

# 保证无线上行链路 TCP 公平的调度算法

李学勇<sup>1,2</sup>, 黄佳玮<sup>2,3</sup>, 王建新<sup>2</sup>

(1. 长沙大学信息与计算科学系, 长沙 410003; 2. 中南大学信息科学与工程学院, 长沙 410083;

3. 湖南广播电视大学计算机与工程学院, 长沙 410007)

**摘要:** 针对无线局域网中上行链路 TCP 流之间存在严重的无线信道资源分配不公平问题, 提出一种上行链路 TCP 流的调度算法 UFWFQ。该算法在接入节点上对上行 TCP 流采用与 WFQ 算法相同的轮询机制, 根据流的传送速率动态计算其服务概率, 分配给高速发送流较大的权重, 使各上行 TCP 流占用的信道时间相等, 以保证各流的加权公平性, 提高网络的总吞吐率。仿真实验结果验证了该算法的有效性。

**关键词:** 无线局域网; 上行链路 TCP 流; 调度算法; 公平

## Schedule Algorithm Guaranteeing Fairness of Wireless Uplink TCP

LI Xue-yong<sup>1,2</sup>, HUANG Jia-wei<sup>2,3</sup>, WANG Jian-xin<sup>2</sup>

(1. Department of Information and Computing Science, Changsha University, Changsha 410003;

2. School of Information Science and Engineering, Central South University, Changsha 410083;

3. College of Computer and Engineering, Hunan Radio TV University, Changsha 410007)

**【Abstract】** Aiming at the problem that the wireless channel resources are significantly unfairly allocated among the uplink TCP flows in WLAN, this paper proposes an uplink TCP flow schedule algorithm named UFWFQ(Up Fair WFQ). It uses the same polling as Weighted Fair Queuing (WFQ) at Access Point(AP). The service weights are dynamically set according to the transmission rate of flows. Flows with high transmission rate get more opportunities, and each flow occupies the same channel time. As a result, UFWFQ achieves the weighted fairness between single flow and total network efficiency. Simulation result proves that UFWFQ is effective.

**【Key words】** WLAN; uplink TCP flow; schedule algorithm; fairness

### 1 概述

随着无线局域网(WLAN)、WMN 等无线网络技术的发展, 各种无线网络使用户能更加方便地接入 Internet, 并提供便捷的无线网络服务。目前的无线网络中普遍采用 IEEE 802.11 标准的分布式协调功能(DCF)机制作为介质访问控制(MAC)层无线信道竞争规则。DCF 机制采用 CSMA/CA 技术, 保证各无线节点具有相同的信道接入概率, 公平地共享无线信道。然而, 无线接入网络 MAC 层的公平接入并不能保证传输层资源分配的公平性。作为当前主流的传输层控制协议, TCP 协议流占据了 Internet 的绝大部分网络流量。在无线接入网络中, 当多条上行(从无线节点发送到有线网络的节点)TCP 流共存时, 上行 TCP 流间出现了严重的吞吐率不公平问题<sup>[1]</sup>; 而当各 TCP 流的无线信道速率不同时, 慢速 TCP 流会占据大量的无线信道时间, 造成网络总体效率的低下<sup>[2]</sup>。因此, 在无线网络接入有线网络时如何保证 TCP 流公平和高效的信道资源分配成为一个重要的研究课题。

针对 TCP 流在无线接入网络中出现的公平和效率问题, 本文基于 WFQ(Weighted Fair Queuing)调度算法, 提出一种实现上行 TCP 公平的调度算法 UFWFQ(Up Fair WFQ)。该算法结合多速率无线网络的信道速率, 动态调节上行 TCP 流的 ACK 分组的服务概率, 将发送分组形成一个发送时间相等的轮询服务系统, 保证各上行 TCP 流占用相等长度的信道时间, 从而提高网络总吞吐率。

### 2 相关工作

在无线接入网络中, 接入节点(Access Point, AP)既与无线节点建立无线连接, 又连入有线 Internet, 成为无线网络和有线网络的结合点。当多条上行 TCP 流同时发送时, 各上行 TCP 流的 ACK 在 AP 的下行缓存排队等待发送到无线节点。如果下行缓存出现拥塞而丢弃 ACK 分组, 在上行流之间可能产生不公平问题。由于 ACK 分组的累计确认效应, TCP 发端在超时之前收到 ACK 分组就不会引发超时重传。当 ACK 分组拥塞丢弃时, 由于大拥塞窗口的 TCP 流中正在排队的 ACK 分组较多, 因此出现大量连续的 ACK 丢弃而引发超时重传的概率较小, 表现出较强的健壮性, 而小拥塞窗口的 TCP 流由于 ACK 分组丢弃而超时的可能性较高, 从而造成拥塞窗口大小不同的上行流间吞吐率的不公平。

针对无线网络接入有线网络时上行 TCP 流出现的不公平问题, 文献[1]提出了在 AP 上的 ACK 过滤算法 CCF。CCF 算法统计每流的 ACK 平均到达间隔, 为间隔大的流设置较小

**基金项目:** 国家自然科学基金资助项目(60673164); 湖南省自然科学基金资助项目(06JJ10009); 湖南省教育厅科学研究基金资助项目(05C056)

**作者简介:** 李学勇(1972 -), 男, 副教授、博士研究生, 主研方向: 无线网络传输协议, 网络信息检索; 黄佳玮, 博士; 王建新, 教授、博士生导师

**收稿日期:** 2009-09-11 **E-mail:** xylicsu@163.com

的 ACK 包排队上限。如果当前某条 TCP 流的 ACK 排队数量超过其排队上限,算法就选择该流的 ACK 分组优先发送。由于 TCP 流的拥塞窗口大小和 ACK 分组到达间隔呈反比,因此小拥塞窗口流 ACK 分组的排队时间较少,从而获得更高的 ACK 分组的发送机会。由于小拥塞窗口 TCP 流降低了发生超时重传的概率,最终通过提高小拥塞窗口 TCP 流的吞吐量改进了上行 TCP 流之间的公平性。但该算法需要统计每流 ACK 分组的到达间隔,而大规模复杂网络中 ACK 分组到达呈现较大的抖动性和突变性,这就导致 CCF 根据 ACK 到达间隔来实施调度策略的可靠性下降。还有一些调度算法可以保证上下行 TCP 流的吞吐量公平<sup>[3-4]</sup>,但都无法解决上行 TCP 之间的不公平问题。

以上调度算法的设计目标都追求每流的吞吐量公平,而忽视了目前无线网络中普遍存在的多速率信道情况。在无线网络中,各无线节点处于不同的信道状态,信道自适应策略使各节点都采用不同的无线信道速率发送。为了保证吞吐量公平性,会将大部分信道分配给信道状态差的用户,从而降低了系统吞吐量。

### 3 UFWFQ 算法

#### 3.1 算法思想

通用处理器共享(Generalized Processor Sharing, GPS)<sup>[5]</sup>是最早的公平类调度算法模型,是关于延时性能和带宽公平性指标的理想策略。基于 GPS 模型,出现了众多基于分组的调度算法,其中最具有代表性的是 WFQ<sup>[6]</sup>。WFQ 算法的目的是将排队分组按流形成加权的轮询服务系统,从而保证各流之间的加权公平性。因此,在接入节点的下行缓存中对上行 TCP 流的 ACK 分组实施 WFQ 调度算法,能调节各流的 ACK 分组出队速率。同时,TCP 协议 ACK 分组具有“自同步”特性,最终可以实现上行 TCP 流的公平调度。但是在不同信道速率的流之间保证吞吐率的公平将导致网络总体吞吐量急剧下降。

因此,本文提出新的调度算法 UFWFQ,以各流的无线信道速率作为每流的服务权重。分配给高速发送流较大的权重,保证高速流能够发送更多的分组,从而在保证各流加权公平的同时提高网络的总吞吐量,实现效率和公平的理想折中。

#### 3.2 算法实现

在接入节点下行缓存的队列中,UFWFQ 调度算法对上行 TCP 流的 ACK 分组进行以下实施调度,其中,为缓存排队流的每条流分配了一个服务权重  $\phi_i$ ,而对流  $i$  的第  $k$  个分组  $p_i^k$  分别设置了服务开始时间  $S(p_i^k)$  和服务结束时间  $F(p_i^k)$ :

(1)  $S(p_i^k) = \max\{V(A(p_i^k)), S(p_i^{k-1}) + L_i^{k-1}/\phi_i\}$ , 其中,  $L_i^k$  是流  $i$  第  $k$  个分组的长度;  $A(p_i^k)$  是该分组的到达队列时间;  $V(t)$  是时间  $t$  所对应的虚拟时间。

(2)  $F(p_i^k) = S(p_i^k) + L_i^k/\phi_i$

(3)  $dV/dt = C(t)/\sum_{i \in B(t)} \phi_i$ , 其中,  $B(t)$  是在时间  $t$  的排队流集合;  $C(t)$  是在时间  $t$  的瞬时队列出口带宽。

(4) 在发送分组时,选择具有最小服务结束时间  $F(p_i^k)$  的分组出队列。

#### 3.3 权重计算

UFWFQ 调度算法依据各流的无线信道发送速率设置每

流的权重。但实际上在无线网络中,MAC 层 IEEE 802.11 协议的控制开销占用了相当长的信道时间。因此,设置权重还必须考虑发送分组的控制开销。如果忽略传输时延,无线网络中发送一个数据包的信道占用时间  $t$  由发送时延  $t_{tr}$  和控制开销时延  $t_{ov}$  组成:

$$t = t_{tr} + t_{ov} \quad (1)$$

其中,  $t_{tr} = S_i/r_i$ ,  $S_i$  为第  $i$  条流的包的大小;  $r_i$  为第  $i$  条流的无线信道发送速率。在采用 RTS/CTS 机制的情况下,假设不存在数据包碰撞,且节点的退避时间分布服从均匀分布,可得

$$t_{ov} = \frac{CW_{min} \cdot Slot + DIFS + 3 \cdot SIFS + \frac{RTS + CTS + ACK + PHY_{header}}{Basic\ rate}}{2} \quad (2)$$

由于 TCP 流中每一个 TCP DATA 分组都需要一个 ACK 分组确认,因此在计算发送 TCP 流有效速率时需要考虑 TCP DATA 分组和 ACK 分组及其控制开销。假设各流 ACK 的分组大小都相等,可以得到发送一对 TCP DATA 分组和 ACK 分组的有效速率  $T_i$  为

$$T_i = \frac{S\_DATA_i + S\_ACK}{\frac{S\_DATA_i + S\_ACK}{r_i} + 2t_{ov}} \quad (3)$$

最终设定流  $i$  的服务权重  $\phi_i = T_i$ , 以保证高速信道流得到更大的调度权重,使快速 TCP 流和慢速 TCP 流占用相等的信道时间,从而提高了网络总体效率。

### 4 仿真实验和性能分析

为了验证算法的有效性,利用 NS2<sup>[7]</sup>进行仿真实验。实验拓扑场景见图 1。

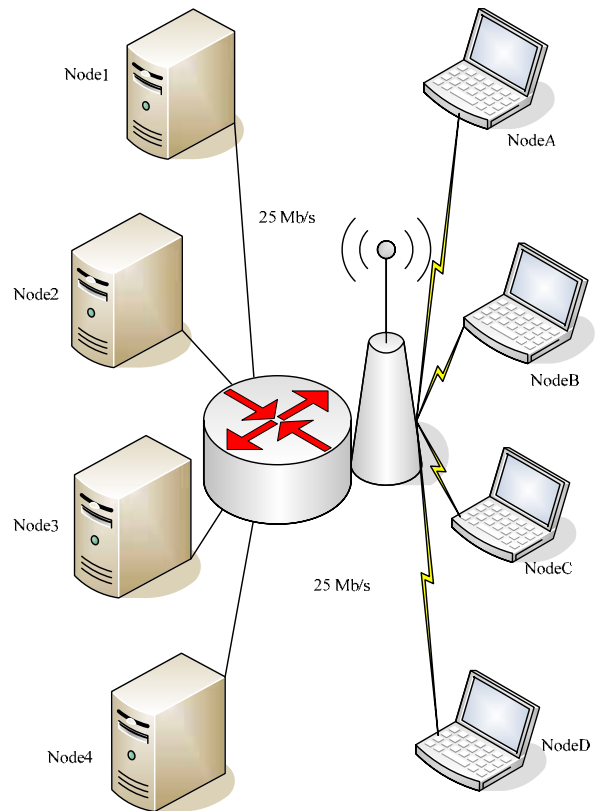


图 1 实验场景

实验设置的有线链路带宽都是 25 Mb/s，延时为 2 ms。AP 缓存容量为 100 个分组。设定 4 条上行 TCP NewReno 流 TCP1~TCP4，分别从无线节点 NodeA~NodeD 发送到有线节点 Node1~Node4。无线路由协议采用 NOAH(NO Ad-Hoc Routing)，MAC 层采用 IEEE 802.11b 协议。仿真实验分各流信道速率相等和不相等 2 种情况进行，同时对 FIFO、CCF 和 UFWFQ 调度算法进行模拟测试。

#### (1) 各 TCP 流信道速率相等

在本实验中，TCP1~TCP4 的无线信道速率都是 11 Mb/s。表 1 显示了在各流信道速率相等的情况下，4 条 TCP 上行流 TCP1~TCP4 的单流吞吐率和总吞吐率。

**表 1 TCP 流信道速率相等时各算法吞吐率比较** Kb/s

| 算法    | TCP1 | TCP2  | TCP3 | TCP4  | Total |
|-------|------|-------|------|-------|-------|
| FIFO  | 31   | 1 040 | 577  | 1 201 | 2 849 |
| CCF   | 720  | 715   | 723  | 724   | 2 882 |
| UFWFQ | 714  | 726   | 731  | 719   | 2 890 |

在 FIFO 中，各流的吞吐率相差很大，暴露出严重的上行 TCP 流间的不公平问题。其中，TCP1 几乎饿死；TCP4 的吞吐率高达 1 200 Kb/s。使用 CCF 作为 AP 的调度算法，各流的吞吐率基本相等，实现了上行流之间的吞吐率公平。在 UFWFQ 算法下，各 TCP 流的性能与 CCF 相同。这是因为在各流信道速率相等的情况下，UFWFQ 算法为每流设置的服务权重都相等，从而保证了各流吞吐率基本相等。

#### (2) 各 TCP 流信道速率不相等

表 2 显示了在 TCP4 的信道速率为 1 Mb/s、其他 TCP 流的信道速率为 11 Mb/s 的情况。

**表 2 TCP 流信道速率不相等时各算法吞吐率比较** Kb/s

| 算法    | TCP1 | TCP2 | TCP3 | TCP4 | Total |
|-------|------|------|------|------|-------|
| FIFO  | 401  | 735  | 651  | 842  | 2 629 |
| CCF   | 654  | 648  | 643  | 652  | 2 597 |
| UFWFQ | 912  | 904  | 921  | 212  | 2 949 |

在 FIFO 中，各流的吞吐率相差很大，仍然出现了上行 TCP 流不公平问题。CCF 中各流的吞吐率基本相等。但其反

作用在于 TCP4 降低了其他 TCP 流的吞吐率。这就暴露出多速率无线网络中吞吐率公平和网络总体效率之间的矛盾。在 UFWFQ 算法下，TCP1~TCP3 的吞吐率基本相等，而 TCP4 的吞吐率很小。原因在于 UFWFQ 算法为 TCP4 设置了较小的服务权重，降低了其信道发送概率。通过这种效率和公平性的折中，UFWFQ 取得了 3 种算法中最大的网络总吞吐率。

## 5 结束语

本文基于经典的 WFQ 调度算法，提出保证上行 TCP 公平的调度算法 UFWFQ。UFWFQ 在 AP 节点的下行缓存对上行 TCP 流的 ACK 分组实施调度，利用 ACK 分组的自同步机制，依据各流的无线信道速率动态调节，以保证上行 TCP 流之间的公平。

当各流的无线信道速率相同时，UFWFQ 保证各流的 ACK 分组具有相等的出队概率，实现了上行 TCP 流之间的吞吐率公平；当各流的无线信道速率不同时，UFWFQ 依据信道速率调节快慢速信道的 ACK 分组出队概率，实现了不同信道速率的上行 TCP 流之间信道时间公平。今后将在真实网络环境下进一步测试和改进算法性能。

## 参考文献

- [1] Keceli F, Inan I, Ayanoglu E. TCP ACK Congestion Control and Filtering for Fairness Provision in the Uplink of IEEE 802.11 Infrastructure Basic Service Set[C]//Proceedings of ICC'07. Glasgow, Scotland: IEEE Press, 2007.
- [2] Heusse M, Rousseau F, Sabbatel G B. Performance Anomaly of 802.11b[C]//Proceedings of INFOCOM'03, San Francisco, USA: IEEE Press, 2003: 836-843.
- [3] Wu Yi, Niu Zhisheng, Zheng Junli. Upstream Downstream Unfairness Issue of TCP over Wireless LANs with Per-flow Queuing[C]//Proceedings of ICC'05. [S. l.]: IEEE Press, 2005.
- [4] Ha Juho, Choi Chongho. TCP Fairness for Uplink and Downlink Flows in WLANs[C]//Proceedings of GLOBECOM'06. [S. l.]: IEEE Press, 2006.
- [5] Parekh A K, Gallager R G. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-node Case[J]. IEEE/ACM Transactions on Networking. 1993, 1(3): 344-357.
- [6] Demers A, Keshav S, Shenker S. Analysis and Simulation of a Fair Queuing Algorithm[C]//Proceedings of SIGCOMM'89. [S. l.]: ACM Press, 1989.
- [7] NS-2 Network Simulator[Z]. [2009-02-11]. <http://www.isi.edu/nsnam/ns/>.

编辑 张帆