

## CDMA2000 系统反向链路中一种改进的速率控制算法

刘兵章 张欣 杨大成  
(北京邮电大学电信工程学院 北京 100876)

**摘要:** 该文针对 CDMA 系统反向链路中两种传统的速率分配算法,即调度算法(SCH)和专用速率控制算法(DRC)的优缺点,提出了一种改进的 DRC 算法:多步上升速率控制算法(MUDRC)。新算法在保留 DRC 算法优点的同时,克服了 DRC 算法慢启动的缺点,使得移动台(MS)在某些情况下可以一次提高多个速率等级。仿真结果表明,MUDRC 算法可以在有效利用系统资源的同时,一定程度上提高用户的时延性能。与传统的 DRC 算法相比,MUDRC 算法的前反向开销没有增加,这些开销远小于调度算法。该算法可以用于其他使用速率控制算法的通信系统中。

**关键词:** 调度; 速率控制; 时延; 开销

中图分类号: TN914.53

文献标识码: A

文章编号: 1009-5896(2007)12-2952-04

## A Modified Rate Control Algorithm for the Reverse Link in CDMA2000 Systems

Liu Bing-zhang Zhang Xin Yang Da-cheng

(School of Telecommunications, Beijing University of Posts and Telecommunications, Beijing 100876, China)

**Abstract:** This paper focuses on the rate allocation algorithms on the reverse link (RL) of CDMA2000 system. Two conventional algorithms, the SCHEDULE (SCH) algorithm and the Dedicated Rate-Controlled (DRC) algorithm are compared and analyzed. Based on their *pros* and *cons*, a modified version of DRC algorithm is proposed, which is called Multi-step-Up DRC (MUDRC). While holding the advantages of DRC, the proposed algorithm can combat for the inherent slow ramp-up nature of DRC and enable a mobile to increase its rate by multiple steps when needed. Simulation results show that the proposed algorithm can efficiently utilize the radio resources and improve the user's performance of delay. In addition, the performance gain is achieved by using the same amount of signaling overhead as that of DRC, which is much less than that of SCH. The proposed scheme can be generalized to other similar communication systems.

**Key words:** Schedule; Rate control; Delay; Overhead

### 1 引言

CDMA2000 Revision D物理层标准<sup>[1]</sup>通过引入反向分组数据信道 (R-PDCH) 对CDMA2000 1xEV-DV系统的反向链路进行了增强。本文主要研究了CDMA2000 1xEV-DV系统中的反向的速率分配算法。在CDMA2000 1xEV-DV系统中,移动台(MS)在R-PDCH信道上的传输速率受基站端(BS)的控制。反向链路中的速率分配方案大致可以分为两类:调度(SCH)算法和速率控制(RC)算法。调度算法的基本思想是每次选择部分用户传输数据。由于反向链路中单用户申请的速率不高,所以可以同时选择多个用户,但一般不是所有用户。调度算法可以使得系统获得多用户分集增益。速率控制算法可以使所有需要传输数据的用户同时进行数据的传输,它们的速率根据接收到的前向控制命令(RCB)进行调整。速率控制算法可进一步分为专用速率控制(DRC)和公共速率控制(CRC)算法。其中,前者是通过为每个移动台发送单独的速率控制命令实现,而后者是为所有的移动台广播一个公

共的速率控制命令实现的<sup>[2]</sup>。本文只讨论DRC算法。

与SCH算法比较而言,DRC算法具有信令开销小,速率变化慢的特点,因此,采用这种算法可以达到比较高的系统效率。但是,在DRC算法中,移动台的传输速率至多上升一个速率等级。因此,在某个时刻,如果移动台需要以比当前速率大得多的速率传输时,这个速率的调整过程经历的时间就比较长。在突发型的业务中,这个特点表现为慢启动的特性,它会影响数据包的传输时延。针对上述问题,文献[3]提出了一种SCH和DRC的混合算法,该算法在使得前向信令开销尽量小的情况下,可以保证系统的吞吐量并减小分组数据包的传输时延。本文提出了一种改进的DRC算法,这种算法使用比文献[3]中小信令开销可以实现较好的性能。本文通过系统级仿真对算法的性能进行评估。

本文的结构如下:第2节详细说明了传统的速率分配算法以及本文提出的改进算法,第3节给出了系统级仿真模型和仿真结果分析,最后在第4节对全文做出了总结。

### 2 SCH, DRC, MUDRC 算法

本小节描述传统的SCH和DRC算法,并提出一种改进

的 DRC 算法; MUDRC 算法。

速率分配实际上是在移动台之间分配资源的过程。由于 CDMA 系统是干扰受限的系统, 因此, 速率控制应该最终使得系统的干扰控制在一定的范围之内。通常用系统基站端的 RoT(Rise over Thermal)值来表征系统的干扰程度。基站端的 RoT 值是基站总的接收功率与热噪声功率的比值, 可通过式(1)计算得到:

$$Z = \left( \sum_{i=1}^N E_i + E_n \right) / E_n \quad (1)$$

其中  $E_i$  表示基站接收到的 MS*i* 功率,  $E_n$  是基站端的热噪声功率。

在任何一种速率分配算法中, 必须使得系统满足 RoT 的超标率标准, 即控制 RoT 大于某个特定门限值的概率在一定范围之内, 例如, 控制 RoT 值超过 7dB 的概率小于 1% ( $\Pr[\text{RoT} > 7\text{dB}] < 1\%$ )。

### 2.1 调度(SCH)算法

SCH 算法中, 在每个调度周期内, 系统选择部分用户传输数据, 进而可以得到用户分集增益。每个 MS 通过反向请求信道(R-REQCH)将它当前的状态上报给 BS 端, 这些状态可以看作是 MS 根据当前的数据量和功率情况而确定的一个请求速率。通过执行 SCH 算法, BS 确定每个 MS 的传输速率并通过前向许可信道(F-GCH)将确定的速率值下发给每个 MS。反向链路中由于单用户的申请速率不高以及功率有限, 因此, 可以允许多个 MS 同时传输数据, 但可以不是全部的 MS。有关调度算法更详细的过程可以参考文献[4, 5]。

### 2.2 专用速率控制(DRC)算法

在 DRC 算法中, 所有需要传输数据的移动台同时进行传输。移动台通过在反向信道上发送 1bit 的 MSIB(MS Indicator Bit)信息来向基站上报它是否可以在当前速率的基础上进一步提高速率。MSIB 的设置是由移动台当前需要传输的数据量, 移动台的最大可用发射功率以及它当前的传输速率决定的。通过执行 DRC 算法, 基站得到每个移动台对应的速率控制命令(RCB)。然后, 基站在前向速率控制信道(F-RCCH)上发送速率控制命令。速率控制命令共有 3 种, “up”, “down”, “hold” 信息分别用来通知移动台上升一个速率等级, 下降一个速率等级或者保持当前的速率不变。有关专用速率控制算法更详细的过程可以参考文献[2, 6]。

### 2.3 改进的 DRC 算法——多步上升 DRC(MUDRC)算法

在传统的 DRC 算法中, 移动台接收到的 RCB 消息是它向基站发送的 MSIB 消息的反馈信息。相对于发送的 MSIB 消息而言, 这个反馈需经过一段时延才能被接收到。移动台根据接收到的 RCB 消息来调整传输速率。需要说明的是, 移动台最后确定的实际传输速率不仅受 RCB 的控制, 而且还受到其他速率限制, 这些速率包括由移动台当前需要传输的数据量和最大发射功率决定的最大传输速率( $R_{\max}$ ), 以及系统为保证移动台的最低吞吐量而设定的自动传输速率( $R_{\text{auto}}$ )。移动

台通过比较当前时刻的  $R_{\max}$  和用户的实际传输速率  $R_{\text{tran}}$ , 得到下一个需要发送的 MSIB 的值(0 或者 1), 并将它发送到 BS 端。移动台决定 MSIB 消息及最终传输速率的算法由式(2)–式(4)描述。

$$R_{\text{temp}} = R_{\text{last}} + F_{\text{step}}(\text{RCB}) \quad (2)$$

$$R_{\text{trans}} = \min(R_{\max}, \max(R_{\text{auto}}, \min(R_{\text{temp}}, R_{\max}))) \quad (3)$$

$$\text{MSIB} = \text{bool}(R_{\max} > R_{\text{trans}}) \quad (4)$$

在以上几个式子中,  $R_{\text{last}}$  表示移动台在确定新的传输速率之前使用的速率等级,  $R_{\text{tran}}$  是移动台最终确定的实际传输速率,  $R_{\text{temp}}$  是一个临时使用的中间变量。  $F_{\text{step}}$  是以 RCB 为参数的一个函数, 它表示移动台可以变化的速率步长, 如式(5)所示:

$$F_{\text{step}}(\text{RCB}) = \begin{cases} -1, & (\text{RCB} = \text{down}) \\ 0, & (\text{RCB} = \text{hold}) \\ 1, & (\text{RCB} = \text{up}) \end{cases} \quad (5)$$

由上面的说明可以看出, 在传统 DRC 算法中, 单步上升的特点是引起慢启动的主要原因。文献[3]中提出了一种 SCH 算法和 DRC 算法的混合算法。在常规情况下, 它使用 DRC 算法。在新的突发数据包到达的开始阶段, 使用 SCH 算法, 使得传输速率能在短时间内上升到合适的等级上, 然后再采用 DRC 算法。由于在反向业务信道中采用了自动混合重传(HARQ)技术, 几路不同的子信道传输速率可能差别很大, 这种混合控制算法不容易控制, 使得何时切换到 DRC 算法不明确。在本文提出的算法中, 不采用调度算法, 而只是对传统 DRC 算法进行改进, 算法简单, 便于操作。

在本文提出的算法中, 通过一个滑动窗口来记录当前时刻之前的一段时间内  $R_{\max}$  的值, 移动台通过跟踪一段时间内  $R_{\max}$  的变化情况来决定是否对速率进行多步调整。

图 1 是滑动窗口的示意图。其中, 需要定义两个变量: (1)  $T_d$ , 即从移动台发送 MSIB 消息到接收到相应的 RCB 命令的时间; (2)  $T_w$ , 即滑动窗口的时间长度。两个变量都是以数据帧长为单位, 前者表示了移动台的请求反馈时延, 后者表示了算法跟踪记录  $R_{\max}$  的时间长度。每个移动台可以有不同  $T_d$  和  $T_w$  的值。在图 1 中, 我们将  $T_d$  和  $T_w$  分别设为 3 帧长和 4 帧长来说明我们的算法。假设在第  $i$  帧时刻, 某个移动台需要确定一个新的传输速率。  $R_{\max\text{-ave}}$  表示滑动窗口内  $R_{\max}$  的平均值, 根据式(8)计算得到。  $R_{\max-i}$  表示当前第  $i$  帧的  $R_{\max}$  值。如果  $R_{\max-i}$  大于  $R_{\max\text{-ave}}$  的差值超过了一定的门限值, 并且当前接收到的 RCB 为 “up”, 那么移动台的速率就可以上升多步, 而不像 DRC 算法中那样只上升一步。算法的具体执行如式(6)–式(8)所示。

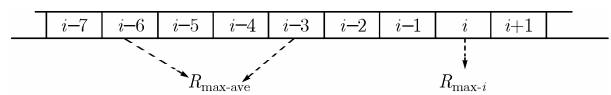


图 1 记录  $R_{\max}$  的滑动窗口示意图

$$R_{\max-ave} = \left( \sum_{n=i-T_d-T_w+1}^{i-T_d} R_{\max-n} \right) / T_w \quad (6)$$

$$M_i = \text{bool}(R_{\max-i} - R_{\max-ave} > R_{th}) \quad (7)$$

$$F_{\text{step}}(\text{RCB}) = \begin{cases} -1, & (\text{RCB} = \text{down}) \\ 0, & (\text{RCB} = \text{hold}) \\ 1 + C \times M_i \times (R_{\max-i} - R_{\max-ave}), & (\text{RCB} = \text{up}) \end{cases} \quad (8)$$

其中  $R_{\max-n}$  表示第  $n$  帧的  $R_{\max}$  值。Bool 操作是一个判断条件是否满足的操作, 当条件满足时, 返回 1, 否则, 返回 0。 $M_i$  是取值为 0 或者 1, 用来标志移动台是否需要多步速率上升。 $R_{th}$  是一个判决门限。 $C$  是一个调整系数, 决定了移动台的速率上升的步长。当  $C$  为 0 时, 算法回归到 DRC 算法。

### 3 仿真模型及结果分析

#### 3.1 仿真条件

本文将 SCH, DRC, MUDRC 3 种算法在 CDMA2000 1xEV-DV 仿真平台上进行了仿真。假设在一次仿真的过程中, 系统的用户数保持不变, 移动台在服务区中均匀分布, 详细的仿真配置可参见文献[7]。与文献[3]类似, 仿真使用了两种业务模型: FB(Full Buffer) 和 FTP(File Transfer Protocol) [8]。其中, 在 FB 模型下衡量算法的扇区吞吐量性能上限。在另外一组仿真中, 使用 FTP 业务模型来衡量数据包的传输时延性能。

#### 3.2 仿真结果及分析

表 1 给出了 FB 业务模型下系统的平均扇区吞吐量仿真结果。仿真中设置每小区有 10 个移动台, 并且在 RoT 超标率标准大致相同 ( $\text{Pr}[\text{RoT} > 7\text{dB}] < 1\%$ ) 的条件下得到仿真结果。仿真结果表明, DRC 算法的吞吐量要比 SCH 算法的吞吐量大。这种性能差异的原因主要有两个: (1) 专用速率控制只允许移动台的速率变化至多一个速率等级, 因此系统的稳定性比较好, 即在同样的 RoT 超标率要求下, 例如  $\text{Pr}[\text{RoT} > 7\text{dB}] < 1\%$ , DRC 算法可以使得系统稳定状态的 RoT 均值比较大, 系统可以承担较多的负载; (2) DRC 算法的反向开销小于 SCH 算法<sup>[1]</sup>, 可以节省一部分系统资源。

由表 1 还可以看出, MUDRC 与 DRC 的吞吐量性能相当, 且明显大于 SCH 的吞吐量性能。这是由于 FB 业务模型下不会引起  $R_{\max}$  的有比较大的变化。因此, MUDRC 算法保留了 DRC 算法中吞吐量大的优点。

表 1 FB 业务模型下的吞吐量仿真结果

控制算法	平均扇区吞吐量(bps)	RoT 标准( $\text{Pr}[\text{RoT} > 7\text{dB}]$ )
SCH 算法	530710	0.010031
DRC 算法	661907	0.010593
MUDRC 算法	657612	0.010038

在考察各种算法下的数据包传输时延性能时, 仿真设置每扇区 5 个 FTP 用户, 用户每隔一段时间需要上传一个数据包。每个用户的数据包到达过程是相同的, 数据包到达时间间隔服从 Poisson 分布, 并且一次仿真过程中数据包的大小是固定的。本文进行了多次仿真, 以考察不同数据包大小及不同平均时间间隔下的文件传输时延性能。在 5 用户/扇区条件下, 数据包平均到达时间间隔 2.5s, 3.33s, 5s, 10s 分别对应着 2 packet/(sector · s), 1.5packet/(sector · s), 1 packet/(sector · s) and 0.5 packet/(sector · s) 的扇区数据包到达率。仿真中, 参数  $T_d$ ,  $T_w$ ,  $T_j$ ,  $C$  的值分别设为 3 帧, 4 帧, 2 和 0.5。需要注意的是,  $R_{th}$  和  $C$  的取值大小决定了算法的性能, 本文的取值可以使得算法在多步上升和单步上升操作之间进行合理的切换。

在有新的突发数据包到达时, 调度算法和专用速率控制算法的不同如图 2 所示。图 2 是引用文献[3]中的图例。图 3, 图 4 给出了仿真结果。图 3 是不同数据包大小以及不同到达率下各算法的时延性能。其中, 数据包传输时延是指从数据包的数据开始传输到它的所有数据被基站接收到所经历的时间。

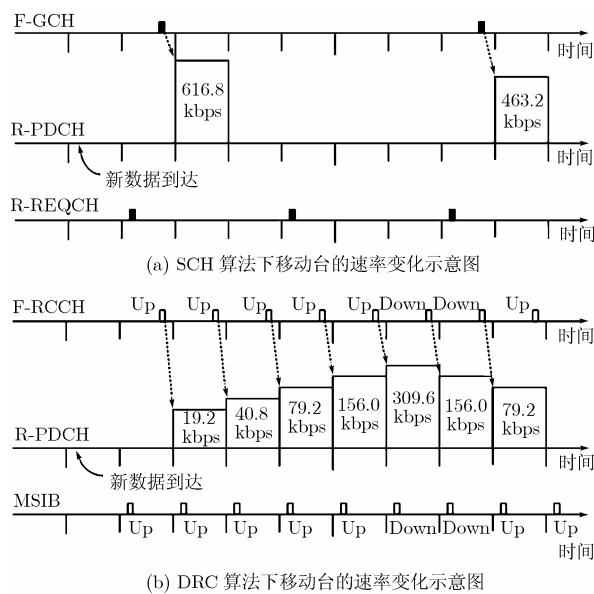


图 2 SCH 和 DRC 算法下的速率变化示意图

图 3 中的结果 SCH 算法的时延性能要优于 DRC 算法, 这是由 DRC 算法本身固有的慢启动特性引起的。慢启动特性使得空闲状态下的移动台在突然有数据包要传输时, 达到它的稳定传输速率需要比较长的时间。而 SCH 算法中, 移动台可以快速调整到一个合适的速率等级上(图 2)。另外, MUDRC 算法的数据包传输时延与 SCH 算法的时延比较接近, 明显小于 DRC 的数据包传输时延, 这说明本文提出的改进算法可以明显地改进原有 DRC 算法的慢启动特性, 进而可以提高用户的时延性能。

RoT 是扇区负载情况的一个直接的反映, 因此系统 RoT 值与系统的稳定性有着直接的联系。本文通过 RoT 标准差

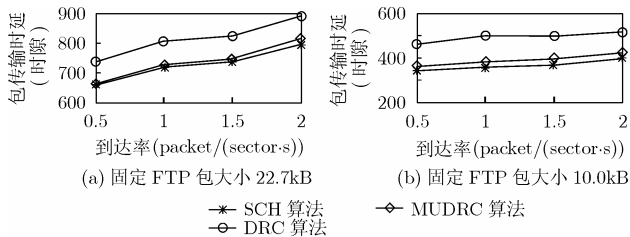


图 3 FTP 业务模型下不同算法的数据包传输时延性能比较

和超标率标准来衡量扇区的稳定性。由于本组仿真中采用的是 FTP 业务模型, 系统工作在轻负载状态下, 因此 RoT 的超标率远小于 1%, 但通过观察它在不同算法下的相对大小依然可以对系统的稳定性作比较。从图 4 可以看出, 在每个数据包的大小固定为 10kB 时, MUDRC 算法的 RoT 性能(标准差和超标率)介于 SCH 算法和 DRC 算法之间。MUDRC 算法在某些时刻会将速率提高多个速率等级, 与 DRC 算法相比, 这就一定程度上影响系统的稳定性, 但在我们设置的参数下, 它依然比 SCH 算法下的系统稳定性好。当数据包大小为 22.7kB(文献[8]中的典型值), 仿真得到的结果类似, 限于篇幅, 不再赘述。因此, MUDRC 算法可以在系统稳定性比 SCH 好的情况下达到与 SCH 算法接近的时延性能。

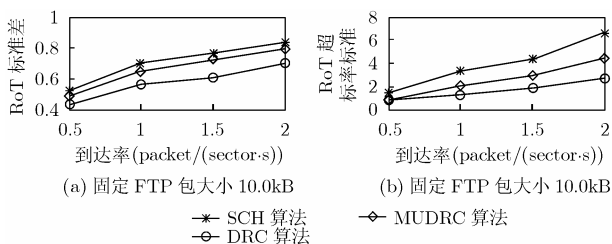


图 4 不同算法下的 RoT 性能比较(数据包大小为 10.0kB)

另外, 由文献[1]可以知道, 调度算法的前反向开销都比专用速率控制算法大。从 2.3 小节可以看出, 与 DRC 算法相比, MUDRC 算法并未增加任何前反向信令开销。另外, MUDRC 算法中的前反向信令开销比文献[3]中的信令开销也要小。

从以上仿真结果和分析可以看出, MUDRC 算法具有以下特点: (1) 优于 DRC 算法的时延性能; (2) 吞吐量性能优于 SCH 算法, 与 DRC 算法基本相同; (3) 与 DRC 相同的前反向信令开销。

#### 4 结束语

本文给出了 CDMA2000 系统中传统的调度(SCH)算法和专用速率控制(DRC)算法的性能比较, 基于这两种算法的优缺点, 提出了一种改进的专用速率控制算法: 多步上升速率控制算法(MUDRC)。仿真结果表明, 在不增加前反向信

令开销的前提下, MUDRC 算法保留了传统 DRC 算法吞吐量大的优点, 并一定程度上克服了 DRC 算法慢启动的问题。

由于移动台的数据传输速率情况受信道条件的影响, 而且本文提出的算法需要设置多个参数的值。如何设置算法中的各个参数值, 使得传输速率既可以适应信道条件的变化, 又可以适应各种不同的业务模型, 以及设计更好的速率控制算法是以后研究的重点。

#### 参考文献

- [1] 3GPP2, cdma2000 standard for spread spectrum systems, Revision D, Feb.2004.
- [2] Chung Young-uk, Koo Changhoi, Bae Beomsik, Lee Hyeon-oo, and Cho Dong-Ho. An efficient reverse link data rate control scheme for 1xEV-DV system. Proceedings of the Vehicular Technology Conference, VTC2001-Spring, Atlantic, USA, vol. 2: 820-823.
- [3] Kwon Hwanjoon (Eddy), Kim Younsun, Han Jin-kyu, and Kim Donghee. An efficient radio resource management technique for the reverse link in CDMA2000 1xEV-DV. Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC2005, New Orleans, LA, USA, 2005, vol. 1: 364-368.
- [4] Damnjanovic Jelena, Jain Avinash, Chen Tao, and Sarkar Sandip. Scheduling the CDMA2000 reverse link. Proceedings of the Vehicular Technology Conference, VTC2002-Fall, Vancouver, Canada, 2002, vol. 1: 386-390.
- [5] Elliott Robert and Krzymien Witold. Scheduling algorithms for the CDMA2000 packet data evolution. Proceedings of the Vehicular Technology Conference, VTC2002-Fall, Vancouver, Canada, 2002, vol. 1: 304-310.
- [6] Jain Avinash and Lundby Stein. Reverse-link control mechanisms in CDMA2000 revision D: performance evaluation. Proceedings of the Personal, Indoor and Mobile Radio Communications PIMRC2004, Barcelona, Spain, 2004 vol. 3: 2040-2045.
- [7] 3GPP2 C30-20030414-064R1, IS-2000 release C simulation configuration specification rev.11[S]. 2003.
- [8] 3GPP2 TSG-C Working Group 3, 1xEV-DV Evaluation Methodology V0.4, 2004.

刘兵章: 男, 1982 年生, 硕士, 研究方向为 CDMA 系统无线资源管理算法。

张欣: 男, 1975 年生, 副教授, 硕士生导师, 研究方向为移动通信。

杨大成: 男, 1951 年生, 教授, 博士生导师, 研究方向为移动通信。