

# 基于妥善排序的无线局域网功率控制策略

盛敏, 田野, 李建东, 江帆

(西安电子科技大学 综合业务网理论及关键技术国家重点实验室, 陕西 西安 710071)

**摘要:** 提出了一种基于妥善排序的功率控制策略, 将接入点缓存的业务数据按其到达顺序进行了时间上的有效排序, 通过合理的规范, 在保证数据可靠发送的同时, 有效地避免了处于休眠状态的节点激活后的竞争行为和不必要的退避, 减少各普通节点碰撞和退避所浪费的资源. 基于妥善排序的功率控制策略可以有效地改善系统的吞吐率和时延性能, 降低系统的功耗, 延长系统中各设备的有效工作时间.

**关键词:** 无线局域网; 功率控制; 妥善排序; QoS 保证

**中图分类号:** TN915.04 **文献标识码:** A **文章编号:** 1001-2400(2007)01-0001-05

## A perfect scheduling power control strategy for wireless LAN

SHENG Min, TIAN Ye, LI Jian-dong, JIANG Fan

(State Key Lab. of Integrated Service Networks, Xidian Univ., Xi'an 710071, China)

**Abstract:** The research on the power control strategy in Wireless LAN is a very important issue in the wireless communication field. In this paper, a Perfect Scheduling Power Control strategy PS\_PC is presented for decreasing the power consumption of equipments in WLAN, and it schedules the traffics buffered in Access Point (AP) according to their arrival time, so that the QoS guarantee can be provided and the consumed system resources by collision or backoff procedure are decreased. The theoretical analysis and simulation results show that the PS\_PC is an effective power saving method.

**Key Words:** Wireless LAN; power control; perfect scheduling; QoS guarantee

无线局域网<sup>[1~3]</sup> (WLAN) 是一种结合了无线通信技术和计算机网络优势的、可实现短距离通信的无线网络. 目前, WLAN 有代表性的标准是 IEEE802.11<sup>[1,4]</sup>, 它主要对物理层和链路层协议进行规范. 由于 WLAN 中的设备是以电池供电为主的便携式设备, 所以如何有效地减少 WLAN 网络中各设备的耗能, 从而延长网络设备的工作时间是目前 WLAN 领域中的一个研究热点. 笔者在 IEEE802.11 协议功率控制策略的基础上, 提出了一种基于妥善排序策略的功率控制方法 (PS\_PC), 有效地改善了网络的性能.

## 1 IEEE802.11 WLAN 有结构网络中功率控制过程存在的主要问题

在 IEEE802.11 WLAN 的 Infrastructure<sup>[1]</sup> 网络中, 所有的数据都必须经过 AP (接入点) 进行中转, 所以 AP 将一直处于激活状态, 而网络中其他 STA (站点) 可以进入休眠模式来减少不必要的电能消耗. AP 周期性地广播 Beacon<sup>[1]</sup> 帧 (信标帧) 来通知各休眠的 STA 是否有数据缓存在 AP 中. 但这样的工作模式 (具体过程可参考文献 [1,4]) 存在如下问题:

(1) STA 通过竞争方式向 AP 发送 PS\_POLL 帧<sup>[1]</sup>, 请求 AP 将其缓存的数据发给自己. 由于各 STA 采取的是竞争方式, 所以就存在 PS\_POLL 帧碰撞, 并导致节点退避的问题. 这种情况很容易造成 AP 缓存的数据因不能及时被取走而被 AP 丢弃, 同时由于存在竞争和退避, 在时延性能上会出现很大范围的波动. 这些问题使得 IEEE802.11 功率控制策略无法保证数据的 QoS 要求.

收稿日期: 2006-01-21

基金项目: 国家自然科学基金和微软亚洲研究院联合资助项目 (60572146, 60372048), 863 重大课题资助 (2005AA123910)

作者简介: 盛敏 (1975-), 女, 副教授, 博士.

(2) 在没有正确收到数据帧或得到 AP 将缓存的数据丢弃的准确信息之前,各 STA 一直处于激活状态,这就消耗大量的电能。

## 2 PS\_PC 功率控制策略

针对传统 IEEE802.11 有结构的 WLAN 网络中功率控制过程存在的主要问题,PS\_PC 算法有效地避免了多个处于休眠状态的 STA 在激活后同时发送 PS\_POLL 帧而产生的碰撞,并减少了退避过程所增加的时延,从而保证业务的 QoS 要求,同时降低了网络设备的电能消耗。

### 2.1 PS\_PC 算法描述

PS\_PC 算法中,在每个 Beacon 周期内,AP 根据记录各 STA 的 Listen Interval<sup>[1]</sup> (休眠时间)值,判断哪些处于休眠状态的 STA 将在下一个 Beacon 周期激活,并且确定是否为这些 STA 缓存数据。在此之后,AP 将对到达不同 STA 的数据进行排序。排序的原则是:按照产生的先后,对到达不同 STA 的第一个分组进行排序,然后再按照产生的先后,对到达同一 STA 的数据进行二次排序。实际上,AP 构造了一个虚拟的二维队列,如图 1 所示。纵向上到达不同目的 STA 的第 1 个分组相当于预约分组,在其排序预约后,AP 会将到达同一目的 STA 的数据横向顺序发送,这样可减少因多次预约所浪费的时间,提高系统的工作效率。

在 PS\_PC 策略中,AP 按照图 1 所示的虚拟队列进行顺序发送。同时处于激活状态的 STA 只需根据接收到的 Beacon 帧中的标识即判断是继续处于激活状态还是转入休眠状态,而无需主动竞争发送 PS\_POLL 帧。此外,由于 AP 对各 STA 缓存的数据进行了妥善排序,所以不存在各 STA 竞争发送 PS\_POLL 帧的问题,从而避免了 PS\_POLL 帧碰撞或退避造成的 AP 缓存数据的延时(或超时丢弃),也避免了发送 PS\_POLL 帧和退避时消耗的电能。

目的节点	分组序号	分组序号	分组序号	分组序号	分组序号
1	1	2	...	$N_1$	
3	1	2	3	...	$N_3$
6	1	2	...	$N_6$	
⋮	...	...	...	...	...
$m$	1	2	3	...	$N_m$

图 1 AP 构造的虚拟二维排序队列

### 2.2 PS\_PC 算法性能分析

#### 2.2.1 时延性能分析

根据上面对 PS\_PC 算法的描述,发现虚拟的发送队列形成了一个二维数组(如图 1 所示)。为了便于描述,用队列  $i$  来表示其中第  $i$  个队列(即到达同一目的 STA 的分组所构成的队列),用元素  $j$  来表示属于某一队列的第  $j$  个分组。假设系统中共有  $n$  个不同的 STA,每一个 STA 的数据流的到达过程是相互独立且服从参数为  $\lambda/n$  的 Poisson 过程,从而系统总的到达过程是一个参数为  $\lambda$  的 Poisson 过程,系统的服务能力为  $\mu$ 。将任意一个到达的分组表示为 Packet $[i, j]$ ,即该分组属于第  $i$  个队列,是该队列的第  $j$  个元素,则分组 Packet $[i, j]$  的等待时延可用图 2 表示。

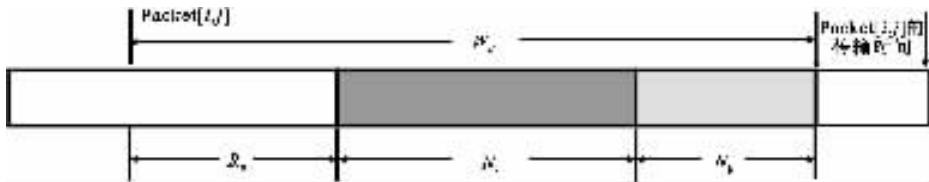


图 2 分组 Packet $[i, j]$  的等待时延示意图

在图 2 中,  $R_{ij}$  表示分组 Packet $[i, j]$  到达时的剩余休眠期,即分组 Packet $[i, j]$  产生时,其目的 STA 由休眠状态转入激活状态的剩余时间;  $N_i$  表示在队列  $i$  之前的所有队列中元素的数目;  $N_{ij}$  表示队列  $i$  中在元素  $j$  之前产生的元素数目;  $W_{ij}$  表示分组 Packet $[i, j]$  的等待时延;  $\mu$  为元素的发送速率。

由于帧间间隔以及相应的 Beacon 传输期是固定的,并且相对于整个休眠期而言较小,因此可以将其忽略,并能够得到

$$W_{ij} = R_{ij} + N_{ij}/\mu + N_i/\mu \quad (1)$$

在 PS\_PC 算法中,休眠期是一个固定的值,令其等于  $T$ ,则根据 Little<sup>[5]</sup> 定理,在一个休眠期内分组 Packet $[i, j]$  到达之前产生的分组个数为

$$N_{ij} = (T - R_{ij}) \times \lambda_i = (T - R_{ij}) \lambda / n \quad (2)$$

将式(2)代入式(1),并对式(1)取均值,可得

$$W = R + (T - R) \times \lambda / (\mu \times n) + Y \quad (3)$$

其中  $Y = E[N_i] / \mu$ . 由于休眠期是一个确定的区间,显然  $R = T/2$ , 因此式(3)可化简为  $W = (T/2)(1 + \lambda / (n\mu)) + Y$ .

下面计算在队列  $i$  之前产生的队列中所有元素对应分组产生的时延  $Y$ . 令  $P_{ik}$  表示在队列  $i$  之前有  $k$  个队列的概率;  $Y_{ik}$  表示在队列  $i$  之前有  $k$  个队列时,这  $k$  个队列中各个元素所对应的分组全部发送完毕所需要的时间. 由此  $Y = \sum_{k=1}^{n-1} P_{ik} \times Y_{ik}, k \neq i$ . 令  $\tau_i = T - R_{ij}$ ,只要在  $\tau_i$  区间内有其他队列产生(该队列至少有一个元素),那么根据 PS\_PC 策略,这些队列中的所有元素对应的分组都将在分组 Packet $[i, j]$  之前发送,所以只要讨论在区间  $\tau_i$  内是否有其他队列产生即可. 可以得到  $P_{ik} = C_{n-1}^k \prod_{l=1}^k [1 - p_l(0)] \prod_{a=1}^{n-(k+1)} p_a(0)$ , 其中  $p_l(j)$  表示第  $l$  个队列在区间  $\tau_i$  内有  $j$  个分组到达的概率. 该式表示有  $k$  个队列在队列  $i$  之前产生的概率,其中  $C_{n-1}^k$  表示有  $k$  种组合;  $\prod_{l=1}^k [1 - p_l(0)]$  表示有  $k$  个独立的队列在队列  $i$  之前有分组产生的概率;而  $\prod_{a=1}^{n-(k+1)} p_a(0)$  表示有  $n - (k + 1)$  个队列在队列  $i$  之前没有分组产生的概率. 由于每个队列分组的产生都服从参数为  $\lambda/n$  的 Poisson 过程,即  $p_l(s) = \left(\frac{\lambda}{n} \tau_i\right)^s \exp\left(-\frac{\lambda}{n} \tau_i\right) / s!$ ,  $s = 0, 1, \dots$ , 而  $p_l(0)$  表示第  $l$  个队列没有分组到达的概率,于是有  $p_l(0) = \exp\left(-\frac{\lambda \tau_i}{n}\right), l \in n$ . 由于所有队列的到达过程都是相同的,所以只讨论其可能产生队列的个数,而不讨论具体是哪一个队列.

同时,应用 Little 定理知  $Y_{ij} = \sum_{i=1}^j \frac{\lambda T}{n\mu} = j \times \frac{\rho T}{n}, \rho = \frac{\lambda}{\mu}$ . 综上所述,可以得到

$$Y = \frac{\rho T}{n} \sum_{k=1}^{n-1} k C_{n-1}^k \left[1 - \exp\left(-\frac{\lambda T}{2n}\right)\right]^k \left[\exp\left(-\frac{\lambda T}{2n}\right)\right]^{n-k-1}$$

式中由于  $R = T/2$ , 所以将区间  $\tau_i$  的均值近似为  $T/2$ . 则在 PS\_PC 算法中,任意一个分组的平均等待时延

$$W = \frac{T}{2} \left(1 + \frac{\rho}{n}\right) + \frac{\rho T}{n} \sum_{k=1}^{n-1} k C_{n-1}^k \left[1 - \exp\left(-\frac{\lambda T}{2n}\right)\right]^k \left[\exp\left(-\frac{\lambda T}{2n}\right)\right]^{n-k-1}$$

最终,可得任意一个分组的平均时延为  $T_{\text{average}} = W + T_{\text{trans}}, T_{\text{trans}} = l/\mu, l$  为分组的等效长度.

## 2.2.2 网络吞吐率性能分析

定义系统的单位系统时间为传输一个分组所需要的时间,即单位系统时间 = 分组的长度 / 信道的容量,则系统的吞吐率定义为:单位系统时间内成功传输的分组数,其单位为分组数每系统时间. 由于采用的是妥善排序的调度策略,并且在一个休眠期内产生的分组被保留一个休眠期的长度,所以只讨论在怎样的到达率情况下,在长度为一个休眠期的时间内可以处理的在上一个休眠期内到达的所有分组.

仍然令一个休眠期的长度为  $T$ ,它由  $m$  个 Beacon 周期组成,系统总的分组到达率为  $\lambda$ , 则根据 Little 定理,在一个休眠期中到达的总分组数(Total.Pkt)为  $\lambda \times T$ . 系统参数如表 1 所示.

表 1 系统参数

信道容量/(Mbit · s <sup>-1</sup> )	1	Beacon 周期/ms	2 048Slot ≈ 102 ms
时隙 Slot 长度/μs	50, 1slot	SIFS 长度/μs	28, 3slot
分组长度/bit	8 184, 164slot	DIFS 长度/μs	128
ACK 长度/bit	128, 3slot	PS_POLL 长度/bit	160
Beacon 帧长/Byte	110, 18slot	分组到达率/(分组数每系统时间)	$\lambda$

在下一个处理的  $T$  时间内至多可处理的分组数(Deal.Pkt)为(一个休眠期的长度 -  $m$  个 Beacon 周期) / (分组数 + SIFS 长度 + ACK 长度 + DIFS 长度). 显然,系统应满足在一个休眠期内到达的总分组数小于在下一个处理的  $T$  时间内可处理的分组个数. 根据表 1 有  $\lambda T \leq (T - m \times 18) / (164 + 3 + 6)$ , 推导可得系统

的最大饱和吞吐率与系统中的节点数无关,并且与休眠期的长度也无关.最终解得系统总的吞吐率为 0.939 345 分组数每系统时间.很显然,该值包括了系统中所有的开销,将其化简到纯数据吞吐率时,有效值(Throughput\_MAX)为  $0.939\ 345 \times 164/173 \approx 0.88$  分组数每系统时间.

### 3 PS\_PC 算法仿真

采用的仿真参数如表 1 所示,到达过程服从泊松分布.为了便于描述,将传统的 WLAN 功率控制策略用 O\_PC 表示,而基于妥善排序的 WLAN 功率控制策略用 PS\_PC 表示.在仿真中的一个很重要的系统参数是平均休眠期,是指网络中所有节点在 AP 中注册的 Listen\_Interval 的平均值,用  $m$  表示平均休眠期是  $m$  倍的 Beacon 周期.这里,当分组缓存的时间从其第一次激活开始达到  $m$  个 Beacon 周期时,分组将被 AP 丢弃.

#### 3.1 系统吞吐率和时延性能

系统的吞吐率是指单位时间内成功传输的分组数,在这里定义其单位为分组数每系统时间.平均时延是指分组从产生到成功传输所需要的总时间.采用有 10 个处于休眠状态的 STA 和一个 AP 的网络场景(网络中的 STA 数目大于等于 10),并讨论了不同休眠期对系统性能的影响,其仿真结果如图 3 和图 4 所示.

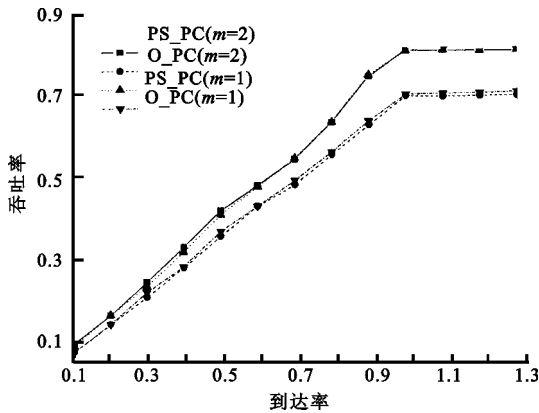


图 3 吞吐率曲线图

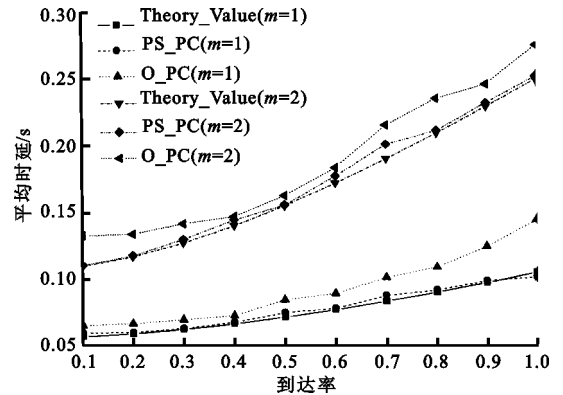


图 4 平均时延曲线

从图 3 可以看出,无论是传统的 WLAN 功率控制策略还是 PS\_PC 功率控制策略,由于都是基于点到多点的有中心控制的工作方式,所以系统的吞吐率较随机竞争方式下的都要好很多,并且 PS\_PC 算法的吞吐率明显优于 O\_PC 系统的吞吐率,PS\_PC 策略下系统的最大吞吐率比 O\_PC 的最大吞吐率性能好近 11.44%。随着到达率的持续增长,系统的吞吐率性能呈线性增长,当到达率超过系统可以处理的最大容量时,系统的吞吐率曲线开始趋于平缓.从图 3 也可以看到,PS\_PC 策略下系统最大的吞吐率可以到达 86%左右,和理论分析值基本吻合.仿真也表明了当采用 PS\_PC 的缓存策略时,吞吐率与休眠期的长短无关.

图 4 是系统的平均时延曲线.由图可见,PS\_PC 算法的性能优于 O\_PC,当平均休眠期为 1 倍和 2 倍 Beacon 周期( $m=1$  和  $m=2$ )时,PS\_PC 比 O\_PC 的平均时延性能分别改善了约 14.5 ms 和 14.1 ms;并且随着休眠期的增长,系统的平均时延也有所增长,这是由于剩余休眠期长度的增加而使得整个平均时延增加的缘故.从图 4 可以看出 PS\_PC 算法的仿真曲线和理论分析曲线基本吻合,但略高于理论分析曲线.这是因为理论分析模型忽略了所有的帧间间隔和 Beacon 周期所消耗的时间,从而使理论分析为系统性能的保守估计.图 4 中,当休眠期  $m$  分别为 1 和 2 时,PS\_PC 算法的仿真值和理论值的平均偏差分别为 0.1% 和 0.27%.

#### 3.2 节能性能

为了分析 PS\_PC 功率控制策略和 IEEE802.11WLAN 中传统的功率控制策略的耗电性能,采用了 Cisco AIR-PCM350 无线网卡的耗电参数<sup>[6]</sup>(见表 2),通过仿真进行比较.

表 2 的参数表明在不同的状态下每秒钟消耗的电能情况.在仿真过程中,笔者仅关心两种策略下传输相同分组时系统的耗电情况.所讨论的网络中存在 10 个处于休眠期的普通 STA 和一个 AP,研究系统处于饱和状态下(每秒钟到达 100 个分组)的耗电性能,结果如图 5 和图 6 所示.

表 2 Cisco AIR-PCM350 网卡耗电参数表

网卡型号	休眠时的耗电量/mW	空闲时的耗电量/mW	接收时的耗电量/mW	发送时的耗电量/mW
Cisco AIR-PCM350	45	1 080	1 300	1 875

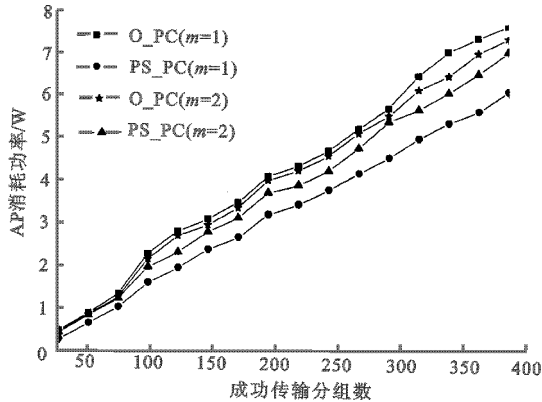


图 5 AP 的耗电情况

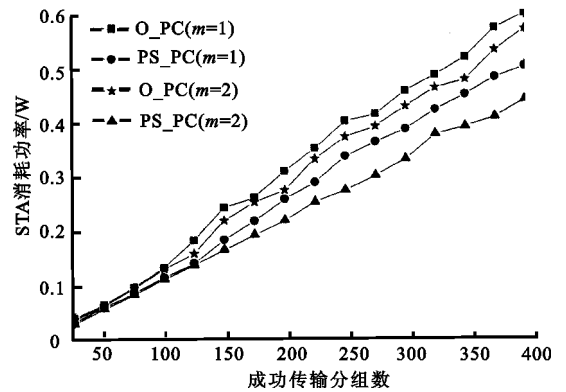


图 6 STA 的耗电情况

从图中可以看到,无论如何选择平均休眠期,PS\_PC 策略的节能性能都优于 O\_PC,并且 AP 和 STA 消耗的电能会随着分组数的增多而增多.对于 PS\_PC 功率控制策略,平均休眠期 Listen\_Interval 值较大时,AP 传输同样数量的分组消耗的电能相对较多.这是因为此时对于同样到达率的情况,发送同样数目的分组,除了固定需要的发送分组和接收应答所消耗的电能外,AP 空闲消耗的电能较大.但对于 STA 的平均耗能情况却恰恰相反,这是因为 STA 会在无数据接收后处于休眠状态,越早关闭收发信机,消耗的电能就越小.所以对于 STA 而言,平均休眠期值比较大的情况下,消耗的电能较少.从图中还可以看到当平均休眠期等于 1 倍 Beacon 周期时,两种算法性能接近,PS\_PC 策略略好于 O\_PC 策略.而当平均休眠期等于 2 倍 Beacon 周期时,PS\_PC 功率控制策略消耗的电能明显低于 O\_PC.这是因为平均休眠期越长,每一次 AP 缓存等待发送的分组数越多,各 STA 的竞争就越激烈,O\_PC 功率控制策略因 PS\_POLL 帧碰撞和退避所消耗的电能就会明显增多.

## 4 结束语

笔者提出了一种基于妥善排序的新型功率控制策略 PS\_PC. PS\_PC 将缓存在 AP 中的数据按到达先后进行了妥善排序,从而避免了处于休眠状态的节点在激活后的竞争行为和不必要的退避,有效地改善了系统的吞吐率和时延性能,降低了系统功耗,延长了系统中各设备的有效工作时间.

### 参考文献:

- [1] IEEE Standard 802.11b. Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specification; High-speed Physical Layer in the 2.4GHz band[S]. 1999.
- [2] He Hong, Li Jiandong, Sheng Min, et al. Performance Analysis of a Novel Multiple Access Algorithm for the Wireless LAN [J]. Journal of Xidian University, 2005, 32(1): 26-30.
- [3] Natkaniec M. Wireless LAN Standards and Applications [J]. IEEE Communications Magazine, 2002, 40(2): 32-34.
- [4] IEEE Standard 802.11a. Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specification; High-speed Physical Layer in the 5GHz band[S]. 1999.
- [5] Bertsekas D, Gallager R. Data Network [M]. Indiana: Prentice-Hall, 1992.
- [6] Shih E, Bahl V, Sinclair M. Reducing Energy Consumption of Wireless, Mobile Devices Using a Secondary Low-Power Channel[R]. Cambridge: MIT Laboratory for Computer Science, 2003.

(编辑:郭 华)