks are distributed according to the algorithm given in Algorithm 2.

All in all, the relaxation approach consists of three different parts: solving the corrsponding LP, generating initial integer-valued packet- and chunk assignments from the real-valued LP solutions, and finally redistributing non-allocated chunks. Note that the second and third step are quite simple algorithms, which have extremely low run times in practise. Complexity-wise, the largest share is with solving the LP.

5.2 Heuristic Approach

Another way of obtaining integer-valued assignments is to derive a heuristic. The Chunk-Packet-Matching (CPM) algorithm (Algorithm 3) first selects a terminal j_0 according to the rule defined in algorithm Step 5. Then it searches j_0's queues for the next packet and, if successful, assigns the necessary amount of chunks. The terminal assignment in Step 5 follows the *maximize cell throughput* policy. However, as for the optimization goal in Chapter 3 other policies can be defined. E.g. proportional throughput fairness can be achieved, by substituting Step 5 by:

$$j_0 = \arg\max_{j \in J} \left(\frac{\chi(\overline{\gamma_j^{(t)}})}{\Psi_{j,\mathrm{k}}^{(t)}} \right) \quad , \tag{5.1}$$

where $\chi(\overline{\gamma_j^{(t)}})$ is terminal j's average chunk capacity, which depends on its average chunk SNR $\overline{\gamma_j^{(t)}}$. In this case, at the end of each run (once each TTI) the accumulated throughput values need to be updated accordingly:

$$\Psi_{j,\mathrm{k}}^{(t+1)} = \Psi_{j,\mathrm{k}}^{(t)} + \mathcal{S}_j^{(t)} - \mathcal{S}_j^{(t-k)}, \quad \forall j \in \mathcal{J} \tag{5.2}$$

Note that - in contrast to the optimal MILP decisions - the order with which the QoS classes are arranged in $\mathcal{C}$ has a major impact on the sub-optimal scheduling results. As lower indices in $\mathcal{C}$ are preferred, the QoS classes have to be added to $\mathcal{C}$ in decreasing priority order. To further prioritize a certain QoS class, the objective definition in Step 5 can be modified accordingly.

Initialize: $\forall j, n, c, p:\ y_{j,c,p}^{(t)} = 0,\ x_{j,n}^{(t)} = 0,\ \mathcal{X}_j^{(t)} = \mathcal{S}_j^{(t)} = 0$

for $(j_0 \in \mathcal{J})$ **do**
    for $(c_0 \in \mathcal{C})$ **do**
        for $(p_0 \in \mathcal{P})$ **do**
            if $(\tilde{y}_{j_0,c_0,p_0}^{(t)} > 0)$ **then**
                for $(n_0 \in \mathcal{N})$ **do**
                    if $(\tilde{x}_{j_0,n_0}^{(t)} > 0.5)$ **then**
                        $x_{j_0,n_0}^{(t)} = 1$
                        $N := N \setminus \{n_0\}$
                        $\mathcal{X}_{j_0}^{(t)} = \mathcal{X}_{j_0}^{(t)} + \chi_{j_0,n_0}^{(t)}$
                        if $(\varsigma_{j_0,c_0,p_0}^{(t)} \leq \mathcal{X}_{j_0}^{(t)} - \mathcal{S}_{j_0}^{(t)})$ **then**
                            $y_{j_0,c_0,p_0}^{(t)} = 1$
                            $\mathcal{S}_{j_0}^{(t)} = \mathcal{S}_{j_0}^{(t)} + \varsigma_{j_0,c_0,p_0}^{(t)}$
                        end
                    end
                end
            end
        end
    end
end

Algorithm 1: The Real-To-Binary (RTB) algorithm,

$\mathcal{C}, \mathcal{J}, \mathcal{P}, \mathcal{N}$, $\mathcal{X}_j$ and $\mathcal{S}_j$ are taken from RTB algortihm (Alg. 1).

for $(c_0 \in \mathcal{C})$ **do**
  for $(j_0 \in \mathcal{J})$ **do**
    for $(p_0 \in \mathcal{P})$ **do**
      if $(\tilde{y}^{(t)}_{j_0,c_0,p_0} > 0 \quad \&\& \quad y^{(t)}_{j_0,c_0,p_0} \neq 1)$ **then**
        while $(\#\mathcal{N} > 0)$ **do**
          $n_0 = \arg\max_{n \in N}(\chi^{(t)}_{j_0,n})$
          $N := N \setminus \{n_0\}$
          if $(\tilde{x}^{(t)}_{j_0,n_0} > 0.5)$ **then**
            $x^{(t)}_{j_0,n_0} = 1$
            $N := N \setminus \{n_0\}$
            $\mathcal{X}_{j_0} = \mathcal{X}_{j_0} + \chi^{(t)}_{j_0,n_0}$
            if $(\varsigma^{(t)}_{j_0,c_0,p_0} \leq \mathcal{X}_{j_0} - \mathcal{S}_{j_0})$ **then**
              $y^{(t)}_{j_0,c_0,p_0} = 1$
              $\mathcal{S}_{j_0} = \mathcal{S}_{j_0} + \varsigma^{(t)}_{j_0,c_0,p_0}$
            end
          end
        end
      end
    end
  end
end

Algorithm 2: Remaining Chunk Distribution (RCD) algorithm.

Initialize: $\forall j, n, c, p: \; y^{(t)}_{j,c,p} = 0, \; x^{(t)}_{j,n} = 0, \; \mathcal{X}^{(t)}_{j} = \mathcal{S}^{(t)}_{j} = 0$

while $(\#\mathcal{N} > 0)$ **do**
    $\mathcal{J}^* := \mathcal{J}$
    for $(c_0 \in \mathcal{C})$ **do**
        $j_0 = \arg\max_{j \in J^*}(\chi^{(t)}_{j,n}) \quad \forall n \in \mathcal{N}$
        $\mathcal{J}^* := \mathcal{J}^* \setminus \{j_0\}$
        for $(p_0 \in \mathcal{P})$ **do**
            while $(\varsigma^{(t)}_{j_0,c_0,p_0} > 0 \quad \&\& \quad \#\mathcal{N} > 0)$ **do**
                $n_0 = \arg\max_{n \in \mathcal{N}}(\chi^{(t)}_{j_0,n})$
                $\mathcal{N} := \mathcal{N} \setminus \{n_0\}$
                $x^{(t)}_{j_0,n_0} = 1$
                $\mathcal{X}^{(t)}_{j_0} = \mathcal{X}^{(t)}_{j_0} + \chi^{(t)}_{j_0,n_0}$
                if $(\varsigma^{(t)}_{j_0,c_0,p_0} \leq \mathcal{X}^{(t)}_{j_0} - \mathcal{S}^{(t)}_{j_0})$ **then**
                    $y^{(t)}_{j_0,c_0,p_0} = 1$
                    $\mathcal{S}^{(t)}_{j_0} = \mathcal{S}^{(t)}_{j_0} + \varsigma^{(t)}_{j_0,c_0,p_0}$
                    break;
                end
            end
        end
    end
end

Algorithm 3: The Chunk-Packet-Matching (CPM) algorithm,

Chapter 6

Performance Analysis

6.1 Simulation Model

The simulated system follows the model described in Chapter 2, exact parameter settings are given in Table 6.1. As wireless channel model we assume the open space model of ETSI C [10] ($\alpha = 2.4$, shadowing variance of 5.8 dB, exponential power delay profile with a delay spread of $\Delta\sigma = 0.15$, Jakes power spectrum with $v_{\max} = 1\frac{\mathrm{m}}{\mathrm{s}}$). We consider the three proposed scheduling approaches: (i) Optimal IP with respect to proportional fair objective (4.1) and the system and QoS constraints (3.2)–(3.4) and (4.2) (cf. Chapters 3 and 4); (ii) Relaxation approach in combination with the RTB and RCD algorithms (Algorithm 1 and 2), as well as (iii) Heuristic scheduling approach CPM (Algorithm 3) presented in Chapter 5. We assume each terminal in the cell to receive two data streams: a delay restricted VoIP stream and a best-effort FTP stream. VoIP packets arrive deterministicly according to the VoIP inter-arrival time ΔT_{VoIP}. For the FTP down-load per terminal, we assume very large file sizes such that each FTP queue always has packets to be conveyed (Table 6.1). The performance of the three variants are studied for a varying cell radius and for a varying number of terminals in the cell.

6.2 Throughput and Delay Results

The performance results in terms of average throughput per terminal and average VoIP delay per packet are shown in Figure 6.1. From the graphs it can be stated that both sub-optimal approaches do not reach optimal performance in any scenario. However, in the case of ten wireless terminals per cell, the differences are rather small over all radiuses for both sub-optimal schemes. This changes for cases of 50 or 100 wireless terminals in the cell. While the relaxation approach (LP+RTB+RCD) gets closer to the optimum (especially for small radiuses), there is a significant decrease in performance of the heuristic (CPM) approach (especially for radiuses greater than 700m, where it experiences a decrease in throughput of up to almost 50%). This is mainly due to the fact that the multi-user diversity increases with the number of users per cell and the cell radius. The CPM algorithm does not benefit from the increased diversity as much as the MILP approach and its relaxed version do, and thus is more susceptible to the increasing path loss per cell.

Regarding the delay results shown in Figure 6.1 it can be stated that for all three approaches the average delay stays in reasonable bounds. First of all notice that the relaxation approach achieves

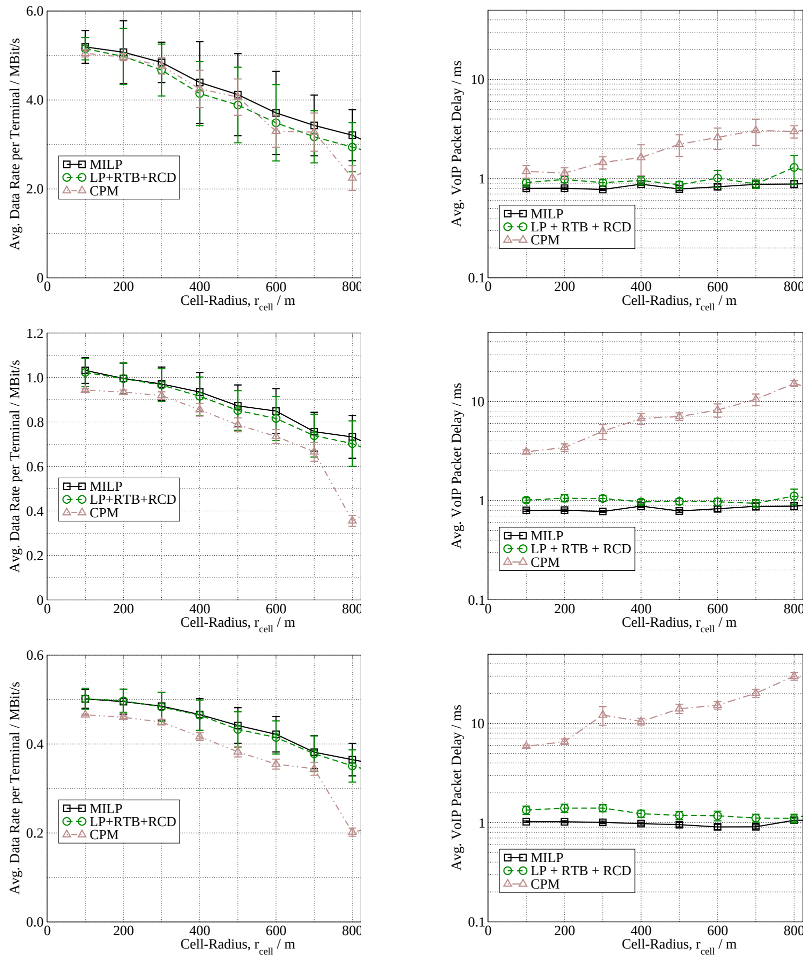

Figure 6.1: Left side: average throughput per terminal for the 10(top), 50 (middle), and 100 (bottom) terminals per cell scenario; right side:Average VoIP delay per packet for the 10, 50, and 100 terminals per cell scenario.

almost an identical delay compared to the results of the optimal approach. In contrast, the heuristic achieves significantly worse results. This is due to the fact that VoIP delay minimization is included in the objective of the optimization problem formulation, while it is not included in the CPM algorithms' terminal choice (Step 3 of the CPM algorithm). However, incorporating a better delay behavior in the heuristic would lead to a worse behavior in terms of the FTP throughput.

Parameter		**Value**
Center Frequency	f_c	5 GHz
System Bandwidth	B	20 MHz
Number of OFDM Sub-Carriers	S	1536
TTI-frame duration	T_{TTI}	1,34 ms
Chunk (Down-Link) Duration	T_{chn}	0,67 ms
OFDM-Symbol Duration	T_s	76.8 μs
Guard-interval Duration	T_g	6.95 μs
Sub-Carriers per Chunk	Φ_{sub}	16
OFDM-Symbols per Chunk	Φ_{sym}	8
Number of Chunks	N	96
Prop Fair - Evaluated TTIs	k	$J/2$
Maximum Transmission Power	P_t	1000mW
Number of Terminals	J	10, 50, 100
Cell Radius	r_{cell}	100m, 200m, . . ., 1000m
Modulation Types Applied	M	4 (BPSK, . . ., 64-QAM)
Optimization Scaling Factor	ω	$8.3\, e^{-5}$
Simulated Down-Link Phases	ξ_{dl}	1000
VoIP Packet Size	ς_{VoIP}	36 Bytes
VoIP Data Rate per Terminal	r_{VoIP}	14.4 kBit/s
VoIP Interarrival-Time	ΔT_{VoIP}	20 ms
VoIP Max. End-toEnd Delay	$\mathcal{T}_{VoIP}$	150 ms
FTP Packet Size	ς_{FTP}	500 Bytes

Table 6.1: OFDM system and data stream parameters.

6.3 Signaling Results

Using the FSSF signaling scheme (Chapter 2.4), the signaling overhead per TTI amounts to a fixed number of bits (between 672 and 960 bits, depending on the number of terminals), as shown in Table 6.2. This is a quite low number. However, the signaling overhead becomes more relevant if one considers the coding requirements as well, which easily triples this number. Hence, any decrease of the signaling information is of interest. In addition to the calculation of the overhead due to the FSSF scheme, we have also compressed the FSSF signaling information per TTI and calculated the resulting average overhead per frame. For this we have applied a compression scheme referred to as BSTW [11]. In this scheme the base station and all terminals in the cell maintain a list of assignment *words* (i.e. terminal and modulation combinations) and their (shorter binary) encodings. Newly occurring words are added to the list at the top and have shorter encodings than words at the bottom. The list ends at a point where the encodings theoretically would be longer than the original words. Using this scheme, longer runs of identical assignments can be compressed efficiently. Also, the scheme is quite simple to implement and does not require the transmission of a complete code-book, as Lempel-Ziv does for example.

As can be seen from the results, the signaling overhead can be significantly reduced by compres-

#Terminals	Radius/m	Raw	MILP	LP+RTB+RCD	CPM
10	**100**	672	125	125	227
10	**500**	672	157	119	244
10	**1000**	672	184	119	236
100	**100**	960	222	230	367
100	**500**	960	225	213	344
100	**1000**	960	274	222	266

Table 6.2: Signaling overhead in bits/frame.

sion schemes. Note in particular that the different approaches now have a different overhead. While the relaxation approach has the lowest overhead, the heuristic performs much worse. The IP optimal solution performs between both sub-optimal approaches but is closer to the relaxation approach. In general we observe that on average there is a smaller number of terminals per TTI scheduled in the sub-optimal cases than in the optimal case, which has an impact on the throughput (exploiting multi-user diversity) but also on the signaling overhead (higher overhead). Note that in the case of compression, the error correction coding requirements are higher as a bit error in the compressed signaling bit stream can lead to very negative effects at terminals.

6.4 Complexity Observations

All simulation runs have been performed in Linux user-space on a standard Pentium 4–3.2Ghz dektop machine. The MILP solutions have been obtained by the use of the commercial MILP solving software *Cplex*. Depending on the MILP instance (where one instance stands for one TTI), it took several minutes up to hours to obtain the optimal solution. To obtain the relaxed solutions, we used the free LP-solving software *Soplex*[1] [12]. The run-time results given in Table 6.3 indicate that relaxation is

# Terminals	**Radius/m:**	**100**	**500**	**1000**
10		0.050	0.044	0.070
50		0.471	0.554	0.523
100		1.000	1.531	1.281

Table 6.3: LP solving run-times in seconds.

a promising candidate for real-world implementations. This is due to the fact that specialized hard- and software can lead to a much faster execution – eventually achieving a deterministic run time of about one TTI duration. The CPM algorithm complies with the system's runtime constraints. Thus, it is applicable with comparatively cheap system components.

[1]The authors would like to thank R. Wunderling, T. Koch and the Zuse-Institute Berlin for making Soplex freely available for academic purposes.

Chapter 7

Conclusions

Finding optimal scheduling decisions for the base-station scheduler of a packet-centric OFDMA cell is a complex task. Adequate mathematical problem formulations belong to the group of np-hard MILPs. Near optimal decisions can be obtained by relaxing the MILPs to LPs and using heuristics to derive binary decisions. Alternatively, fully heuristic approaches can be formulated. In this paper we have presented a general mathematical problem formulation that can be used to benchmark the performance of sub-optimal scheduling approaches with different scheduling goals. We have used the optimal scheduling results to compare a relaxation, as well as a fully heuristic approach we derived from the optimal problem formulation. While the relaxation approach does not reach throughput optimality in any case, it requires the least signaling information. Depending on the signaling information code rate, this factor can balance the throughput shortcomings. The fully heuristic approach is worst in throughput performance and signaling costs. However, the complexity observations show that only the sub-optimal solutions are feasible for real-world implementations, where the relaxation approach requires advanced hard- and software equipment.

Bibliography

[1] C.Y. Wong, R.S. Cheng, K.B. Letaief, and R. Murch, "Multiuser OFDM with adaptive subcarrier, bit and power allocation," *IEEE Journal on Selected Areas of Comm.*, vol. 17, no. 10, pp. 1747–1758, Oct 1999.

[2] I. Kim, H. Lee, B. Kim, and Y. Lee, "On the use of linear programming for dynamic subchannel and bit allocation in multiuser OFDM," in *Proc. of the Global Telecommunications Conference*, November 2001.

[3] M. Bohge, J. Gross, and A. Wolisz, "The potential of dynamic power and sub-carrier assignments in multi-user OFDM-FDMA cells," in *Proc. of IEEE Globecom 2005*, St. Louis, MO, USA, Nov. 2005.

[4] Y. Zhang and K. Letaief, "Adaptive resource allocation and scheduling for multiuser packet-based OFDM networks," in *Proc. IEEE Int. Conf. on Communications (ICC)*, Paris, France, June 2004, pp. 2249–2953.

[5] G. Song, Y. Li, Jr. L. J. Cimini, and H. Zheng, "Joint channel-aware and queue-aware data scheduling in multiple shared wireless channels," in *Proc. of the IEEE Wireless Communications and Networking Conference 2004 (WCNC '04)*, Atlanta, GA, USA, Mar. 2004.

[6] J. Cai, X. Shen, and J.W. Mark, "Downlink resource management for packet transmission in OFDM wireless communication systems," *IEEE Trans. on Wireless Communications*, vol. 4, no. 4, pp. 1688–1703, 2005.

[7] M. Bohge, J. Gross, and A. Wolisz, "A new optimization model for dynamic power and sub-carrier allocations in packet-centric OFDMA cells," in *Proc. of the 11th International OFDM-Workshop 2006 (InOWo '06)*, Hamburg, Germany, Aug. 2006.

[8] D. Astely, E. Dahlman, P. Frenger, R. Ludwig, M. Meyer, S. Parkvall, P. Skillermark, and N. Wiberg, "A future-radio-access framework," *IEEE JSAC, Special Issue on 4G Wireless Systems*, Mar. 2006.

[9] J. Gross, H. Geerdes, H. Karl, and A. Wolisz, "Performance analysis of dynamic OFDMA systems with inband signaling," *IEEE JSAC, Special Issue on 4G Wireless Systems*, Mar. 2006.

[10] J. Medbo and P. Schramm, *Channel Models for HIPERLAN/2*, ETSI EP BRAN document 3ERI085B, March 1998.

[11] J. Bentley, D. Sleator, R. Tarjan, and V. Wei, "A locally adaptive data compression scheme," *Communication of the ACM*, vol. 29, no. 4, pp. 320–330, 1986.

[12] R. Wunderling, *Paralleler und Objektorientierter Simplex-Algorithmus*, Ph.D. thesis, Zuse-Institute Berlin (ZIB), 1996.