



收端有  $R$  根天线, 则子载波  $i$  的频域响应为

$$\mathbf{H}_{R \times T}^i = [\mathbf{H}_{r,t}^i]_{R \times T} \quad (1)$$

其中  $\mathbf{H}_{r,t}^i$  为子载波  $i$  从第  $t$  根发送天线到第  $r$  根接收天线的信道响应。可以得到:

$$\mathbf{H}_{R \times T}^i = \mathbf{U}_{R \times R} \mathbf{S}_{R \times T} \mathbf{V}_{T \times T}^H \quad (2)$$

其中  $\mathbf{U} = [U_1, U_2, \dots, U_M]$ ,  $\mathbf{V} = [V_1, V_2, \dots, V_M]$  为奇异值向量,  $\mathbf{S}_{R \times T}$  为非零矩阵, 矩阵的对角元素由  $\mathbf{H}_{R \times T}^i$  的非零特征值组成, 且  $M = \text{rank}(\mathbf{H}_{R \times T}^i)$  是  $\mathbf{H}_{R \times T}^i$  的秩,  $\mathbf{U}^H \mathbf{H}_{R \times T}^i \mathbf{V} = \mathbf{S}_{R \times T}$ , 因此

$$\lambda_l = (\mathbf{U}^H \mathbf{H}_{R \times T}^i \mathbf{V})_l = \mathbf{U}_l^H \mathbf{H}_{R \times T}^i \mathbf{V}_l \quad (3)$$

其中  $\mathbf{U}_l^H$ 、 $\mathbf{V}_l$  是特征值  $\lambda_l$  对应的奇异值向量, 因此在发送端发送符号乘上  $\mathbf{V}_l$ , 同时接收端的符号乘以  $\mathbf{U}_l^H$ , 则对应于子载波  $i$  的特征值模式  $l$ , 接收符号  $d_l^i$  就可以表示为

$$d_l^i = \mathbf{U}_l^H \mathbf{H}_{M \times N}^i \mathbf{V}_l d_i = \lambda_l d_i \quad (4)$$

其中  $d_i$  是子载波  $i$  对应的特征值模式  $l$  的调制符号。则对应接收端的信噪比  $\text{SNR}_l^i$  为

$$\text{SNR}_l^i = \frac{P_s \lambda_l^2}{N_0 B / N} = \frac{(E[d_i])^2 \lambda_l^2}{N_0 B / N} \quad (5)$$

其中  $N_0$  是噪声功率谱密度,  $B$  是信道带宽,  $N$  是系统子载波数目。假定数据的误码率为  $\text{BER}_l$ , 可以得到对于子载波  $i$  对应的模式  $l$  能够支持的比特传输速率  $r_l^i$  为

$$r_l^i = \log_2(1 + \text{SNR}_l^i / \Gamma) \quad (6)$$

其中  $\Gamma = -\ln(5\text{BER})/1.6$  是一个常数 SNR 量级(gap)。

因此对于系统来说, 子载波  $i$  能够支持的速率为

$$R_k^i = \sum_{l=1}^M r_l^i = \sum_{l=1}^M \log_2(1 + \text{SNR}_l^i / \Gamma) \quad (7)$$

### 3 调度和资源分配算法

基站侧 MAC 层的调度器对保证业务的 QoS 以及维护用户公平性、提高系统性能方面起着重要的作用。本文提出的调度器有 3 部分组成: 包分类器(PC), 缓存器管理器(BM)和包调度器(PS), 如图 2 所示。PC 根据业务的种类将数据包分别送往 BM 中相应的缓存器中, BM 则负责记录业务的服务记录, 例如包的到达时间、时延限制、数量以及每个缓存器队列第一个包(HoL)的时延等。最后 PS 根据记录断定业务的优先级, 进行相应的资源分配。

假定系统中有  $K$  个用户, 对用户  $k$  而言, 在时隙  $j$  内有

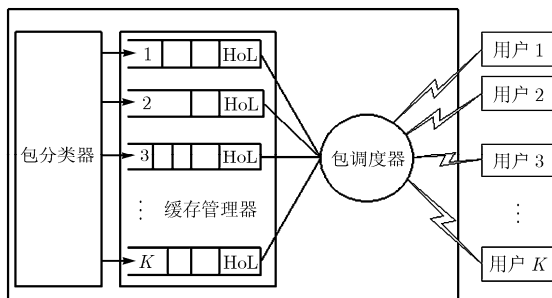


图2 调度器结构

$N_k^p$  个包需要调度和发送, 第  $i$  个包包含  $l_k^i$  个比特, 则需分配给用户  $k$  的子载波数目  $N_k$  由下式确定:

$$l_k^*(j) + \sum_{i=2}^{N_k^p} l_k^i(j) = \sum_{n=1}^{N_k} c_{n,k}(j) R_k^n(j) \quad \forall j, k \quad (8)$$

其中  $l_k^1(j)$  和  $l_k^*(j)$  分别表示调度到的第 1 个包和第 1 个包被分割后的大小, 如果第一个数据包没有被分割, 则  $l_k^*(j) = l_k^1(j)$ , 否则  $l_k^*(j) < l_k^1(j)$ ; 并且假定在一个时隙内每个子载波只能分配给一个用户,  $c_{n,k} = 1$  表示子载波  $n$  分配给用户  $k$ ,  $c_{n,k} = 0$  表示子载波  $n$  没有分配给用户  $k$ 。即

$$c_{n,k} \in \{0, 1\} \text{ 且 } \sum_{k=1}^K c_{n,k} = 1, \quad \forall n, j, k \quad (9)$$

同时, 在时隙  $j$  内用户  $k$  能发送的比特数也要受到缓存器中包数目的影响, 用  $Q_k(j)$  表示时隙  $j$  开始时缓存器的长度, 则  $Q_k(j)$  满足以下条件:

$$l_k^*(j) + \sum_{i=2}^{N_k^p} l_k^i(j) \leq Q_k(j), \quad \forall j, k \quad (10)$$

为叙述简洁, 在以后的叙述中省略时隙  $j$  的标号。同时考虑到基站总功率  $P$  一定的限制条件, 即

$$\sum_{k=1}^K \sum_{n=1}^{N_k} c_{n,k} P_{n,k} \leq P, \quad \forall j \quad (11)$$

因此基站调度的目标就是在满足业务 QoS 需求的同时最大化系统吞吐量, 即

$$\arg \max_{c_{n,k}, r_{n,k}} \sum_{k=1}^K \sum_{n=1}^{N_k} c_{n,k} r_{n,k}, \quad \forall j \quad (12)$$

为了降低计算复杂度, 采用次优解法进行求解, 对每个子载波采用等功率分配<sup>[8]</sup>; 并且为了降低实现复杂度和信令交互开销, 假定最小的传输单元为时频块, 由时域上的一个时隙(包含  $n_t$  个连续的 OFDM 符号)和频域上的一个子带(包含  $n_f$  个子载波)组成, 则对于用户  $k$  而言, 其分配到的时频块  $n$  的比特传输速率  $R_{n',k}$  为

$$R_{n',k} = M \times n_f \times n_t \times \log_2(1 + \overline{\text{SNR}}_k / \Gamma) \quad (13)$$

其中  $\overline{\text{SNR}}_k$  是对用户  $k$  而言该时频块中各子载波的平均信噪比。

本文提出的算法分 4 步, 每步都要进行包调度和子带的分配过程。首先要定义如下参数:  $w_k^i$  为到时刻  $j$  为止, 用户  $k$  的队列中第  $i$  个包的时延。  $e_k$  为用户  $k$  的数据包队列中第 1 个包距离超过时延限制的时间(以时隙为单位), 该参数数值由包允许的最大时延参数  $d_{\max}^k$  以及  $w_k^1$  决定, 即  $e_k = d_{\max}^k - w_k^1$ 。  $r_{n',k}^{\text{effec}}$  为时隙  $j$  内分配给用户  $k$  的子带  $n'$  的有效传输速率, 即  $r_{n',k}^{\text{effec}} = \min\{R_{n',k}^*, Q_k\}$ 。  $\Omega$  为系统内可用子带的集合, 其每个时隙开始的初始值为  $\Omega = \{1, 2, \dots, N_b\}$ ,  $N_b$  为系统最大可用子带数目。  $\Phi$  为一个调度周期内所有用户的集合。

$\text{fir\_users}$  为第 1 类用户, 定义为最紧急需要调度的用户, 即在时隙  $j$  开始时  $e_k(j) = 1$  的用户集合。

$\text{sec\_users}$  为第 2 类用户, 即平均包时延  $\bar{w}_k(j)$  满足

$\frac{\bar{w}_k(j)}{d_{\max}^k} w \geq \delta$  且  $Q_k(j) > 0$  的用户集合, 其中  $\delta \in (0,1)$  是控制阈值。

thi\_users 为第 3 类用户, 即条件满足  $R_k^{\text{req}} - \bar{R}_k(j-1) > 0$  并且  $Q_k(j) > 0$  的用户集合, 其中  $R_k^{\text{req}}$  和  $\bar{R}_k(j-1)$  分别是用户  $k$  的最小速率要求和时隙  $(j-1)$  的实际平均速率。

以上 3 类用户 fir\_users, sec\_users, thi\_users 是根据每个调度周期内调度的优先级顺序所定义。

$\Phi_u$  为以上用户总集合  $\Phi$  中除去已经调度用户的集合。

步骤 1 首先确定 fir\_users 的集合, 并进行资源分配。fir\_users 的用户按照以下公式以降序调度:

$$p_u^k = \exp\left(\frac{\text{PLR}_k}{\text{PLR}_k^{\text{req}}}\right) + \frac{l_k^*}{l_k} \cdot I_{\{l_k^* < l_k\}} \quad (14)$$

其中  $\text{PLR}_k$  表示截止到时隙  $j$  用户  $k$  的丢包率,  $\text{PLR}_k^{\text{req}}$  表示用户  $k$  所允许的最大丢包率。当  $\alpha$  为真时, 指示函数  $I_{\{\alpha\}} = 1$ , 否则  $I_{\{\alpha\}} = 0$ 。

通过式(12)可以看到 fir\_users 的优先级由两个因素所决定: 首先, 具有较高丢包率的用户将获得较高优先级; 其次, 队列中第 1 个包被分割的用户获得较高优先级, 这是因为如果被分割的数据包如果不能及时得到传输, 则该包就要被抛弃。为降低实时业务的丢包率, 第 1 个因素具有较高权重。下面进行相应的子带分配。

对于具有最高优先级的用户  $k^*$ , 首先断定系统的可用资源能否完整地传输 HoL 包。如果不能, 则传输该数据包会引起资源的浪费, 因为根据 fir\_users 的定义可知, 本次调度周期后, 该包已经超过其最大时延, 就要被抛弃,  $\Phi_{\text{fir\_users}}$  也应该删掉该用户; 反之, 则根据以下公式计算  $\Omega$  中子带的优先级  $p_s^n$ :

$$p_s^n = \left( \frac{\max\{r_{n,k}^{\text{effec}} \mid \forall n \in \Omega, k \in \Phi\} - r_{n,k^*}^{\text{effec}}}{r_{n,k^*}^{\text{effec}}} \right)^{-1} \quad (15)$$

式(13)根据子带的有效传输能力定义了调度准则, 子带有效传输能力越大, 则优先级越高。给用户  $k^*$  分配子带后, 更新  $Q_k^*$  和  $\Omega$ , 并重新计算剩余子带的优先级, 直到分配给用户  $k^*$  的资源可以完整传输 HoL 包。对用户  $k^*$  的调度之后, 更新  $\Phi_{\text{fir\_users}}$ , 并继续调度优先级最高的其他用户, 直到集合  $\Phi_{\text{fir\_users}}$  为空。

步骤 2 如果  $\Omega$  仍有可用子带, 将确定 sec\_users 的用户, 并为其分配资源。sec\_users 的用户按照以下公式以降序调度:

$$p_u^k = \frac{\bar{w}_k}{d_{\max}^k} \quad (16)$$

子带分配的步骤与步骤 1 的资源分配类似, 按照式(13)计算可用子带的优先级, 并将其分配给优先级最高的用户  $k^*$ , 更新  $Q_k^*(j)$  和  $\Omega$ 。但是该步骤与步骤 1 不同的是, 每次只给用户分配一个子带, 并根据下式更新 sec\_users 的平均包时延

$$\bar{w}_k(j) = \frac{\sum_{i=1}^{M_k^*} w_k^i}{M_k^*} \quad (17)$$

其中  $M_k^*$  是用户  $k^*$  队列中的包数。

计算完优先级后, 重复上述过程直至集合  $\Phi_{\text{sec\_users}}$  或  $\Omega$  为空。

步骤 3 若集合  $\Omega$  仍有可用子带, 将继续确定 thi\_users 的用户, 并为其分配资源。对于 thi\_users 用户, 除了考虑它们的最低速率要求, 还要考虑信道效率。定义用户  $k$  的信道效率  $\eta_k$  如下:

$$\eta_k = \frac{Q_k}{\sum_{n \in \Omega} R_{n,k} / \|\Omega\|} \quad (18)$$

其中  $\|\Omega\|$  表示集合  $\Omega$  中用户的数目。同时, 为了在保证每个用户的最低速率要求的同时最大化系统传输效率, 用户  $k$  的优先级以及子带  $n$  分配过程如下式以降序确定:

$$p_u^k = \exp\left(\frac{R_k^{\text{req}} - \bar{R}_k}{\bar{R}_k} \cdot \eta_k\right) \quad (19)$$

其中

$$\bar{R}_k(j) = \left(1 - \frac{1}{T_c}\right) \cdot \bar{R}_k(j-1) + \frac{1}{T_c} \cdot r_{n^*,k^*}(j) \quad (20)$$

$T_c$  是时间窗口<sup>[9]</sup>。

找出具有最大  $p_u^k$  对应的子带  $n^*$  和用户  $k^*$ , 分配过程如步骤 2, 并更新  $\bar{R}_k(j)$  和  $p_u^k$ 。重复上述过程直至  $\|\Phi_{\text{thi\_users}}\| = 0$  或  $\|\Omega\| = 0$ 。

步骤 4 以上步骤考虑了业务的 QoS 需求, 如果  $\Omega$  中仍有可用子带, 则该步骤中就以增大系统吞吐量为目标。将剩余的子带逐一分配给具有最大传输效率的用户  $k^*$  以提高系统的吞吐量:

$$k^* = \arg \max_{k \in \Phi} \{r_{n,k}^{\text{effec}}\} \quad (21)$$

## 4 算法性能仿真

将本文提出的算法与文献[10]中的算法(以后称为 XL 算法)分别应用到 MIMO 系统中进行了在系统吞吐量、平均包时延等方面的性能仿真和比较。文献[10]提出的算法给予实时业务最高的优先级, 非实时业务最低的优先级, 系统在全部分满足实时业务的传输需求后才考虑对非实时业务进行调度, 因此该算法虽然很好的满足实时业务的需求, 但是不能最大化系统吞吐量。性能仿真中, 假设系统中实时业务以 VoIP 业务为代表, 非实时(NRT)业务以 WWW 业务为代表<sup>[6]</sup>。系统的基本参数如表 1 所示<sup>[11]</sup>。基站半径为 1km, 用户在小区内均匀分布; 参数  $\delta$  设为 0.8, 发送天线和接收天线数目都为 2。

图 3, 图 4 示出了在 VoIP 业务和数据业务混合情况下, 提出的调度算法与 LX 算法在系统的总吞吐量、数据业务吞吐量方面的性能比较。从图 3 可以看出, 当系统中的数据用户数量少时, 本文提出的算法与 LX 算法在系统吞吐量方面性能相近; 但随着数据用户的增多, 两种算法性能差距逐渐拉大, 前者吞吐量能够超过  $(5 \times 10^7)$  bit, 而后者仅能达到

表1 系统参数

参数	数值
系统带宽	5MHz
子载波数目	512
子带数目	32
帧长	10ms
时隙长度	1ms
每个时隙的OFDMA符号数目	10
MIMO信道模型	3GPP 25.996
语音业务的最大丢包率	$10^{-2}$
语音业务最大的包时延	20ms
语音业务的速率要求	24kbps
数据业务最小的速率要求	100kbps
BS总发送功率	10W

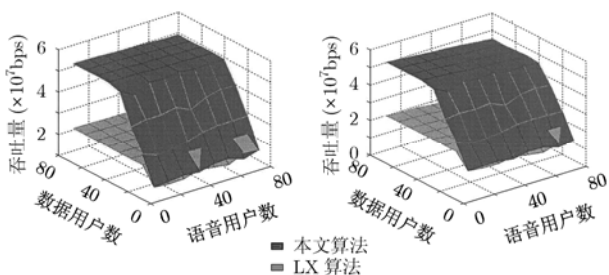


图3 系统的总吞吐量比较 图4 数据业务的总吞吐量比较

( $2 \times 10^7$ )bit, 这种性能差距从图4中也可以看出来, 两者对实时业务的吞吐量取得相近的情况下, 后者对数据用户的支持远远小于前者, 不能够高效地支持数据用户业务的传输。

图5示出了两种算法对于实时业务在时延方面的性能比较。可以看出, LX算法在保证实时用户的实时性方面明显优于本文的算法, 能够对实时业务的平均时延保证在2ms左右; 虽然本文的算法随着数据用户的增多而对实时用户的时延影响逐渐变大, 这是因为该算法在系统吞吐量方面的优越性能是以牺牲实时用户的部分时延特性为前提的, 但是该算法仍然能保证实时业务的时延在14ms以下, 能够满足实时业务的QoS需求。

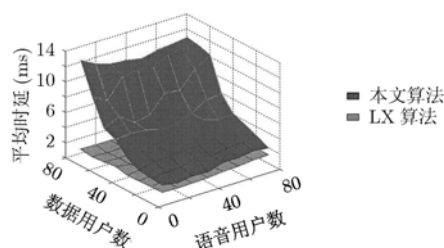


图5 VoIP业务包的平均时延比较

## 5 结束语

本文提出了一种适用于MIMO-OFDMA系统的调度和资源分配算法, 该算法不仅利用了先进的物理层技术, 而且从MAC层考虑了不同种业务的QoS需求、信道信息、用户公平性等因素, 能够满足实时业务和非实时业务的传输要

求, 尤其是该算法在满足实时业务的时延性要求的前提下更加有效地支持数据业务的传输, 提高了系统的吞吐量和总体性能, 适合未来移动通信系统发展的趋势和要求。

## 参考文献

- [1] 佟学俭, 罗涛. OFDM移动通信技术原理与应用. 北京: 人民邮电出版社, 2003年6月: 188-210.
- [2] Zhang Ping *et al.* A visions from the Future: Beyond 3G TDD. *IEEE Commun. Mag.*, 2005, 43(1): 38-44.
- [3] 陶小峰, 印海莹, 徐晋, 刘宝玲, 张平. 广义MIMO通用信道矩阵建模. 北京邮电大学学报, 2005, 28(5): 47-50.
- [4] Kivanc D, Li G, and Liu H. Computationally efficient bandwidth allocation and power control for OFDMA. *IEEE Trans. on Wireless Commun.*, 2003, 2(6): 1150-1158.
- [5] Zhang Guodong. Subcarrier and bit allocation for real-time services in multiuser OFDM systems, IEEE International Conference on Communications, June 20-24, 2004, Vol.5: 2985-2989.
- [6] Kim Dong-Hoi and Ryu Byung-Han. Packet scheduling algorithm considering a minimum bit rate for non-real-time traffic in an OFDMA/FDD-based mobile internet access system. *ETRI Journal*, 2004, 26(1): 48-52.
- [7] Liu Guangyi, Liu Xiantao, and Zhang Ping. QoS oriented dynamical resource allocation for eigen beamforming MIMO OFDM, VTC-2005-Fall, Dallas. 25-28 Sept., 2005, Vol.3: 1450-1454.
- [8] Jang Jiho and Lee K B. Transmit power adaptation for multiuser OFDM systems. *IEEE J. Select. Areas Communication.*, 1999, 17(1): 1747-1758.
- [9] lalali A, Padovani R, and Pankaj R. Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system. VTC 2000-Spring, Tokyo, 2000, Vol.3: 1854-1858.
- [10] Xiao Liang, Zhou Shidong, and Yao Yan. QoS-oriented scheduling algorithm for mobile multimedia in OFDM. 2003. PIMRC 2003. 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 7-10 Sept. 2003, Vol.1: 545-549.
- [11] 3GPP TR25.996. Spatial channel model for Multiple Input Multiple Output (MIMO) simulations (Release 6). Sept. 2003.

高有军: 男, 1980年生, 博士生, 研究方向为调度和无线资源管理、MAC协议研究。

徐海博: 男, 1981年生, 博士生, 研究方向为MAC协议、联合资源管理技术研究。

田辉: 女, 1961年生, 教授, 博士生导师, 研究方向为B3G关键技术研究、MAC协议、调度和无线资源管理技术研究。

张平: 男, 1959年生, 教授, 博士生导师, 研究方向为B3G系统设计及其关键技术研究。