

应用预留资源进行补偿的无线网络公平调度算法的探讨

尹建璋

(长征学院, 浙江 杭州 310023)

摘要:提出了一种结合预留资源进行补偿的无线公平调度算法。当系统呼叫切换频率很低时, 将预留资源的空闲部分中的一部分用于补偿, 以提高系统的资源利用率; 当系统呼叫切换频率很高时, 进行补偿的业务流均为那些获得额外服务的业务流, 本算法应用在系统呼叫的切换频率很低的情况下, 可以充分利用频带资源的优势。

关键词: 预留资源; 无线网络; 算法

中图分类号: TN929.5

文献标识码: A

Study of wireless fair scheduling algorithm with the resources set aside applied to compensation

YIN Jian Zhang

(Changzheng Institute, Hangzhou 310023, China)

Abstract: In order to improve the utilization of system resources, this article presents a combination of earmarked resources to compensate for wireless fair scheduling algorithm. When the system calls in very low switching frequency, it will set aside some free resources to compensate for the part in order to improve system resource utilization; When the system calls in a very high switching frequency, the flow of business to compensate those who were given extra services business stream, the algorithm used with system calls with very low switching frequency, the band can take full advantages of the resources.

Key words: reserve resources; wireless network; algorithm

随着无线网络的发展, 移动通信用户数和 Internet 用户数急剧增加, 人们期望新一代移动通信系统不仅具有更大的容量, 还要支持移动多媒体业务, 除了提供话音业务外, 还支持低/高速数据、图像等非话音业务的传输。不同业务有不同的服务质量 (QoS) 要求, 如对时延、误比特率、数据速率的要求不同。无线网络设计有两大目标: 一是保证各类业务的 QoS 要求, 二是使网络的资源利用率达到最大, 这就需要借助于无线资源管理。而目前的无线分组调度算法主要集中在保障各连接的 QoS 前提下, 追求系统吞吐量极大化, 也就是系统利用率的提高。但是系统吞吐量极大化与公平服务是一对矛盾。大多数算法的公平性主要体现在长期上, 而未考虑短期公平性的问题, 也就是未考虑无线信道特殊性引入的补偿问题: 当一个连接的链路从故障恢复后, 如何对这个连接进行有效的补偿是值得研究的。这些补偿方式从某种程度上说也是不公平的, 因为从完全公平的角度出发, 应该是那些得到额外服务的连接对滞后流做出补偿。因此, 在无线网络中对预留资源进

行补偿的无线公平调度算法进行研究具有重要意义。

1 应用预留资源进行补偿的无线网络公平调度算法的描述

由于采用了加权的调度算法, 额外带宽的分配直接隐含在连接获得的总带宽分配中, 这是由于加权调度算法中每一个连接获得的带宽满足公式 (1)。

$$b_i = \frac{w_i}{\sum_j w_j} \times B_s(t) \quad (1)$$

其中 w_i 为连接 i 的权重, $\sum_j w_j$ 是系统中所有无差错的连接的权重和, $B_s(t)$ 是当前时刻的系统带宽。

算法描述如下:

- (1) 当进入系统的切换呼叫频率不高时, 此时预留资源有部分资源处于空闲状态, 将这部分资源的一部分用于补偿;
- (2) 当进入系统的切换呼叫频率很高时, 预留资源的使用率很高, 采用领先流释放一部分带宽用于补偿, 而不是

《微型机与应用》2009 年第 20 期

网络与通信 Network and Communication

将其所有带宽都用于补偿,直至这个领先流变成同步流。这样做的好处是补偿时并不中断对领先流的服务。补偿方式采用针对滞后流固定比例获得补偿的方式^[1]。

为了简化计算,算法定义一个预留资源空闲率参数 $S = \text{空闲预留资源} / \text{总预留资源}$,并且设定一个预留资源空闲率门限 S_t ,当 t 时刻 $S(t) \geq S_t$ 时,采用方式 1 进行补偿;当 t 时刻 $S(t) < S_t$ 时,采用方式 2 进行补偿^[2]。

方式 1 的计算较为简单,滞后流直接获得一定比例 (λ) 的空闲预留资源 (BFR) 用于补偿,即获得 $\lambda \times \text{BFR}$ 的额外带宽。不将全部预留资源用于补偿的目的是为了避免在补偿过程中系统拒绝新的切换呼叫。

可以推得理想模式下 (即所有领先流都完全进行了补偿) 补偿所需要的时间为:

假设 f_i 连接从 t_i 时刻开始故障,故障时间为 T_i ,补偿时间为 X_i ,那么 f_i 在 T_i 时刻以及补偿结束时刻均为同步流状态。未发生故障时间在此期间 $(T_i + X_i)$ 得到的服务 S_i 应该等于发生故障后 f_i 在补偿过程中 (X_i) 总共得到的服务 S_i^* 。

即 $S_i = b_i \times (T_i + X_i) = S_i^* = b_i^* \times X_i$, 此时 $S_i^* = b_i + \lambda \times \text{BFR}$, 可以解得:

$$X_i = \frac{b_i}{\lambda \times \text{BFR}} \times T_i \quad (2)$$

方式 2 中,算法采用了滞后流固定比例获得补偿的无线公平调度算法。针对一个滞后流 1,当它的链路恢复正常时,算法直接分配其预约带宽的固定比例 Δ_i ($0 < \Delta_i < 1$) 用于补偿。这部分补偿带宽由领先流按照各自权重分配。其方法如下列关系式所示:

$$w_{la}^*(i) = w_{la}(i) \times (1 + \Delta_i) \quad (3)$$

$$w_{le}^*(j) = w_{le}(j) - \frac{w_{le}(j)}{\sum_j w_{le}(j)} \times \sum_j (\Delta_j \times w_{la}(i)) \quad (4)$$

式中 w_{la}^* 和 w_{le}^* 代表更新后的滞后流和领先流的权重, w_{la} 和 w_{le} 代表更新前的滞后流和领先流的权重。每当有连接状态发生变化时 (包括监测到故障恢复、滞后流或者领先流恢复成同步流),各个相应流的权重都根据公式进行调整。

由式 (3) 和式 (4) 可知,在补偿时滞后流的权重变大,领先流的权重变小,并且可以计算出此时所有连接 (包括领先流、滞后流和同步流) 的权重和 $\sum_k w_k$ 仍保持不变

$$\left(\sum_i w_{la}^* + \sum_j w_{le}^* = \sum_i w_{la} + \sum_j w_{le} \right)$$

由公式 (1) 可知补偿时滞后流实际得到的带宽变大,而领先流得到的带宽变小^[3]。

2 应用预留资源进行补偿的无线网络公平调度算法的仿真结果及分析

传统调度算法采用了 WF2Q+, 并且各个连接的权重直接等于其预约速率的大小,未做归一化处理。

2.1 所有连接均无差错

根据其分配的权重,根据式(1),可知,其和有线网络状

《微型机与应用》2009 年第 20 期

况基本一致,这证明了系统在无差错状态下工作是正常的。

2.2 有连接出现链路故障

选择分组数据流 3 来代表出现信道故障的连接,链路在时间 12 s 时出现故障,14 s 时恢复正常。

(1) 考虑特殊情况,此时预留带宽未被使用,即系统中无预约切换呼叫,采用方式 1 进行补偿, $\lambda = 3/4$, $\text{BFR} = 200 \text{ kb/s}$, Flow3 获得的补偿带宽为 $3/4 \times 200 = 150 \text{ kb/s}$ 。图 1 为采用本算法时的仿真结果。通过图 1 可以看到,当 Flow3 的连接发生故障时 (12 s~14 s),这部分额外带宽将被分配给无故障的连接,Flow1、Flow2、Flow4 获得的发送速率均得到了提高 (从图线的斜率变化可以看出,斜率变大),这种变化一直持续到 Flow3 的连接恢复 ($t = 14 \text{ s}$)。此时 Flow3 开始获得补偿 (发送速率提高),直至补偿结束 ($t = 22 \text{ s}$)。整个补偿过程中所有领先流均未对 Flow3 进行补偿 (发送速率维持在初始状态)^[4]。

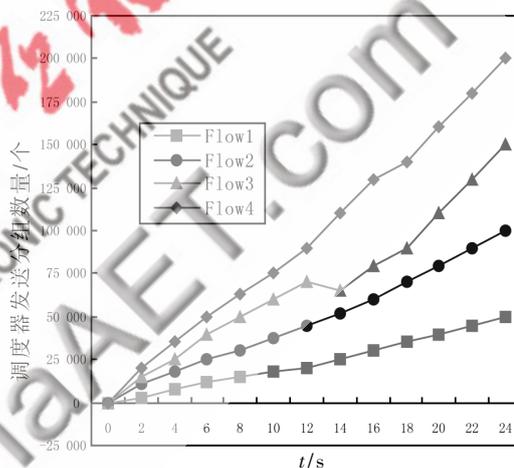


图 1 采用本算法方式 1 补偿的仿真结果

Flow 2、3 的传输速率曲线如图 2 所示。Flow 1、Flow4 与 Flow 3 类似。

根据式 (2) 可以计算补偿时间为 8 s,即补偿在 $14 + 8 = 22 \text{ s}$ 时结束,图 1 和图 2 说明了这一点。从图 2 中可以看到,Flow 3 在链路故障时传输速率降为 0,其额外带宽被分配给了其他链路状态良好的连接,Flow 2 的传输速率因此得到了提升,根据式 (1) 可以计算出此时 Flow 2 的传输速率为 570 kb/s ,这也在图 2 中得到印证。当 Flow 3 的链路恢复时,补偿开始,Flow 3 得到了 150 kb/s 的补偿带宽,因此速率上升到 750 kb/s ,而此时 Flow 2 并未对 Flow 3 进行补偿,故它的速率仍然保持为初始传输速率 400 kb/s 。在补偿结束后 (22 s),Flow 3 的速率恢复到初始传输速率 600 kb/s 。Flow 1、Flow4 的传输速率曲线与 Flow 2 的类似,这里略过^[5]。

(2) 系统中无空闲预留资源,采用方式 2 进行补偿 (针对滞后流固定比例获得补偿的方式), $\Delta_3 = 1/3$ 。图 3 为仿真结果。通过图 3 可以看到,当 Flow3 的连接恢复时,获得额外服务的连接将自己的部分带宽用于补偿 (斜率变小),

欢迎网上投稿 www.pcachina.com 37

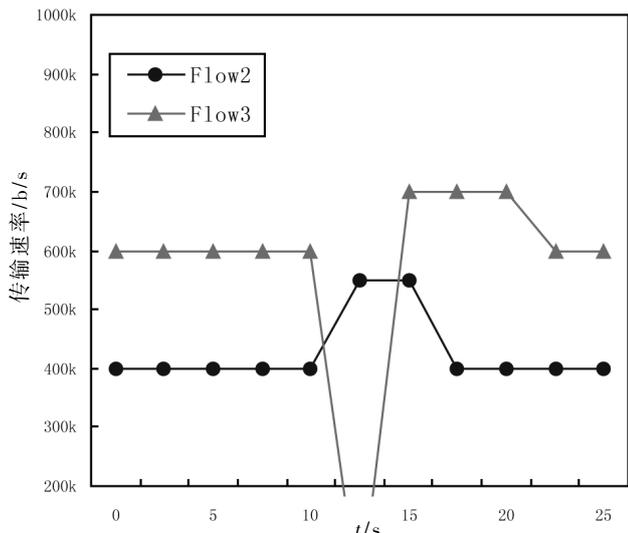


图2 Flow 2、Flow3的传输速率曲线

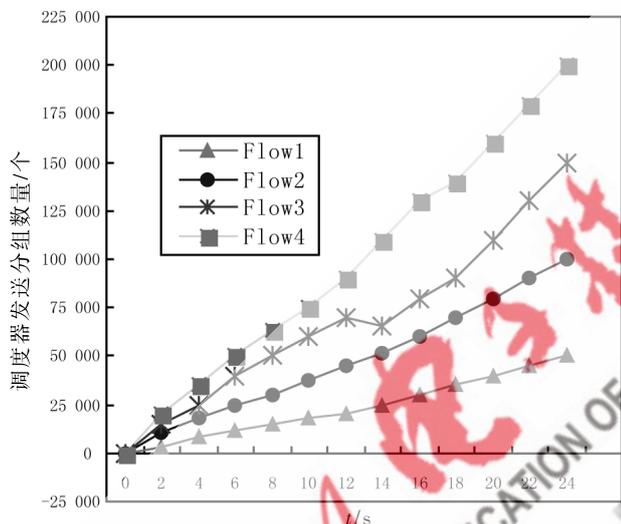


图3 采用本算法方式2补偿的仿真结果

直到补偿结束，各个连接的发送速率均恢复到初始状态(斜率与原来一致)。由于采用补偿策略为针对滞后流固定比例获得补偿，补偿耗费时间为 $1/\Delta_s$ 倍故障时间，即 6 s。Flow 3 在 $14+6=20$ s 时恢复同步流状态。

由于这里采用的是滞后流固定比例获得补偿的无线公平调度算法，因此其结果与图 1 的仿真结果一致。

图 4 比较了无差错状态与有差错状态(采用方式 1 和方式 2 进行补偿)时 Flow 4 的仿真结果。

由图 4 可见，系统一直无差错时，仿真结果为一条直线；当系统有差错时，Flow 4 将在差错状态时获得额外服务(12 s~14 s)。当采用方式 1 进行补偿时，Flow 4 并未受影响，其获得的额外服务未用于补偿，这意味着整个系统的资源利用率得到了提高。当采用方式 2 进行补偿后，补偿结束时 Flow 4 状态和无差错状态的仿真结果一致，也就是说采用方式 2 补偿后连接所得到的实际服务在补偿结束后与其预约的服务是一致的。这对所有的连接都是公平的。Flow1、Flow2 的结果与 Flow4 类似。

这里没有比较 Flow 3 在无差错状态和有差错状态的结

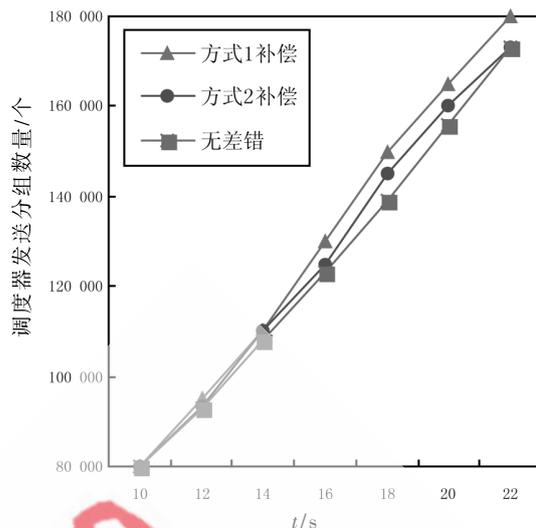


图4 无差错状态与有差错状态时 Flow4 的仿真结果比较

果，需要指出的是，无论采用方式 1 还是方式 2，在补偿结束后，Flow 3 得到的实际服务与其预约的服务是一致的。

3 应用预留资源进行补偿的无线网络公平调度算法复杂性分析

从空间复杂度上看，算法有 $3N$ 个固定存储空间。当采用方式 1 进行补偿时，时间复杂度上仅需计算一次乘法和一次加法。当采用方式 2 进行补偿时，时间复杂度的计算与采用的算法有关，由于本算法采用了无线公平调度算法，因此调整权重时最差情况下的时间复杂度为 $O(N^2)$ 。

综上所述，该文提出了一种无线公平调度算法——应用预留资源进行补偿的无线公平调度算法。当系统呼叫切换频率很低时，将预留资源的空闲部分中的一部分用于补偿；当系统呼叫切换频率很高时，进行补偿的业务流均为那些获得额外服务的业务流。

当采用方式 1 进行补偿时，直接利用预留资源中的部分空闲带宽进行补偿，这样可以提高系统的资源利用率。

当采用方式 2 进行补偿时，算法实际是通过补偿算法和再分配算法对各个连接的权重进行调整，从而实现补偿。该算法对于系统呼叫的切换频率很低的情况下，可以充分利用频带资源的优势。

参考文献

- [1] 纪阳,李迎阳,邓钢,等.一种适用于宽带无线 IP 网络的分组调度算法[J].电子学报,2003,31(05): 103-107.
- [2] 宣孝英,石冰心,邹玲.无线网络包调度算法综述[J].计算机工程与应用,2003,39(17):23-24+58.
- [3] 任颖颖,张文军,王彬.无线调度算法[J].计算机工程,2004,30(15):102-103+126.
- [4] 王燕,伍博,杨豪强,等.一种支持多业务的调度算法的研究与仿真[J].河南师范大学学报(自然科学版),2006,34(04): 195-197.
- [5] 宋舰,李乐民.一种支持服务类别的无线公平调度算法[J].电子学报,2004,32(01):60-64.

(收稿日期:2009-07-24)

《微型机与应用》2009 年第 20 期