

## 基于动态分组的 M2M 上行优先级调度算法

金叶奇<sup>1,2</sup>, 徐佑宇<sup>1,2</sup>, 郑 敏<sup>1</sup>, 谭 冲<sup>1</sup>, 王 虹<sup>1,2</sup>

(1. 中国科学院上海微系统与信息技术研究所, 上海 200050; 2. 中国科学院大学, 北京 100049)

**摘 要:** 以延迟容忍剩余时长为度量依据, 提出一种对业务进行动态分组的上行优先级调度算法。采用资源块(RB)大小可变的分配方式满足 RB 的邻接约束, 引入混合自动轮询机制解决算法对低优先级业务调度不公平的问题。仿真结果表明, 相比 PF 算法、RR 算法, 该算法的系统吞吐量分别提升约 15%、41%, 且在业务时延、业务区分度方面具有较好性能。

**关键词:** 物联网; 业务动态分组; 延迟容忍; 优先级调度; 自动轮询机制

**中文引用格式:** 金叶奇, 徐佑宇, 郑敏, 等. 基于动态分组的 M2M 上行优先级调度算法[J]. 计算机工程, 2019, 45(8): 129-134.

**英文引用格式:** JIN Yeqi, XU Youyu, ZHENG Min, et al. M2M uplink priority scheduling algorithm based on dynamic grouping[J]. Computer Engineering, 2019, 45(8): 129-134.

## M2M Uplink Priority Scheduling Algorithm Based on Dynamic Grouping

JIN Yeqi<sup>1,2</sup>, XU Youyu<sup>1,2</sup>, ZHENG Min<sup>1</sup>, TAN Chong<sup>1</sup>, WANG Hong<sup>1,2</sup>

(1. Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China)

**[Abstract]** Based on the Delay Tolerance Remaining Time (DTRT), an uplink priority scheduling algorithm for dynamic grouping of traffic is proposed. A Resource Block (RB) allocation method with variable size is adopted to satisfy the adjacency constraints of RB, and a hybrid automatic polling mechanism is introduced to solve the unfair scheduling problem of low priority traffic. Simulation results show that the system throughput of this algorithm is increased by 15% and 41% compared with PF algorithm and RR algorithm respectively, and it has better performance in terms of traffic delay and traffic differentiation.

**[Key words]** Internet of Things (IoT); traffic dynamic grouping; delay tolerance; priority scheduling; automatic polling mechanism

**DOI:** 10.19678/j.issn.1000-3428.0051120

### 0 概述

随着低功率广域 (Low Power Wide Area, LPWA) 技术的快速发展, 预计至 2020 年, 物联网 (Internet of Things, IoT) 中物与物 (Machine-to-Machine, M2M) 的连接数量将达 32 亿<sup>[1]</sup>。其中, 窄带物联网 (Narrow Band IoT, NB-IoT) 以安全性高、覆盖能力强而得到研究者的广泛关注<sup>[2]</sup>。机器类型通信设备 (Machine Type Communication Devices, MTCD) 产生的海量 M2M 业务主要由上行链路方向小型突发载荷构成, 包括特定服务质量 (Quality of Service, QoS)、功耗以及传输期限等要求<sup>[3]</sup>, 其对有限信道资源下的调度提出了挑战, 而以 NB-IoT 为代表的基于 LTE 的 M2M 上行调

度作为缓解信道资源紧张的关键技术, 成为当前的研究热点之一。

M2M 上行调度的难点在于如何利用有限的无线资源来满足海量终端数据传输的要求, 并适应由其业务特征多样性带来的各类约束, 包括时延、功耗、吞吐量及系统容量等。文献[4-6]对经典的轮询调度算法 RR 和比例公平调度算法 PF 进行改进, 使其适合于 M2M 业务。但由于未改变调度原理, 导致算法不能充分利用 M2M 时延容忍特性来满足海量终端调度的要求。文献[7]同时考虑功率、延迟和网络容量, 提出一种基于负载的多时隙调度算法, 其能较好地满足业务的特定需求, 但优化过程中所作出的某些假设在实际应用中难以实现。文献[8]研究

**基金项目:** 国家自然科学基金“无线多媒体传感器网络最优化资源分配与传输技术研究”(61401445)。

**作者简介:** 金叶奇 (1994—), 男, 硕士研究生, 主研方向为宽带无线通信、物联网资源调度技术; 徐佑宇, 硕士研究生; 郑 敏, 研究员、博士、博士生导师; 谭 冲, 副研究员、博士; 王 虹, 硕士研究生。

**收稿日期:** 2018-04-08      **修回日期:** 2018-06-28      **E-mail:** yeqi.jin@mail.sim.ac.cn

M2M 系统在过载情况下的混合资源分配问题,提出一种最大化准入 MTCD 下解决非线性规划问题的方法。

在无线资源有限时,应尽可能权衡不同业务数据的紧急程度。相关研究表明,满足数据传输期限需求的调度用于 M2M 时较为有效<sup>[9]</sup>。文献[10]提出基于类的优先级(Class Based Priority, CBP)算法,其依据业务 QoS 参数确定数据优先级并进行分类,为解决由多业务需求引发的调度问题提供了思路。但是,在 M2M 流量突发时,该算法静态优先级分配方式中某些优先级较低但仍具备传输期限的业务极有可能因高优先级业务的突发而错过传输期限。针对该不足,文献[11]提出平衡交替技术,在信道和延迟之间实现动态自适应平衡,通过非直接方式(降低误码率)来保障延迟性能。文献[12]在 CBP 算法的基础上,提出一种基于类的总体优先级调度算法(Class Based Overall Priority, CBOP),但该算法仅能调节各类优先级的顺序,并未对业务所属的类别进行动态调整。

为克服静态优先级分配方式在上行业务高并发下的不足,以及 CBOP 算法业务所属分组不可调的问题,本文在 CBP 算法的基础上,以业务延迟容忍剩余时长(Delay Tolerance Remaining Time, DTRT)为度量依据对 M2M 业务进行动态分组,提出一种上行优先级调度算法(Priority Scheduling Algorithm, PSA),以满足业务的时延约束,在此基础上,设计一种混合机制来解决优先级算法的调度不公平问题。

## 1 系统模型

### 1.1 问题分析

本文以满足数据的时延约束为目标,赋予时延容忍程度低、紧急性强的业务以更高的调度优先级。延迟约束下的分组问题表述如下:当前时刻请求服务的业务集合定义为 $\{1, 2, \dots, m, \dots, M\}$ ,各业务所属分组集合定义为 $\{G_1, G_2, \dots, G_x, \dots, G_g\}$ ,以业务的延迟容忍时长(Delay Tolerance Time, DTT)为指标进行业务分组,如下:

$$DTT_{G_{x-\min}} < DTT_{m \in G_x} \leq DTT_{G_{x-\max}} \quad (1)$$

其中,DTT 值即业务传输时延期限值,其代表业务的时延敏感程度。在式(1)中,属于分组 $G_x$ 的业务 $m$ ,其 DTT 值处于分组 $G_x$ 的 DTT 范围之内。根据 DTT 值对业务进行分组,对应 $n$ 个优先级 $\{P_1, P_2, \dots, P_n\}$ 。由于 DTT 是一个静态参数,为实现动态分组,本文采用动态的业务指标参数 DTRT 代替 DTT。

本文上行采用峰均比更低的单载波频分多址(Single-Carrier Frequency-Division Multiple Access, SC-FDMA)技术,因此,动态分组后的用户资源块(Resource Block, RB)分配存在限制,不能将离散的 RB 分配给用户。SC-FDMA 下 RB 分配算法的性能

通常用有效容量进行衡量<sup>[13]</sup>。业务 $m$ 的有效容量指为保障由 $\theta_m$ 指定的业务 $m$ 的 QoS 要求,调度算法可支持的最大业务到达速率,其计算如下:

$$E_c^m(\theta_m) = -\frac{1}{\theta_m} \ln \Psi(e^{\theta_m R_m}) \quad (2)$$

其中, $E_c$ 为有效容量, $\Psi(\cdot)$ 表示期望操作, $\theta_m$ 是业务 $m$ 的 QoS 参数, $R_m$ 是业务 $m$ 的数据速率。考虑常数到达率 $\lambda$ ,业务 $m$ 超出延迟容忍的概率为:

$$\delta = \text{Prob}\{D_m > DTT\} \approx \varphi_m e^{-\theta_m DTT} \quad (3)$$

$$E_c^m(\theta_m) \geq \lambda_m \quad (4)$$

其中, $D_m$ 是业务 $m$ 数据包经历的延迟, $\varphi_m$ 是设备缓冲区非空的概率。为保证业务的 QoS 需求,需满足约束条件式(4)。为获得 $\theta_m$ ,本文采用香农容量公式,将业务 $m$ 的最大可传输速率表示为:

$$R_m = B \log \left( 1 + \frac{P_m |h_m|^2}{\sigma^2} \right) \quad (5)$$

其中, $B$ 是每个 RB 的带宽, $P_m$ 是业务 $m$ 所对应设备的发射功率, $h_m$ 是信道增益, $\sigma^2$ 代表加性高斯白噪声(Additive White Gaussian Noise, AWGN)的功率, $\frac{P_m |h_m|^2}{\sigma^2} = \text{SNR}$ 为业务 $m$ 对应 MTCD 的信噪比。在

RB 连续分配的约束条件下,最大化式(5)并不容易,其已被证明是一个 NP 难题<sup>[14]</sup>,为获得最优解,需详尽地搜索所有可行的 RB 分配方案,该过程具有指数复杂度 $O(M^K)$ ,在实际应用中,通常采用启发式算法提供接近最佳的解决方案<sup>[15]</sup>。

最低优先级业务的时延容忍占 M2M 通信总量的绝大部分,需要利用此特性简化调度以减小开销。但高优先级业务持续突发时低优先级业务长期得不到调度而“饿死”的情况时有发生,因此,还需解决优先级算法的调度不公平性问题。

### 1.2 业务分组模型

设所有 MTCD 均匀分布于 LTE 单蜂窝小区,对业务 $(1, 2, \dots, m, \dots, M)$ 进行分组。分组数 $g$ 过大会降低调度效率,过小则不能很好地区分业务时延敏感程度。参考文献[16],本文设置 $g=4$ ,将业务分为4组,如图1所示。其中, $G_1$ 为发送数据量少但 DTT 很低的时延不容忍型业务,如各类警报业务,其随机突发性强,需立即上报; $G_2$ 为人为干预下发生的 M2M 通信业务,如通过国网采集终端后台查询电表信息等,其数据量大小不等,由于涉及 H2M 通信,因此时延敏感程度较高; $G_3$ 为具有最低传输速率要求的业务,如视频监控等,其具有一定的时延容忍性,但必须在 DTT 内调度一定量的数据以满足其最低传输比特率; $G_4$ 为发送量小且时延容忍的业务,如环境监测,此类业务占比最大,对实时性要求较低,优先级低于上述3种类型,若无线资源紧张,可适当推迟该类流的调度。

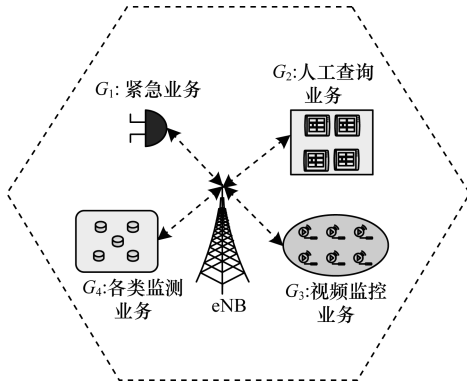


图 1 基于时延的业务分组模型

## 2 SC-FDMA 技术下的动态优先级调度算法

### 2.1 基于 SC-FDMA 的资源分配算法

由于 RB 的邻接约束,需将  $K$  个 RB 划分为  $Y$  个有序的集合,每个集合具有  $k_a$  个相邻 RB,使得  $K = k_1 + k_2 + \dots + k_Y$ 。将分配给业务的 RB 用 1 表示,未分配的 RB 用 0 表示,可得到业务  $m$  的 RB 分配矩阵  $R_{k,j}^m$ 。以  $K=4$  为例,对于业务  $m$ ,其所有的可行 RB 分配模式用分配矩阵表示如下:

$$R_{k,j}^m = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad (6)$$

其中,分配矩阵的列对应可行的 4 个 RB 分配模式,行对应 RB 的索引。例如,式(6)的第 2 列表示 RB1 分配给  $m$ ,而 RB2~RB4 则未分配给  $m$ ,第 11 列表示 RB1~RB4 全都分配给  $m$ 。邻接约束体现在多个 RB 被分配给某一 M2M 业务时,这些 RB 必须彼此相邻。RB 分配矩阵的阶数为  $K \times J$ ,  $J = 1/2 \times (K^2 + K) + 1$  代表所有可行的分配模式总数,在式(6)中,  $J=11$ 。基于 RB 邻接约束的 RB 大小可变分配算法考虑不同业务所需 RB 数不同,以  $N$  表示未被分配且连续的 RB 组,  $R$  表示二维分配矩阵,  $R_j^m$  表示业务  $m$  所需的 RB 数(即块大小),  $j$  是所有可用 RB 中分配给业务  $m$  的  $R_j^m$  个连续 RB 中的首个 RB 索引值,通过索引  $j$  和 RB 数量  $R_j^m$ ,即能确定业务  $m$  所使用的资源位置。在每个调度周期,即传输时间间隔(TTI)中,算法通过  $R_j^m$  的值来确定一个最小且满足要求的块,从而为业务  $m$  提供服务,然后将此块标记为已被分配,并按该步骤继续为业务  $m+1$  安排 RB 块。邻接约束启发式算法描述如下。

**算法 1** 基于邻接约束的块大小可变 RB 分配算法

输入 二维分配矩阵  $R$

输出 业务的 RB 分配方案

1.  $N \leftarrow$  空闲 RB 数  $U_{RB}$

2.  $R_j^m \leftarrow$  业务  $m$  从第  $j$  个 RB 开始所需的 RB 总数

3. while  $N$  未被分配完 do

4. 在  $R$  中寻找满足  $R_j^m$  要求的最小块

5. 分配  $N$  中的第  $j$  个~第  $(j+R_j^m-1)$  个 RB 给业务  $m$

6. 将二维分配矩阵  $R$  中的第  $j$  列~第  $(j+R_j^m-1)$  列的元素置  $\infty$  (标记为不可被选择)

7. 标记  $N$  中的第  $j$  个~第  $(j+R_j^m-1)$  个 RB 已被分配

8. 为业务  $m+1$  分配 RB

9. end while

### 2.2 业务动态分组算法

在式(1)中,以业务的动态  $DTRT$  值代替静态  $DTT$  值作为指标对所有流进行分组,每个 TTI 通过式(7)动态更新不同业务的  $DTRT$  值。

$$DTRT_m = DTT_m - n_m \times t_{TTI} \quad (7)$$

其中,  $n_m$  是业务  $m$  到达后经过的 TTI 数,  $t_{TTI}$  是调度周期 TTI 的时长,本文设为 1 ms。以业务  $DTRT$  值代入 1.2 节提出的业务分组模型中,得到表 1 所示的分组。各业务在其  $DTRT$  值减小到 0 之前必须被调度,否则将超出其时延容忍范围。业务刚到达时,其  $DTRT$  值等于  $DTT$  值,根据该  $DTRT$  值所属范围确定一个初始组以及该组对应的优先级。

表 1 M2M 业务所属分组及其优先级

组别	$DTRT$ 值/ms	典型应用	优先级	流量占比/%
$G_1$	0~20	告警信息	1	10
$G_2$	20~150	人工查询	2	15
$G_3$	150~500	视频监控	3	15
$G_4$	>500	环境监测	4	60

● 优先级 1 为最高优先级,所有延迟不容忍的业务都保存在此组中,在每个 TTI 内将被优先调度。对于初始所属分组为  $G_2$  和  $G_3$  的业务,若在一个调度周期内未被调度,则每经过一个 TTI,  $DTRT$  值就会减少 1 ms,如果属于低优先级的业务未在该分组的  $DTRT$  值范围内被服务,将自动转移到上一个高优先级组中,从而拥有更高的调度优先级。假设某业务的  $DTT$  值为 180 ms,其最初  $DTRT$  值为 180 ms,分组  $G_3$  优先级为 3,若经过 30 ms 后仍未得到调度,其  $DTRT$  值减小为 150 ms,自动转到优先级为 2 的分组  $G_2$  中,完成动态分组。 $G_4$  业务由于占比大且延迟容忍程度高,为提高调度效率,算法不考虑提升其优先级,即  $G_4$  的流始终处于最低优先级。

### 2.3 混合自动轮询机制

文献[13]指出,在低优先级的业务量明显大于高优先级的业务量时,应尽量确保高优先级业务得到服务;反之,则应确保低优先级数据包不会“饿死”。优先级算法保障了业务时延性能,提高了传输效率,但和其他分组相比,分组 4 的业务获得的调度机会较低。此外,MTCD 呈现出集群特性,即在某些 eNB 覆盖范围内,产生高优先级业务的 MTCD 数可能很高。当网络容量较小或大量设备连接到网络时,为避免  $G_4$  业务长时间得不到调度而“饿死”,本

文对上述动态调度进行改进,提出一种混合自动轮询(Hybrid Automatic Round Robin, HARR)机制。将动态调度与轮询调度 2 种策略相结合,每隔一段动态调度周期就进入一次轮询周期,轮询调度周期可避免资源紧张时最低优先级业务始终得不到调度的现象。根据非时延容忍( $G_1$ 、 $G_2$  和  $G_3$ )与时延容忍( $G_4$ )的业务量比例,确定动态调度周期数与轮询调度周期数(固定为 1)的配比  $\alpha$  为:

$$\alpha = \mu \Theta \left( 1 - \frac{N_4}{N_{1,2,3,4}} \right) \quad (8)$$

其中,  $N_4$  为已到达的  $G_4$  业务量,  $N_{1,2,3,4}$  为当前到达的总业务量,  $\Theta(\cdot)$  是  $\alpha$  更新公式, 可为分段或阶梯函数。 $\alpha$  用于控制调度过程中 DTRT 动态调度周期数与轮询调度周期数的比例, 即每 2 个轮询周期的间隔。 $N_4$  越大, 时延容忍业务所占比重越高,  $\alpha$  越大, 2 个轮询调度周期之间的间隔越大。 $\alpha$  值不宜太小, 其仅为最低优先级业务提供传输机会, 不能对系统吞吐量造成太大影响。本文基于 HARR 机制的动态优先级调度算法描述如下:

#### 算法 2 HARR 机制下的动态优先级调度算法

输入 上一 TTI 剩余业务集合  $S_1$ , 当前 TTI 新到达业务集合  $S_2$ ,  $S_1$ 、 $S_2$  内的流 DTRT 值,  $\alpha$

输出 业务的 RB 分配方案, 剩余未被调度的业务集合

1.  $U_{RB} \leftarrow$  空闲 RB 数

2. if  $\beta < \alpha$  then

3. 业务所属分组  $G_1/G_2/G_3/G_4 \leftarrow$  业务 DTRT 值

4. while  $U_{RB} > 0$  do

5. 根据 DTRT 值将 RB 分配给  $G_1$  的所有 M2M 流

6. 根据 DTRT 值将 RB 分配给  $G_2$  的所有 M2M 流

7. 根据 DTRT 值将 RB 分配给  $G_3$  的所有 M2M 流

8. 将 RB 分配给  $G_4$  的所有 M2M 流

9. end while

10.  $\beta \leftarrow \beta + 1$

11. else

12. while  $U_{RB} > 0$  do

13. 对所有 M2M 流执行 RR 算法

14. end while

15.  $\beta \leftarrow 0$

16. 更新  $\alpha$  值

17. end if

18. 更新未被调度业务的 DTRT 值( $G_4$  除外)

在算法 2 中, 步骤 5~步骤 8、步骤 13 中的 RB 分配均采用算法 1。在优先级调度阶段, 每经过一个 TTI, 距离上一次轮询阶段的周期数( $\beta$ )加 1。此阶段调度器首先服务于  $G_1$  的所有流, 无线资源根据 DTRT 值升序的方向进行分配, 以同样的方式依次将 RB 分配给  $G_2$ 、 $G_3$  和  $G_4$  的 M2M 流,  $G_4$  在其他优先级业务都已满足时才被调度。经过  $\alpha$  个优先级调度周期后,  $\beta = \alpha$ , 进入一次轮询调度周期, 此时根据当前业务情况更新  $\alpha$  值。

在每个 TTI 结束时, 对  $G_1$ 、 $G_2$ 、 $G_3$  中未被调度的流 DTRT 值进行更新, 其中, DTRT 值减小至 0 的

数据包意味着超时, 其将视为传输失败而丢弃,  $G_4$  中的业务无需更新。一个 TTI 中要分配给所有 M2M 请求的 RB 数量满足如下约束:

$$\sum_{g=1}^G RB_{mg} < L_m \quad (9)$$

其中,  $RB_{mg}$  是 TTI 中分配给  $g$  组 M2M 业务的 RB 总数,  $L_m$  是可分配的 RB 最大数量。更新 DTRT 时的执行次数随  $G_1$ 、 $G_2$  和  $G_3$  的业务量增长而线性增长, 算法 2 的时间复杂度为  $O(n)$ 。

### 3 仿真结果与分析

本文采用 Matlab 作为平台, 对算法进行系统仿真与评估。将表 1 中的数据作为 M2M 业务源的模拟参数, 主要包括业务 DTRT 值、优先级及流量占比情况。系统仿真参数设置如表 2 所示。

表 2 仿真参数设置

参数名	参数值
小区半径/km	1
eNB 数目	1
MTCD 数目	100 ~ 1 000
调度周期 TTI 时长/ms	1
每个 TTI 中的 RB 数目	15
上行链路带宽/MHz	3
仿真时长/TTI	1 000
基站的发射功率/dBm	43
天线增益/dBi	15
传输功率/dBm	30
MTCD 分布情况	均匀分布

本文算法的目的在于满足不同类型 M2M 流量的 QoS 需求, 仿真主要侧重算法时延和吞吐量的性能评估。首先对无 HARR 机制下的动态算法与 PF 算法进行分析, 其中, 发送消息的平均时延代表调度程序提供的服务质量, 图 2 所示为 MTCD 数目分别为 100 和 400 时 2 种算法下数据发送的平均时延情况。

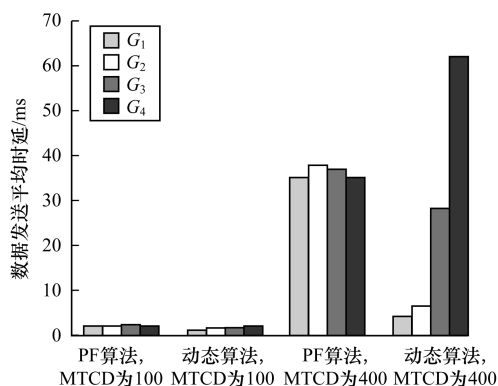


图 2 2 种算法数据发送平均时延对比

从图 2 可以看出, 当 MTCD 数较小时, 2 种调度算法时延性能相差不大, 当 MTCD 数较大时, PF 算法下各优先级数据的发送时延分布相对均匀, 而本

文动态算法的平均时延则表现出很好的区分度。对于  $G_1$ 、 $G_2$  和  $G_3$  的流,动态算法的平均延迟明显优于 PF 算法,对于  $G_4$  的流,动态算法的平均时延则较高,原因是 PF 算法未考虑不同 M2M 优先级,RB 资源在各组业务之间被共享,而动态算法依据优先级确定传输顺序,降低了高优先级业务的时延,但同时会增加低优先级业务的时延。

本文算法在 MTCD 数为 200 ~ 1 000 时针对不同分组业务的吞吐量情况如图 3 所示。

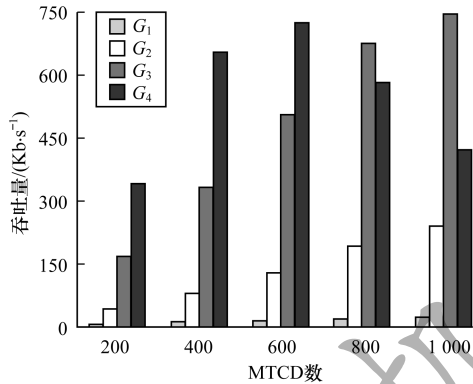


图3 不同 MTCD 数目下各组业务的吞吐量对比

从图 3 可以看出,  $G_1$ 、 $G_2$  和  $G_3$  的吞吐量随着 MTCD 数量的增加而增大。 $G_1$  分组业务由于占比较低,且每次发送的数据量都很小,因此其吞吐量的变化不明显,但该组业务最早被服务。 $G_4$  业务的吞吐量在 MTCD 数较少时逐渐增加,在 MTCD 数增加到 600 之后开始下降,原因是 MTCD 数较多时前 3 个高优先级的业务量变大,算法将 RB 资源优先分配给更高优先级的业务,对于  $G_4$  则采取“尽力而为”的方式提供服务。

在无线资源无法满足大规模业务需求的情况下,到达业务中高优先级业务所占比例越大,低优先级业务“饿死”的情况越普遍。在 MTCD = 1 000、不改变各组业务量配比时对动态算法引入 HARR 机制,将  $\alpha = 5$ 、 $\alpha = 20$  时各组业务的吞吐量与不添加 HARR 机制的算法业务吞吐量进行对比,结果如图 4 所示。

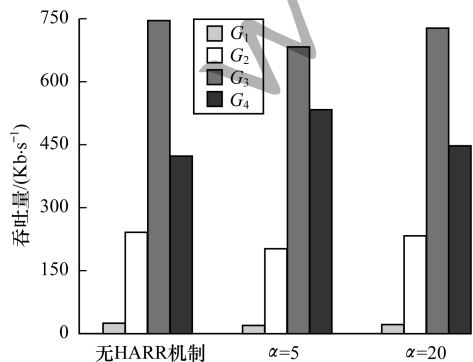


图4  $\alpha$  值对系统吞吐量的影响

从图 4 可以看出,以不添加 HARR 机制时的吞吐量为参照,当  $\alpha = 5$  时,  $G_2$  和  $G_3$  分组的吞吐量均有一定下降,  $G_4$  吞吐量上升,原因是每隔 5 个动态周期就执行一次轮询算法周期,轮询时  $G_4$  业务占用了一部分原来分配给  $G_1$ 、 $G_2$  和  $G_3$  的 RB 资源,提升了算法对  $G_4$  的公平性。 $\alpha = 20$  时,  $G_2$  和  $G_3$  分组的吞吐量下降量、 $G_4$  吞吐量的上升量以及总吞吐量的下降量均比  $\alpha = 5$  时要小。鉴于设置  $\alpha$  的目的是根据高低优先级业务实时比例动态地为最低优先级业务提供一定数据传输机会,而非提高低优先级业务的吞吐量,且不能对动态算法下的系统吞吐量造成太大影响,因此,  $\alpha$  的值可设置为 20 以上。

图 5 所示为本文改进动态算法、CBP 算法、PF 算法、RR 算法下的系统总吞吐量对比结果。

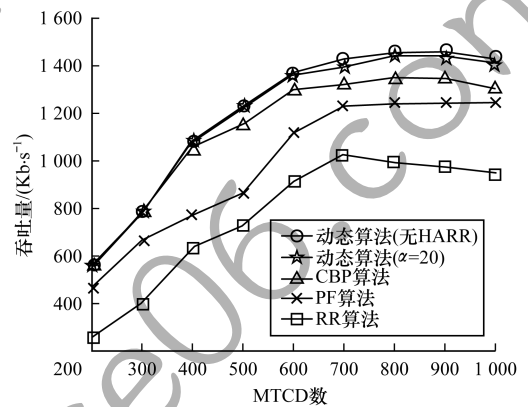


图5 不同算法的系统吞吐量对比

从图 5 可以看出,在 PF 算法和 RR 算法下,为确保 RB 的连续性,业务分配到的资源块大小不变,某些数据量小的业务分配到多余的 RB,造成了资源浪费。本文动态算法分配可变大小的块以确保 RB 的连续性,发送相同量的数据时所需 RB 更少,且与静态 CBP 算法、PF 算法和 RR 算法相比,本文算法中业务因错过最终期限而传输失败的可能性大幅降低,最终算法在提高系统吞吐量的同时也保证了紧急信息更快更有效地被送达。在当前仿真条件下,本文算法的系统吞吐量比 PF 算法提高约 15%,比 RR 算法提高约 41%。添加 HARR 机制的动态算法在  $\alpha$  较大时对系统吞吐量影响很小,但此机制可有效避免网络拥挤时低优先级业务长期得不到服务的现象。

#### 4 结束语

本文提出一种基于延迟容忍剩余时长的 M2M 业务优先级调度算法。每个 TTI 结束时动态更新业务所属分组,引入混合自动轮询机制解决算法对低优先级业务的调度不公平问题。仿真结果表明,与传统静态算法相比,该算法能充分利用 M2M 业务多

QoS 等级的特性,且在网络容量较小时可有效避免低优先级业务长期得不到调度的现象。下一步将结合半静态调度思想对动态调度方案中较高的信令开销进行优化,使其更适用于 M2M 的海量业务调度。

### 参考文献

- [1] Cisco visual networking index: global mobile data traffic forecast update 2015-2020 [R]. Cisco Public Information, 2016: 1-5.
- [2] GUDKOVA I A, SAMOUYLOV K E, BUTURLIN I A, et al. Analyzing impacts of coexistence between M2M and H2H communication on 3GPP LTE system [C]//Proceedings of International Conference on Wired/Wireless Internet Communications. Washington D. C., USA: IEEE Press, 2014: 162-174.
- [3] 赵继波, 谭献海. 基于 M2M 业务的网络流量特性分析研究 [J]. 物联网技术, 2014, 4(8): 36-38.
- [4] ALQAHTANI S A, ALHASSANY M. Comparing different LTE scheduling schemes [C]//Proceedings of Wireless Communications and Mobile Computing Conference. Washington D. C., USA: IEEE Press, 2013: 264-269.
- [5] SUN Zheqi, YU Haifeng, CHI Xuefen, et al. Research on uplink scheduling algorithm of massive M2M and H2H services in LTE [C]//Proceedings of IET International Conference on Information and Communications Technologies. Washington D. C., USA: IEEE Press, 2013: 365-369.
- [6] KUMAR A, ABDELHADI A, CLANCY C. A delay-optimal packet scheduler for M2M uplink [C]//Proceedings of Military Communications Conference. Washington D. C., USA: IEEE Press, 2016: 295-300.
- [7] 闵明慧, 杨志家, 李中胜, 等. 工业物联网应用中多时隙帧调度算法研究 [J]. 计算机工程, 2016, 42(11): 15-21, 26.
- [8] 王鑫, 邱玲. H2H 与 M2M 共存场景的准入控制及资源分配 [J]. 中国科学院大学学报, 2016, 33(3): 427-432.
- [9] MOSTAFA A, GADALLAH Y. A statistical priority-based scheduling metric for M2M communications in LTE networks [J]. IEEE Access, 2017, 5: 8106-8117.
- [10] GILUKA M K, KUMAR N S, RAJORIA N, et al. Class based priority scheduling to support machine to machine communications in LTE systems [M]. Washington D. C., USA: IEEE Press, 2014: 1-6.
- [11] ELHAMY A, GADALLAH Y. BAT: a balanced alternating technique for M2M uplink scheduling over LTE [C]//Proceedings of Vehicular Technology Conference. Washington D. C., USA: IEEE Press, 2015: 1-6.
- [12] CHEN Beichen, FAN Zhong, CAO Fengming, et al. Class based overall priority scheduling for M2M communications over LTE networks [C]//Proceedings of Vehicular Technology Conference. Washington D. C., USA: IEEE Press, 2015: 20-25.
- [13] WU Dapeng, NEGI R. Effective capacity: a wireless link model for support of quality of service [J]. IEEE Transactions on Wireless Communications, 2003, 2(4): 630-643.
- [14] LEE S B, PEFKIANAKIS I, MEYERSON A, et al. Proportional fair frequency-domain packet scheduling for 3GPP LTE uplink [EB/OL]. [2018-03-20]. <http://ants.iis.sinica.edu.tw/3bkj9ltewxtsrrvnoknfdxrm3zfwrr/63/Proportional%20Fair%20Frequency-Domain%20Packet%20Scheduling%20for%203GPP%20LTE%20Uplink.pdf>.
- [15] YANG Hongkun, REN Fengyuan, LIN Chuang, et al. Frequency-domain packet scheduling for 3GPP LTE uplink [EB/OL]. [2018-03-20]. [http://nns.cs.tsinghua.edu.cn/paper/infocom10\\_hky.pdf](http://nns.cs.tsinghua.edu.cn/paper/infocom10_hky.pdf).
- [16] 王东. 面向 5G 的 M2M 通信低功耗覆盖增强及资源调度的研究 [D]. 北京: 北京交通大学, 2017.

编辑 吴云芳