

# 无线局域网吞吐量优化算法

杨义蛟<sup>1</sup>, 赵虹<sup>2</sup>, 鲁五一<sup>3</sup>

(1. 长沙理工大学网络中心, 长沙 410004; 2. 中南大学湘雅二医院信息中心, 长沙 410011; 3. 中南大学信息科学与工程学院, 长沙 410083)

**摘要:** 针对无线局域网吞吐量较小且稳定性较差的问题, 分析影响无线网络吞吐量性能的因素, 提出无线局域网吞吐量优化算法, 采用马尔科夫链理论建立吞吐量计算模型, 运用动态调整竞争窗口、优化数据帧长度、改进退避时间的策略。NS2 仿真结果表明, 该算法能有效提高无线局域网的吞吐量。

**关键词:** 无线局域网; 吞吐量; 马尔科夫链; 优化

## Throughput Optimization Algorithm for Wireless Local Area Networks

YANG Yi-jiao<sup>1</sup>, ZHAO Hong<sup>2</sup>, LU Wu-yi<sup>3</sup>

(1. Network Center, Changsha University of Science and Technology, Changsha 410004; 2. Information Center, The Second Xiangya Hospital, Central South University, Changsha 410011; 3. School of Information Science and Engineering, Central South University, Changsha 410083)

**【Abstract】** The small and unstable throughput is one of main aspects to limit the development of wireless local area networks. This paper proposes a kind of throughput optimization model by the Markov chain theory and designs series of throughput optimization arithmetic based on analyzing the mechanism of affecting the attribute of wireless local area networks throughput. The new throughput optimization algorithm adopts the measures of dynamic adjusting competition window value, mends the length of data frame and improves the strategy of retreat time. The simulation result by NS2 indicates that it can enhance the performance of wireless local area networks throughput effectively.

**【Key words】** wireless local area networks; throughput; Markov chain; optimization

为提高无线局域网吞吐量, 文献[1]提出根据信道负载调整不同QoS等级站点接入参数; 文献[2]提出采用根据负载情况自适用调整的分布式协调功能(Distributed Coordination Function, DCF)算法; 文献[3]考虑改变数据包长度的影响而提出性能评估和改进。本文针对无线局域网吞吐量标准算法的缺陷, 提出一种吞吐量优化模型及改进算法。

### 1 无线局域网吞吐量标准算法

#### 1.1 虚拟载波监听技术

用网络分配矢量(Network Allocation Vector, NAV)指示网络忙闲状态, 并表示介质空闲剩余时间值。每个发送站在发送帧时估计 NAV 并装入帧头, 其他站接收到此帧后如发现本地 NAV 小于此 NAV, 则用此 NAV 更新本地 NAV, 无线客户端通过检查 NAV 决定是否发送。

#### 1.2 随机退避机制

标准算法通过竞争窗口指数退避来实现不同站点的异步自适应接入, 退避时间选取为

$$T = W \times \text{Random}() \times \text{SlotTime}$$

其中,  $\text{Random}()$  是随机数;  $\text{SlotTime}$  是总传播时延;  $W$  是竞争窗口, 为  $\text{SlotTime}$  的整数倍。当一帧进入发送缓存时,  $W$  初始化为  $\text{SlotTime}$ , 在每次尝试重传后  $W$  加倍直至  $W_{\max}$ 。当多个站延迟并进入随机退避状态后, 退避值最小的站将在竞争中获得介质访问权。在竞争中失败的站保持退避状态直到下个 DIFS, 如此可避免某个站永远不能获得介质访问权。

#### 1.3 帧优先级的设置

发送帧之间有一定间隔, 当介质空闲一定时间后才能

访问介质。无线局域网中有 4 种长度由小到大的帧间隔: SIFS, PIFS, DIFS, EIFS, 不同优先级的帧获得相应的介质访问优先权。

### 2 无线局域网吞吐量优化模型

**定义 1**  $v(t)$  为给定站的时间计数器值的随机过程,  $s(t)$  为给定站在时刻  $t$  所在的回退阶段(0, 1, ...,  $m$ ) 的随机过程。采用离散时间刻度,  $t$  对应着时隙的开始, 每个站的回退计数器在每个时隙开始时递减。

假定理想信道条件(即没有隐藏终端和信道捕获), 丢包仅由冲突引起; 系统工作于饱和状态, 每个站的传输队列保持非空, 每个站都有包等待传输; 不论所经历的重传次数为多少, 在每次传输中, 每个包传输的冲突概率恒定为常数  $p$ 。

在 DCF 竞争过程中, 每次传输时帧碰撞概率和过去碰撞次数无关, 其碰撞概率  $p$  恒定且相互独立<sup>[4]</sup>, 用时间离散的二维马尔科夫过程  $\{s(t), b(t)\}$  描述, 其非空一步转移概率为

$$P\{0, k | i, 0\} = (1-p)/W_0, \quad i \in [0, m], k \in [0, W_0-1] \quad (1)$$

$$P\{i, k | i, k+1\} = 1, \quad i \in [0, m], k \in [0, W_i-2] \quad (2)$$

$$P\{i, k | i-1, 0\} = p/W_i, \quad i \in [1, m], k \in [0, W_i-1] \quad (3)$$

$$P\{m, k | m, 0\} = p/W_m, \quad k \in [0, W_m-1] \quad (4)$$

其中, 式(1)表示在任何回退阶段, 包传输成功概率为  $1-p$ ,

**作者简介:** 杨义蛟(1974—), 男, 实验师、博士研究生, 主研方向: 计算机网络; 赵虹, 硕士; 鲁五一, 教授、硕士

**收稿日期:** 2008-12-04 **E-mail:** yyj@csust.edu.cn

回退时隙在 $(0, W_0-1)$ 内一致分布；式(2)表示时间计数器值在每个时隙的开始递减；式(3)表示在任何回退阶段，包传输失败概率为 $p$ ，回退时隙在 $(0, W_i-1)$ 内一致分布；式(4)表示回退阶段到达 $m$ ，传输失败后时隙保持不变。令

$$b_{i,k} = \lim_{t \rightarrow \infty} P\{s(t)=i, v(t)=k\}, i \in [0, m], k \in [0, W_i-1]$$

其中， $b_{i,k}$ 为马尔科夫链(Markov chain)的稳态分布，得

$$b_{i,k} = \frac{W_i-k}{W_i} b_{i,0}, i \in [0, m], k \in [0, W_i-1] \quad (5)$$

$$\sum_{i=0}^m \sum_{k=0}^{W_i-1} b_{i,k} = 1 \quad (6)$$

**定义 2**  $\tau$  为一个站在随机选取的时隙传输包的概率。由于所有的站只有在回退计数器为 0 时才传输包，得

$$\tau = \sum_{i=0}^m b_{i,0} = \frac{b_{0,0}}{1-p} \quad (7)$$

回退窗口  $W_i=2^i W_0, i \in [0, m], W_{\max}=2^m W_0$ ，代入式(6)得

$$\sum_{i=0}^m \sum_{k=0}^{W_i-1} b_{i,k} = \sum_{i=0}^m b_{i,0} \sum_{k=0}^{W_i-1} \frac{W_i-k}{W_i} = \sum_{i=0}^m b_{i,0} \frac{W_i+1}{2} = \frac{1}{2} \left( \sum_{i=0}^{m-1} b_{i,0} W_i + b_{m,0} W_m + \sum_{i=0}^m b_{i,0} \right) = \frac{b_{0,0}}{2} \left( \sum_{i=0}^{m-1} p^i 2^i W_0 + \frac{p^m}{1-p} 2^m W_0 + \frac{1}{1-p} \right) = 1 \quad (8)$$

由式(7)和式(8)得

$$\tau = \frac{2}{1-W_0 + pW_0 \sum_{i=0}^{m-1} (2p)^i} \quad (9)$$

由于所有其他  $n-1$  个站都不传输数据包的概率即是不发生冲突的概率，因此有

$$p = 1 - (1-\tau)^{n-1} \quad (10)$$

由式(9)和式(10)，可得  $\tau$  和  $p$ 。

**定义 3**  $P_s$  为数据包成功传输的概率， $P_n$  为数据包不传输的概率， $P_c$  为数据包传输但发生冲突的概率，则

$$P_s = n\tau(1-\tau)^{n-1} \quad (11)$$

$$P_n = (1-\tau)^n \quad (12)$$

$$P_c = 1 - P_s - P_n \quad (13)$$

**定义 4**  $T_s$  为成功传输包平均的信道忙时间， $T_c$  为包传输冲突平均的信道忙时间， $\sigma$  为时隙时间。

DCF 帧传输方式有基本和 RTS/CTS 这 2 种接入方式，DCF 时隙图见图 1。

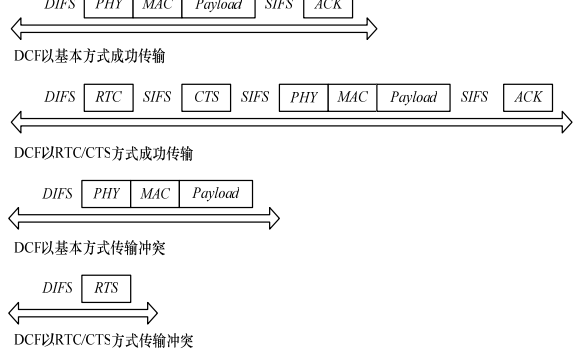


图 1 DCF 时隙图

令  $H=PHY_{\text{header}}+MAC_{\text{header}}$ ，传播延迟为  $\delta$ ， $E[P]$  表示一个时隙中有效载荷期望值，则有

$$T_s^{\text{基本}} = DIFS + H + E[P] + \delta + SIFS + ACK + \delta \quad (14)$$

$$T_c^{\text{基本}} = DIFS + H + E[P] + \delta \quad (15)$$

$$T_s^{\text{RTS/CTS}} = DIFS + RTS + \delta + SIFS + CTS + \delta + SIFS + H + E[P] + \delta + SIFS + ACK + \delta \quad (16)$$

$$T_s^{\text{RTS/CTS}} = DIFS + RTS + \delta \quad (17)$$

信道状态由多个工作周期重复出现构成，吞吐量  $S$  为 1 个时隙中传输的有效载荷，有

$$S = \frac{P_s E[P]}{P_s T_s + P_n \sigma + P_c T_c} = \frac{E[P]}{T_s - T_c + \frac{P_n \sigma + (1-P_n) T_c}{P_s}} \quad (18)$$

将式(11)~式(17)代入式(18)，即可得  $S$ 。

### 3 无线局域网吞吐量优化算法

#### 3.1 动态调整竞争窗口初始值

标准算法中初始竞争窗口大小固定不变。当活动节点少时，能避免产生冲突。但随着活动节点数增多，冲突概率大大增加。初始竞争窗口值太大，虽然能避免冲突，但当活动节点很少时，大量时间用于产生随机回退数后的等待，降低吞吐率。因此，初始竞争窗口大小应随活动节点数动态调整。

##### 3.1.1 初始值计算

假设每个活动节点都有数据包发送，延迟表示同一节点连续成功传输 2 个数据帧的时间间隔，用  $D$  表示：

$$D = nT_s + \frac{1-(1-\tau)^n - n\tau(1-\tau)^{n-1}}{\tau(1-\tau)^{n-1}} T_c + \frac{1-\tau}{\tau} \sigma \quad (19)$$

在竞争信道时，如果某节点在一定的时间间隔  $ATO$  内没有发送任何数据包，就认为它不再是活动节点。 $ATO$  太小时，很多节点不能在一个  $ATO$  时间段内完成一个数据帧的传输，估计出的活动节点数小于实际值。如果  $ATO$  太大，当活动节点减少时，要在一个  $ATO$  时间后才能被发现，延迟过长。所以， $ATO$  的值应随活动节点数变化而变化。

$ATO$  取  $D$  的整数倍，由式(19)可知， $D$  大致上随  $n$  线性增长。因此，有

$$ATO = k \times n, n \geq 1 \quad (20)$$

其中， $k$  为常量。

对式(19)自变量求导，得到不同竞争窗口初始优化值  $W_{opt\_fir}$  和活动节点数  $n$  之间的关系：

$$W_{opt\_fir} \approx n \sqrt{2T_c / \sigma} \quad (21)$$

节点每监听到一个帧，首先更新其  $ATO$  表，然后由式(20)估算活动节点数，再由式(21)调整初始竞争窗口大小到最优值。

##### 3.1.2 初始值修正

从时刻  $t$  到  $t+T$ ，站点  $x$  发送数据包总数  $E[s_x(t+T)]$  与发生碰撞数据包总数  $E[c_x(t+T)]$  之比表示信道负载，即

$$f_x(t+T) = \frac{E[c_x(t+T)]}{E[s_x(t+T)]} \quad (22)$$

考虑信道误码丢包和数据包长度，式(22)修正为

$$f_x(t+T) = \frac{E[\bar{c}_x(t+T)]}{E[\bar{s}_x(t+T)]} \quad (23)$$

其中， $f_x(t+T)$ ， $\bar{c}_x(t+T)$ ， $\bar{s}_x(t+T)$  分别表示站点  $x$  在时间  $t \sim t+T$  信道平均负载、未成功传输字节总数和发送字节总数。

为减少突发碰撞或信道恶化而造成  $f_x(t+T)$  波动太大，对其进行平滑计算：

$$\bar{f}_x(t+T) = \alpha \cdot \bar{f}_x(t) + (1-\alpha) f_x(t+T) \quad (24)$$

其中， $\alpha$  为平滑系数； $\bar{f}_x(t)$  表示上一个周期内负载平均值； $\bar{f}_x(t+T)$  表示该段时间内负载平均值。

竞争窗口最优初始值  $W_{opt}$  为

$$W_{opt} = (n \sqrt{2T_c / \sigma}) / d \quad (25)$$

其中， $d$  为负载系数，通常取  $d = \min(\bar{f}_x(t+T), 0.8)$ 。

#### 3.2 优化数据帧长度

信道重负载时，数据帧长度越长，信道中发生碰撞的概率越高，数据帧误包率上升，数据帧重传增多，吞吐量和时延性能越差。在信道误码率为  $q$ ，数据帧 MAC 头长度为  $h$  的条件下满足系统性能最优(时延小，吞吐量大的)数据帧长度为

$$L_{opt} = \frac{-h \ln(1-q) - \sqrt{-4h \ln(1-q) + h^2(1-q)}}{2 \ln(1-q)} \quad (26)$$

由于信道误码率不确定，每次发送的数据帧长度并不能保证是最优长度。同时数据帧长度突变，可能产生较大延迟抖动，因此数据帧长度变化不宜太快。具体算法描述如下：

(1)当源站点在规定时间内没有收到数据确认帧时，如果当前发送数据帧的长度  $L_k > L_{opt}(t+T)$ ，则下次发送数据帧的长度  $L_{k+1} = L_k/2$ ，但不小于数据帧的最小长度  $L_{min}$ ；如果  $L_k < L_{opt}(t+T)$ ，则  $L_{k+1} = L_k + \zeta$ ， $L_{k+1} \geq L_{min}$ ， $\zeta$  为一正整数。

(2)当源站点在规定时间内收到数据确认帧时，如果  $L_k < L_{opt}(t+T)$ ，则  $L_{k+1} = 2L_k$ ，但不大于数据帧的最大长度  $L_{max}$ ；如果  $L_k > L_{opt}(t+T)$ ，则  $L_{k+1} = L_k + \zeta$ ， $L_{k+1} \leq L_{max}$ ， $\zeta$  为正整数。

### 3.3 改进退避时间策略

信道不成功传输后，标准算法中  $W$  呈指数增长，这在冲突并不严重时将造成信道空闲时间过多。因此，对其进行改进：当一个站点有数据要传输时便监测信道，如果信道空闲达到 DIFS 便传输该数据帧，否则持续监测。在传输数据帧前产生一个随机的回退间隔，以最小化与其它站点包传输冲突的概率，回退时间为  $[0, W-1]$  一致分布。 $W$  的值依赖于该包传输失败的次数。第 1 次传输时  $W$  设为最小竞争窗  $W_{min}$ ，每次不成功传输后  $W$  加倍。当回退次数达到  $l$  时(回退窗口达到阈值)，每次不成功传输后  $W$  线性增加，斜率为  $r$ ，直到最大值  $W_{max} = 2^l W_{min} + r(m-l)$ ，以后保持不变。如果数据成功传输，则  $W$  复位为  $W_{min}$ 。当监测空闲超过 DIFS 时，回退时间计数器重新激活并开始递减，在检测到其他站传输时计数器暂停。计数器到 0 的站点开始传输数据。即

$$\begin{cases} W_i = 2^i W_0 & i \in [0, l] \\ W_i = W_l + r(i-l) & i \in [l, m] \end{cases} \quad (27)$$

将式(27)代入式(9)，得

$$\tau = \frac{2}{1 + p^m (2^l W_0 + r(m-l)) + (1-p) \sum_{i=l}^{m-1} p^i (2^l W_0 + r(i-l)) + (1-p) W_0 \sum_{i=0}^{l-1} (2p)^i} \quad (28)$$

由式(10)和式(28)，可得  $\tau$  和  $p$ ，代入式(18)，可得  $S$ 。

## 4 仿真及结果分析

假设信道误码率已知，所有站点可达，不考虑隐藏和暴露终端，每个活动节点都处于饱和状态，只要发生碰撞和误码就意味着数据帧发送失败，缓存区足够大<sup>[5]</sup>。

仿真实验采用 NS2 进行吞吐量优化，设有 50 个节点，随机分布在 200 m × 200 m 范围内，仿真时间为 200 s。仿真参数见表 1。

表 1 仿真参数

参数	取值	参数	取值
PHY	DSSS	MAC_header	34 Byte
$W_{min}$	32	PHY_header	16 Byte
$W_{max}$	1 024	ACK	304 bit
SIFS	10 $\mu$ s	RTS	352 bit
DIFS	50 $\mu$ s	CTS	304 bit
SlotTime	20 $\mu$ s	数据包	0~2 312 Byte
传输速率	2 Mb/s	$k$	72 ms

吞吐量与站点数的关系如图 2 所示。随着站点数增加，

冲突概率增加，吞吐量降低。

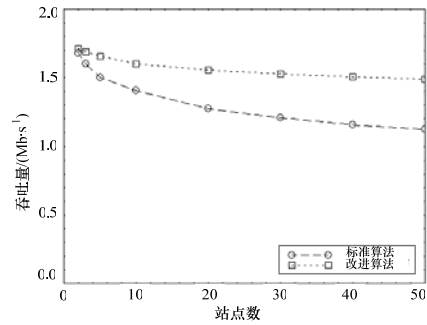


图 2 吞吐量与站点数的关系

当站点数为 2 时，吞吐量为 1.68 Mb/s。当站点数为 50 时，标准算法吞吐量为 1.12 Mb/s，优化算法吞吐量为 1.49 Mb/s。可见优化算法可有效抑制吞吐量随站点数增加而降低的速度。其原因是当节点很多时，优化算法成倍增大了初始竞争窗口值，降低了冲突概率，从而大部分数据包可在一个 ATO 时间内完成传输。而且估计到的活动节点数又可更准确计算 ATO 值，使下次估算的活动节点数更准确。

吞吐量与数据包长度的关系如图 3 所示。

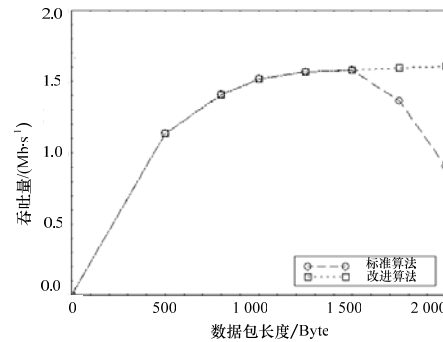


图 3 吞吐量与数据包长度的关系

当数据包长度小于 1 500 Byte 时，2 种算法的吞吐量一致，且单调增加。这是因为小包和大包消耗相同的竞争开销，但小包效率较低。但当数据包长度为 1 750 Byte 时，优化算法吞吐量为 1.60 Mb/s，继续单调增加，而标准算法吞吐量为 1.37 Mb/s，比数据包长度为 1 500 Byte 时降低 0.21 Mb/s，当数据包长度为 2 000 Byte 时，这种变化更明显。可见优化算法根据最优数据包长度值调整实际传输数据包长度，能避免因数据包长度过长而造成冲突增加、吞吐量降低。

优化算法中吞吐量与参数  $r$  的关系见图 4。 $r$  对优化算法的吞吐量影响相对较小。达到阈值后斜率  $r$  的选取可防止在严重冲突时更大幅度地增加等待时间而造成带宽浪费。

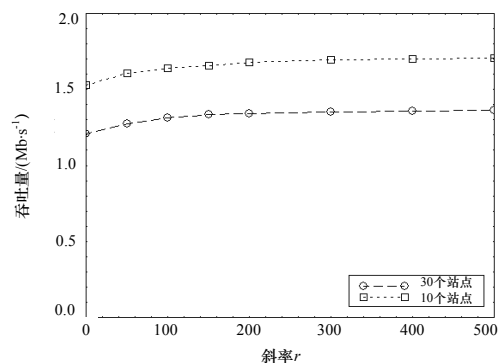


图 4 吞吐量与斜率  $r$  的关系

(下转第 115 页)