文章编号：1000-8608(2016)01-0041-09

基于信道占用率的 Ad Hoc 网络自适应公平性算法

赖晓晨[1,2]， 刘全利*[1]， 任志磊[2]， 赵 莹[2]

（1.大连理工大学 电子信息与电气工程学部，辽宁 大连 116024；
2.大连理工大学 软件学院 辽宁省泛在网络与服务软件重点实验室，辽宁 大连 116620）

摘要： Ad Hoc 网络中，节点通过竞争信道完成通信，如果竞争窗口值选取策略不合理，则容易造成公平性问题.在分析现存典型公平性算法的实现机制基础上，提出了信道占用率的概念，设计了一种基于信道占用率的 Ad Hoc 网络自适应公平性算法.根据节点对通信历史及当前信道占用率与理想信道占用率之间的关系，将通信情况分为 4 种类别，再结合当前网络负载情况，动态设置竞争窗口值.仿真结果表明，该算法在改善吞吐量的同时，显著提高了节点信道接入公平性，在各种负载条件下表现良好，优于 BEB、MILD、MIMD 和 NAVB 等算法.

关键词： Ad Hoc 网络；公平性；竞争窗口；吞吐量

中图分类号： TP393.1 **文献标识码：** A **doi：** 10.7511/dllgxb201601007

0 引 言

Ad Hoc 网络中，随着接入节点增多，节点对信道的竞争趋向激烈，导致不同节点获得的通信机会并不相等，即产生所谓公平性问题.公平性反映了不同的用户、节点或应用共享信道的公平程度，实现公平性是无线网络资源分配的主要目标之一.让每个节点都平等地接入信道，获得平等的带宽是一个非常重要的任务.当公平性得不到保证时，成功发送数据的节点总是可以得到更多的信道占用机会，而失败的节点一直处于忙等状态，出现“饿死”状况，导致现实中有些数据流会因为其他流的恶意破坏而失去发送机会，这在军事通信和紧急医疗通信中是不允许出现的，所以实现公平性显得格外重要.

Ad Hoc 网络中，节点接入信道的机制由 MAC(media access control) 协议实现，信道获取能力主要取决于各节点的退避时间，因此退避算法起着关键性的作用.如何合理设计退避算法，在不显著降低网络吞吐量的同时提高节点通信的公平性，是一直以来研究的热点问题.

IEEE 802.11 无线网络协议 MAC 中采用 DCF(distributed coordination function) 机制实现信道无线接入，基于 DCF 的公平性改进策略主要调整协议中的二进制指数退避(BEB，binary exponential backoff) 算法，这是目前网络公平性研究的主要方法.本文提出信道占用率(channel occupancy rate) 的概念，在单冲突域中，根据发包间隔内节点对占用信道的时间比例与理想信道占用率间的关系，在不同网络负载情况下自适应调整窗口退避值，减少碰撞概率，在保证网络吞吐量的同时，提高节点接入信道的公平性.

1 研究现状

1.1 退避算法

应用在 IEEE 802.11 MAC 协议中的二进制指数退避算法能有效解决节点竞争信道的冲突问题[1].其核心思想是节点的退避时间反映了网络中不同节点接入信道的先后顺序，退避时间较短的节点比退避时间较长的节点获得更大的信道接入机会.初始时节点选取竞争窗口(collision window) 的最小值 W_{min}，如果数据传输失败，则将 W 增加一倍，直至达到竞争窗口的最大值 W_{max}；如果数据发送成功，则重新将 W 置为最小值

收稿日期： 2015-04-10； **修回日期：** 2015-09-28.
基金项目： “九七三”国家重点基础研究发展计划资助项目(2013CB035906)；国家自然科学基金资助项目(61273037，61403057).
作者简介： 赖晓晨(1977-)，男，副教授，E-mail：laixiaochen@dlut.edu.cn；刘全利*(1976-)，男，教授，博士生导师，E-mail：liuql@dlut.edu.cn.

$W_{\min}$,W 取值如式(1)所示[2]:

$$W=\begin{cases}\min(W\times 2,W_{\max}); & \text{碰撞}\\ W_{\min}; & \text{成功传输}\end{cases} \tag{1}$$

$W_{\max}$ 与 $W_{\min}$ 的定义是为了约束 W 变化的范围.W 确定之后,节点在 0 和 W 之间随机选择一个整数,即退避窗口,用此整数乘以物理层时隙长度 s,可以得到退避时间 t,如式(2)所示[3]:

$$t=Random(0,W)\times s \tag{2}$$

由上可知,BEB 算法简单、高效,但是缺点也十分明显:总是倾向于将信道使用权交给最近成功传输信息的节点,造成信道竞争不公平的现象.

为了解决上述不公平问题,Bharghavan 等在 MACAW(multiple access with collision avoidance for wireless)协议中提出了倍数增加线性减少(MILD,multiplicative increase linear decrease)算法[4],按照式(3)选取竞争窗口值[2]:

$$W=\begin{cases}\min(a\times W,W_{\max}); & \text{碰撞}\\ \max(W-b,W_{\min}); & \text{成功传输}\end{cases} \tag{3}$$

式中:a 与 b 是两个可变参数,在 MACAW 协议中分别取值为 1.5 和 1.可以看出,当传输成功时,MILD 算法并未将 W 立即减到最小,因此克服了 BEB 算法的缺点.但是,在活跃节点个数由多到少迅速减少的特定场景中,MILD 算法线性递减的特点导致其无法快速调整 W,造成无谓等待,降低了吞吐量.针对这种情况,文献[5]提出了积式增加和减少(MIMD,multiplicative increase and multiplicative decrease)算法,按照式(4)选取竞争窗口值[2]:

$$W=\begin{cases}\min(a\times W,W_{\max}); & \text{碰撞}\\ \max(W/a,W_{\min}); & \text{成功传输}\end{cases} \tag{4}$$

式中:a 的取值为 2.可见,由于竞争窗口呈倍数变化,多次冲突后 W 快速增大,多次成功传输后 W 快速减小,W 的调整幅度较大,相对于 BEB 算法来说,克服了成功节点长期占用信道的缺陷;相对于 MILD 算法来说,克服了节点较长时间的无谓等待.但是,MIMD 算法中 W 的变化方法均采用同一个公式,未考虑不同的网络负载情况,造成该算法不能同时适应各种网络负载.文献[6]提出一种自适应退避(AEFT,adaptive efficiency-fairness tradeoff)算法,网络中每个节点维持一个连续成功传输次数计数器和一个连续冲突次数计数器,通过比较计数器当前值与阈值来调整 W,阈值由负载决定,因而可以体现网络状态.但是,阈值为固定值,当网络负载不断变化时公平性表现不佳.文献[7]提出的算法也具有自适应特点,通过对 RTS(request to send)和 CTS(clear to send)的计数,来估计节点周围数据冲突情况,并相应修改退避值.该算法有助于降低冲突,提高吞吐量,但是公平性会受到影响.除上述方法外,还有研究者提出其他的退避算法,如带有动态分级缓冲的自适应二进制指数退避算法、扩展非重叠二进制指数退避算法以及增强型二进制指数退避算法等[8-10],然而,这些算法均未考虑到信道接入的公平性问题.

文献[2]提出一种新的自适应网络负载变化的退避算法 NAVB.利用竞争窗口 W 的值作为一个隐式计数器来反映网络负载程度,将 W 的取值范围划分为 3 个连续区间:$[W_{\min},h_1)$、$[h_1,h_2)$ 和 $[h_2,W_{\max}]$,h_1 和 h_2 为区间分段阈值,且 $h_1<h_2$.$[W_{\min},h_1)$ 内网络冲突率最低,当发生冲突时,只需线性增加 W;$[h_1,h_2)$ 内网络冲突率中等,当发生冲突时,以较小倍数增加 W;$[h_2,W_{\max}]$ 内网络冲突率最高,当发生冲突时,以较大倍数增加 W.即冲突发生时,W 按照式(5)所示调整[2]:

$$W=\begin{cases}W+c; & W_{\min}\leqslant W<h_1\\ W\times a; & h_1\leqslant W<h_2\\ \min(W\times b,W_{\max}); & h_2\leqslant W\leqslant W_{\max}\end{cases} \tag{5}$$

其中 a、b 和 c 为常数,并且 $a<b$.同样,在成功传输数据后,针对不同的负载范围而采用不同的方式修改 W.当 W 在 $[W_{\min},h_1)$ 内时,将 W 以较大的幅度递减;当 W 在 $[h_1,h_2)$ 内时,将 W 以一个较小的幅度递减;当 W 在 $[h_2,W_{\max}]$ 内时,W 线性递减.即传输成功时,W 按照式(6)所示调整[2]:

$$W=\begin{cases}\max(W/b,W_{\min}); & W_{\min}\leqslant W<h_1\\ \max(W/a,W_{\min}); & h_1\leqslant W<h_2\\ W-c; & h_2\leqslant W\leqslant W_{\max}\end{cases} \tag{6}$$

其中 a、b 和 c 为常数,并且 $a<b$.式(5)和(6)中,$W_{\min}$ 取值为 31,h_1 取值为 62,h_2 取值为 93,$W_{\max}$ 取值为 1 023.通过实验分析可知,当 $a=1.2$,$b=1.5$,$c=5$ 时,NAVB 的实验结果相对最优.NAVB 算法具体的退避时间按照式(2)计算.

NAVB 算法中,W 不会迅速变大或变小,而是根据网络负载的情况做出适当调整.该算法并不偏向任何节点,每个节点在选择退避时间时都是相对独立的.根据当前负载情况,所有节点退避时间都是在同一个范围内随机取得的,对于每个节点来说是相对公平的.但是,NAVB 算法也有一些不足之处.首先,对于吞吐量较低的节点,该

算法没有区别对待，没有为此类节点提供更多通信机会，以提高公平性；其次，竞争窗口调整时，未考虑本节点吞吐量与网络平均吞吐量之间的关系，不利于进一步提高公平性；再次，NAVB 算法的竞争窗口范围均为从 0 到 W，实际的随机退避时间有可能很小，在网络负载较重的场景下容易导致较大冲突.

1.2　公平性评价指数

为了衡量一种算法对节点信道接入公平性的影响，文献[11]提出了公平性指数(fairness index)概念，如式(7)所示：

$$F=P_{\max}/P_{\min} \tag{7}$$

其中 F 表示公平性指数，$P_{\max}$ 表示节点对最大的吞吐量，$P_{\min}$ 表示节点对最小的吞吐量. F 为 1 时信道资源分配完全公平，F 值越大，节点竞争信道能力越悬殊，公平性越差.

也有学者提出用改进的公平性指数(improved fairness index)衡量算法公平性[12]，如式(8)所示：

$$F'=(P_{\max}-P_{\min})/P_{\text{total}} \tag{8}$$

其中 F' 表示改进的公平性指数，P_{total} 表示网络中所有节点对吞吐量的总和. F' 为 0 时网络公平性最好，此时每个节点对具有相同的吞吐量，信道资源分配完全公平. 然而式(7)与(8)具有明显的缺陷，单从网络节点对最大吞吐量和节点对最小吞吐量来衡量信道竞争公平性不能完整反映网络整体的公平性情况，因而得到的结果也不够准确.

基于标准偏差概念，文献[13]提出了全路径公平性指数(fairness index of all links)的概念，能反映全网所有节点对竞争的公平性，如式(9)所示：

$$F_{\text{a}}=\frac{\sqrt{\sum_{i=1}^{n}(P_i-\overline{P})^2}}{\overline{P}} \tag{9}$$

其中 F_{a} 表示全路径公平性指数，P_i 为网络中第 i 个节点对的吞吐量，$\overline{P}$ 为网络中所有节点对吞吐量的平均值，n 为节点对数. F_{a} 越小，说明各节点的信道竞争公平性表现越好，当 F_{a} 等于 0 时，网络公平性达到最佳状态. 式(9)能够体现网络中每一个节点对的吞吐量情况，相对式(7)和(8)而言更加准确.

NAVB 算法采用式(7)作为公平性评价指数，不能反映每一个节点对的实际情况，具有较明显缺陷. 本文在 NAVB 算法基础上提出基于信道占用率的自适应公平算法(CORAFA 算法)，以有效克服 NAVB 算法的不利因素，既保证较高的网络吞吐量，又在一定程度上提高网络公平性.

2　基于信道占用率的自适应公平性算法(CORAFA 算法)

CORAFA 算法借鉴了 NAVB 算法随网络负载变化调整竞争窗口值的方法，但采用更合理的方式选取竞争窗口值，以改善吞吐量和公平性指标.

2.1　CORAFA 算法介绍

假设网络中节点对数目为 n，对于某节点对，定义从其中发送节点收到 ACK 确认包开始，到该节点下次收到 ACK 确认包为止，其间经历的时间为发包间隔 T. 定义发包间隔过程中，发送节点发送 RTS 包和 DATA 包，接收节点发送 CTS 包和 ACK 包所用时间之和为有效时间 V. 定义有效时间与发包间隔的比值为该发送节点在该发包间隔内的信道占用率，记作 S，则有

$$S=V/T \tag{10}$$

在绝对公平的情况下，每一个节点对的信道占用率应为 $1/n$，定义其为理想信道占用率. 当 $S=1/n$ 时，说明该节点对获得了公平的信道接入机会. 如果 $S>1/n$，则说明该节点对获得的信道占用率偏大，需适当增加竞争窗口值，以降低其成功接入信道的概率. 如果 $S<1/n$，则说明该节点对获得的信道占用率偏小，此时，记录发送节点的当前竞争窗口值为 W^*，然后将竞争窗口值置为 0，以使发送节点可以立即发送 RTS 包，增加节点对的有效时间 V，从而增大该节点对的信道占用率 S. 循环执行此过程，直至 $S\geqslant 1/n$，恢复竞争窗口值为 W^*. CORAFA 算法的目标是通过对 W 的调整，改变节点对接入信道的能力，最终实现所有节点对的信道占用率趋向 $1/n$ 发展的相对公平状态.

式(10)说明了当前发包间隔内节点对的信道占用率情况，但是不应仅仅以此为依据来调整竞争窗口值，因为此前发包间隔内节点对的信道占用率情况也对公平性有影响. 因此，应将式(10)改为递推公式，得到式(11)：

$$S_k=\begin{cases}1; & k=1\\ V/T; & k=2\\ S_{k-1}(1-\alpha)+(V/T)\alpha; & k>2\end{cases} \tag{11}$$

式中：S_k 为本次计算得到的信道占用率；S_{k-1} 为

前一个发包间隔内的信道占用率；α 为比例因子，其值由实验确定.

CORAFA 算法考虑了节点对的信道占用率与理想信道占用率 $1/n$ 间的关系，据此对 NAVB 算法中 W 的计算方法做出调整，使各节点对的实际吞吐量趋近于平均吞吐量.

当节点发包失败时，如果当前信道占用率 $S<1/n$，此时根据式(12)计算竞争窗口值. 通过 W 判断当前网络负载等级，当 W 在$[W_{\min},h_1)$时，以较小幅度线性增大 W；当 W 在$[h_1,h_2)$时，以较大幅度线性增大 W；当 W 在$[h_2,W_{\max}]$时，以较小倍数增加 W. 相对于 NAVB 算法的同一区间而言，W 的增量相对较小，因此增大了信道占用率小的节点对接入信道的机会. 如果当前信道占用率 $S\geqslant 1/n$，此时为了不显著降低网络总吞吐量，竞争窗口值的计算方法不变，仍采用式(5).

$$W=\begin{cases}W+c; & W_{\min}\leqslant W<h_1\\ W+d; & h_1\leqslant W<h_2\\ \min(W\times a,W_{\max}); & h_2\leqslant W\leqslant W_{\max}\end{cases}\tag{12}$$

其中 a、c 和 d 为常数，并且 $c<d$，在实验中，a、c 和 d 的取值分别为 1.2、5 和 10.

当节点发包成功时，如果当前信道占用率 $S<1/n$，说明该节点吞吐量较低，此时如前所述立即将 W 置为 0，使节点对能够以最大概率接入信道，提高信道占用率. 如果当前信道占用率 $S\geqslant 1/n$，此时根据式(13)计算竞争窗口值. 通过 W 判断当前网络负载等级，当 W 在$[W_{\min},h_1)$内时，以较小幅度减小 W；当 W 在$[h_1,h_2)$内时，以较大幅度线性减小 W；当 W 在$[h_2,W_{\max}]$内时，以较小幅度线性减小 W. 相对于 NAVB 算法的同一区间而言，W 的减小量相对较小，使该节点对成功接入信道的机会减小，其他节点接入信道的机会增大.

$$W=\begin{cases}\max(W/a,W_{\min}); & W_{\min}\leqslant W<h_1\\ W-d; & h_1\leqslant W<h_2\\ W-c; & h_2\leqslant W\leqslant W_{\max}\end{cases}\tag{13}$$

其中 a、c 和 d 为常数，并且 $c<d$，在实验中，a、c 和 d 的取值分别为 1.2、5 和 10.

与竞争窗口值分段计算的做法类似，CORAFA 算法的退避时间也分段计算，以适应各种网络负载情况，如式(14)所示. 当 W 在$[W_{\min},h_1)$时，网络负载较小，此时退避随机数在$(0,W)$选取，保证节点有较大吞吐量；当 W 在$[h_1,h_2)$时，网络负载中等，如在$(0,W)$内选取退避随机数，该数值有可能接近 0，此时较易引起冲突，因此在(h_1,W)内选取退避随机数；当 W 在$[h_2,W_{\max}]$内时，网络负载较重，基于同样理由，在(h_2,W)内选取退避随机数.

$$t=\begin{cases}Random(0,W)\times s; & W_{\min}\leqslant W<h_1\\ Random(h_1,W)\times s; & h_1\leqslant W<h_2\\ Random(h_2,W)\times s; & h_2\leqslant W\leqslant W_{\max}\end{cases}\tag{14}$$

CORAFA 算法中，公平性评价指数采用式(9)，相对于 NAVB 算法而言，更全面地考虑到每一个节点对的具体情况.

2.2 CORAFA 算法的参数选择

CORAFA 算法中，按照式(11)求某节点对的信道占用率时，需要确定 α 的值. 图 1 为 16 个节点组成的简单矩形拓扑场景，随机选择通信节点对为(3,4)、(1,10)、(9,0)、(7,13)、(6,15)、(2,11)、(5,8)和(12,14)，括号中前者为发送节点，后者为接收节点. 分别将 α 在 0 和 1 之间等间隔取值，按照式(9)定义的公平性评价指数，通过实验比较求得最佳 α 值为 0.8，如图 2 所示.



图 1 简单矩形拓扑场景

Fig. 1 Simple rectangular topology scene

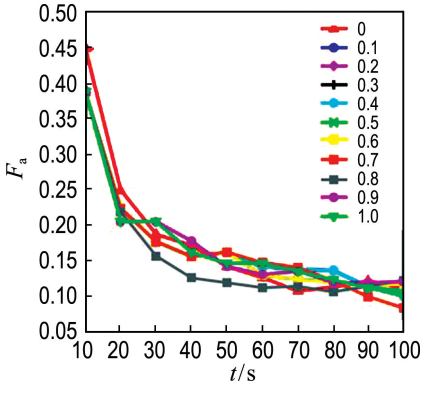


图 2 α 值对比图

Fig. 2 Comparison chart of α

3 算法实例分析

假设在单冲突域内有 8 个通信节点对，分别为节点对 A、B、C、D、E、F、G 和 H. 每个节点对均持续发送数据包，包大小为 1 024 B，发包速率为 1 Mb/s，s 设置为 20 μs，节点对随机选择发包前的退避窗口 τ，取值如式(15)所示：

$$\tau = Random(0, W) \tag{15}$$

在实际工作时，τ 按照如下方式选取. 不失一般性，将当前时间设为随机数种子，生成 8 个随机数，将这 8 个随机数再次作为种子，在[0,1]内分别生成 8 个随机数序列，每个随机数序列对应一个节点对. 用各序列中的每个随机数，乘以竞争窗口值 W，可以得到该节点对当前的退避窗口 τ. 分析中，NAVB 算法和 CORAFA 算法采用相同的随机数序列.

表 1 为 8 个节点对的部分退避窗口值，由于两种算法的对应节点对采用相同随机数序列，并且两种算法有相似之处，因此退避窗口的前 24 列取值相同. 初始时刻，用 8 个随机数序列的第 1 个数字乘以 W_{min}，得到各节点的初始退避窗口值. 此后，每经过一个退避时隙 s，各节点对的退避窗口值均减 1，直至减为 0，表示该节点对退避结束，可以发送 RTS 包. 之后，该节点对从自己的随机数序列中选取下一个数，乘以当前竞争窗口值 W，得到该节点对的下一个退避窗口值，如表中斜线后的数字所示. 如果有两个节点对的退避窗口值同时减为 0，此时发生冲突，两个节点对都不能发起通信，而是按照如上方法重新选取下一个退避窗口值，继续退避.

表 1　退避窗口值

Tab. 1　Value of backoff windows

节点对	第 1 列	第 2 列	第 3 列	第 4 列	第 5 列	第 6 列	第 23 列	第 24 列
A	6	3	0/18	11	8	3	5	3
B	3	0/30	27	20	17	12	8	6
C	13	10	7	0/27	24	19	2	0/16
D	16	13	10	3	0/27	22	13	11
E	26	23	20	13	10	5	7	5
F	30	27	24	17	14	9	0/6	4
G	21	18	15	8	5	0/19	18	16
H	21	18	15	8	5	0/19	33	31

初始时刻，$\tau_A \sim \tau_H$ 的初始退避窗口值分别为 6、3、13、16、26、30、21 和 21，如表 1 中的第 1 列所示. 节点对根据式(2)计算实际退避时间. 节点对 B 首先退避到 0，并且与其他节点对无冲突，因此在第 1 次信道竞争中，节点对 B 成功接入信道，如表 1 第 2 列所示.

对于 NAVB 算法来说，节点对 B 本次传输成功，因此根据式(6)计算新的竞争窗口值. 由于 W 在[W_{min}, h_1)内，所以新的 W 仍为 W_{min}. 节点对 B 的下一个退避窗口值由式(15)计算得到，其值为 30，如表 1 第 2 列所示.

对于 CORAFA 算法，由于节点对 B 本次传输成功，根据式(11)计算信道占用率 S 为 1. 因为 $S > 1/n$，采用式(13)计算新的竞争窗口值. 由于 W 在[W_{min}, h_1)内，所以新的 W 值仍为 W_{min}. 节点对 B 的下一个退避窗口值由式(15)计算得到，其值为 30，如表 1 中的第 2 列所示，与 NAVB 算法计算的结果相同.

图 3 为 NAVB 和 CORAFA 两种算法前 80 个时隙的时序图，坐标表示时间，坐标的刻度单位为一个退避时隙 s，每一个矩形代表一次成功传输，其中的字母代表成功传输的节点对名称. 传输过程中所有节点停止退避计数，因此在不考虑节点对成功传输的条件下，坐标轴刻度连续. 图 3 中，用向下箭头单独标注节点对的冲突时刻. 在坐标为 0 时，所有节点开始退避计数，节点对 B 经过 3 次退避后，成功传输数据. 根据表 1 可知，后续成功传输的节点对依次为 A、C 和 D，在第 21 个时隙处，节点对 G 和 H 同时退避到 0，发生冲突，如表 1 第 6 列所示.

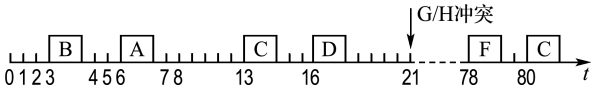


图 3　算法时序图

Fig. 3　Timing sequence chart of algorithm

对于 NAVB 算法，由于节点对 G 和 H 发生冲突，因此均需要根据式(5)计算新的竞争窗口值. 由于 W 在[W_{min}, h_1)内，所以新的 W 线性增加为 36. 节点对 G 的下一个退避窗口值由式(15)计算得到，其值为 19；节点对 H 的下一个退避窗口值也由式(15)计算得到，其值恰好也为 19.

对于 CORAFA 算法，发送冲突时，先根据式(11)计算节点对 G 和 H 的信道占用率 S 均为 1，因为 $S > 1/n$，选择式(12)计算新的竞争窗口值. 由于 W 在[W_{min}, h_1)内，新的 W 应在原基础上加 5，其值为 36. 节点对 G 的下一个退避窗口值由式(15)计算得到，其值为 19；节点对 H 的下一个退避窗口值也为 19，与 NAVB 算法计算的结果相同.

由于前几次竞争中，节点对的信道占用率均大于 $1/n$，且 W 在[W_{min}, h_1)内，因此在 NAVB 与 CORAFA 算法中，节点接入信道顺序一致，直到

表 1 中的第 24 列为止.

图 4 表示的是 NAVB 算法中节点对 C 的接入时序图. 节点对 C 在第 13 个时隙成功传输,在第 40 个时隙发生冲突,之后在第 80 个时隙成功传输. 根据表 1 的第 24 列可知,节点对 A 剩余的退避窗口值最小,其值为 3,因此第 83 个时隙时,节点对 A 接入信道.

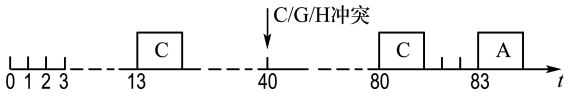


图 4 NAVB 算法中节点对 C 接入时序图

Fig. 4 Access timing sequence chart of nodes C in NAVB algorithm

图 5 是 CORAFA 算法中节点对 C 的接入时序图. 在节点对 C 的第 1 个发包间隔 T_1 内,共有 7 个节点对成功发包,因此在第 40 个时隙时,根据式(11),可求得节点对 C 的信道占用率 $S_2=1/8$. 在节点对 C 的第 2 个发包间隔 T_2 内,共有 8 个节点对成功发包,根据式(11),可求得节点对 C 的信道占用率 $S_3=(1/8)\times 0.2+(1/9)\times 0.8=0.114$. 由于 $S_3<1/n$,说明节点对 C 在这个过程中接入信道的机会偏低. 为了提高 C 的信道占用率,立即将节点对 C 的竞争窗口 W 置为 0,让节点对 C 连续传输数据,直至信道占用率大于 $1/n$,以此提高公平性.

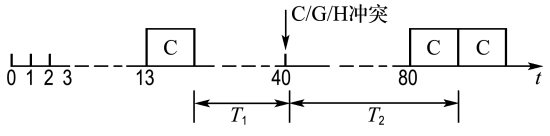


图 5 CORAFA 算法中节点对 C 接入时序图

Fig. 5 Access timing sequence chart of nodes C in CORAFA algorithm

通过仿真,模拟 1 s 内上述信道竞争过程,令节点对 i 每次竞争信道时的信道占用率为 S_i,此时所有节点对的平均信道占用率为 $\overline{S}$,根据式(16),计算所有节点对信道占用率的标准差 σ.

$$\sigma=\sqrt{\frac{\sum_{i=1}^{n}(S_i-\overline{S})^2}{n}} \tag{16}$$

图 6 体现了信道占用率标准差随时间变化情况,由图可知,CORAFA 算法中,各节点对的信道占用率较为接近,明显优于 NAVB 算法,说明 CORAFA 算法的公平性明显好于 NAVB 算法.

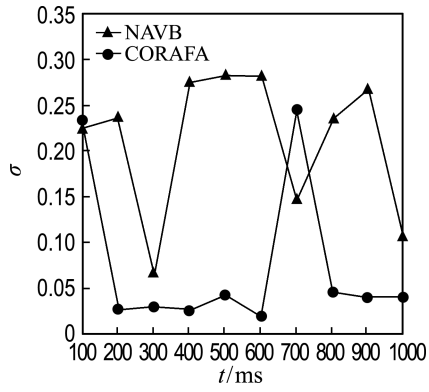


图 6 信道占用率标准差 σ 随时间变化

Fig. 6 Changing of standard deviation σ of channel occupancy rate following time

将仿真时间延长到 1 000 s,CORAFA 算法的信道占用率标准差 σ 可收敛到 0.08～0.22,可知该算法比较稳定.

4 协议仿真

利用 NS2 仿真平台,将 CORAFA 算法与 BEB 算法、MILD 算法、MIMD 算法及 NAVB 算法的吞吐量和公平性进行比较,其中公平性评价指数采用式(9)计算. 实验共分为 3 组,分别比较各算法在节点对数为 4、8 和 12 时的表现.

4.1 仿真场景

仿真场景中,节点间仍采用简单矩形拓扑结构,节点位置随机安排;CORAFA、BEB、MILD、MIMD 和 NAVB 等几种算法采用相同实验参数,如表 2 所示.

表 2 仿真参数

Tab. 2 Simulation parameters

参数	数值
区域范围	500 m×500 m
信号传输半径	250 m
DIFS 时长	50 μs
SIFS 时长	10 μs
信号传播时间	5 μs
传输带宽	2 Mb/s
数据包大小	1 024 B
发包速率	1 Mb/s
仿真时间	100 s
传输层协议	UDP
流量产生器	CBR

4.2 仿真结果及分析

第 1 组实验选取 4 对节点,比较几种算法吞

吐量和公平性随时间变化的情况，分别如图 7 和 8 所示. 图 7 中，由于 BEB 算法仅仅为了提高网络吞吐量性能，不考虑公平性，因此 BEB 算法的吞吐量较高. MIMD、NAVB 和 CORAFA 算法也能保证较高吞吐量，这是因为当前网络负载较低，冲突较少. MILD 吞吐量指标明显偏低，这是由于其倍数增加线性减少 W，导致其 W 值长期偏大，在低负载情况下无谓等待时间很长，因此吞吐量较低.

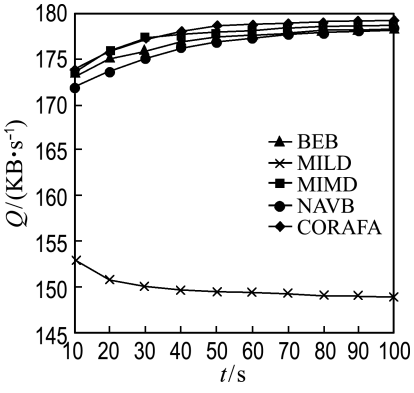


图 7　4 对节点时吞吐量随时间变化情况

Fig. 7　Throughput changing following time with 4 pairs of nodes

图 8 中，由于 MILD 算法倍数增加线性减少 W，导致节点长期无法接入信道，严重影响公平性. BEB 算法每次成功传输后，W 均取为 W_{min}，这在低负载情况下对其他节点通信影响不大，因此 BEB 算法表现优于 MILD 算法. NAVB 算法能够根据负载情况动态调整 W，公平性优于 BEB 算法. MIMD 算法采用倍数增加倍数减少 W，对 W 调整较为迅速，与 CORAFA 算法一样，表现最好.

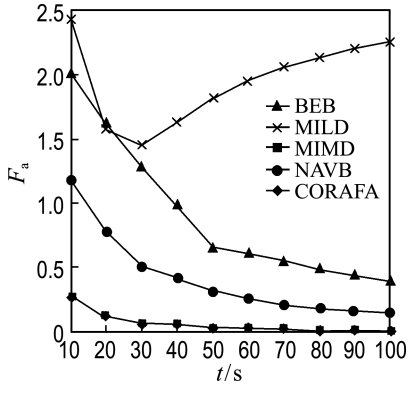


图 8　4 对节点时公平性随时间变化情况

Fig. 8　Fairness changing following time with 4 pairs of nodes

第 2 组实验选取 8 对节点，比较几种算法吞吐量和公平性随时间变化情况，分别如图 9 和 10 所示. 图 9 中，由于网络负载增加，冲突增大，几种算法的吞吐量普遍稍有下降. 基于同样原因，BEB 算法在吞吐量方面表现仍旧较好，CORAFA 算法表现与其相当，在考虑公平性的同时，吞吐量损失不大. MIMD 和 NAVB 算法的吞吐量受负载影响较大，表现有所降低. 但是，MILD 算法线性减少 W 的方法能够降低冲突，因此相对低负载情况而言，吞吐量缓慢上升.

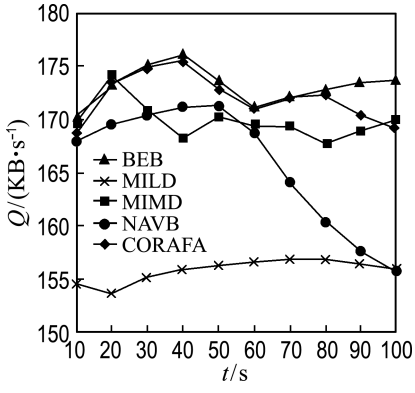


图 9　8 对节点时吞吐量随时间变化情况

Fig. 9　Throughput changing following time with 8 pairs of nodes

图 10 为几种算法的公平性表现. 与 4 对节点的低负载环境相比，MIMD 算法不能根据当前负载情况随时调整 W，因此表现较差. MILD 算法随着冲突增加，节点接入信道的平均时间变长，表现更差. 其他几种算法与低负载情况相差不大，但是 CORAFA 算法的公平性指数值最低，即表现最好.

第 3 组实验选取 12 对节点，比较几种算法吞吐量和公平性随时间变化的情况，分别如图 11 和 12 所示. 图 11 中，由于 CORAFA 算法考虑了节点吞吐量和平均吞吐量之间的关系，能够动态调整竞争窗口值，因此在高网络负载条件下表现仍然最好. BEB 算法仅考虑吞吐量，不考虑公平性，因此吞吐量仍旧较大. MIMD 算法在成功传输之后倍数减少 W，造成较大冲突，降低了吞吐量. MILD 算法线性减少 W，当负载高时，受冲突影响较小. NAVB 算法注重公平性，吞吐量对不同负载的适应性较差，因此吞吐量有明显下降.

图 12 为几种算法在高负载时的公平性表现. BEB 算法不考虑公平性，表现最差. MILD 算法的

竞争窗口取值变化幅度很小,各节点间相差不大,因此公平性最好,但是这是以牺牲吞吐量为代价的. MIMD、NAVB 和 CORAFA 算法的表现介于前述二者之间,但是 CORAFA 算法吞吐量指标优于其他两种算法,综合表现最好.

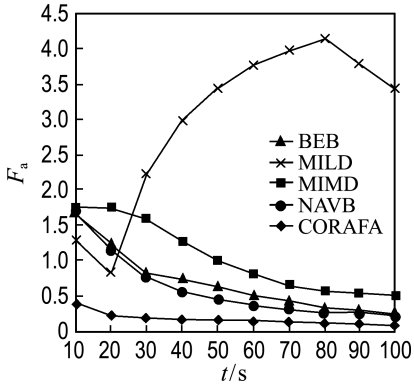


图 10 8 对节点时公平性随时间变化情况

Fig. 10 Fairness changing following time with 8 pairs of nodes

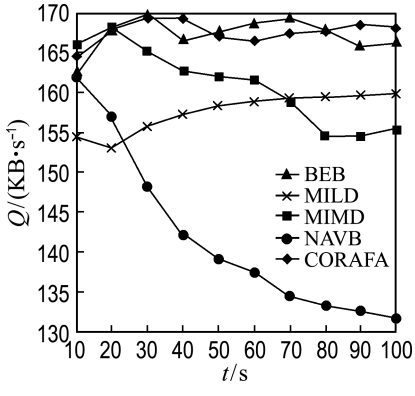


图 11 12 对节点时吞吐量随时间变化情况

Fig. 11 Throughput changing following time with 12 pairs of nodes

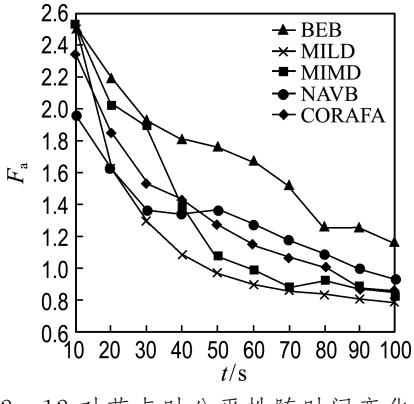


图 12 12 对节点时公平性随时间变化情况

Fig. 12 Fairness changing following time with 12 pairs of nodes

综上所述,在吞吐量方面,CORAFA 算法与 BEB 算法相似,远优于其他 3 种公平性算法;在公平性方面,CORAFA 算法远优于 BEB 算法,并且在低负载场景下表现最佳,在高负载下与 NAVB 和 MIMD 算法表现类似. 综合考虑吞吐量和公平性,CORAFA 算法较其他几种算法表现更好.

5 结 语

本文提出了 CORAFA 算法,节点根据当前网络负载与信道占用率情况,动态调整竞争窗口值,并合理确定随机退避时间选择范围,减少了通信冲突. 仿真结果表明,相对于 BEB、MIMD、MILD 和 NAVB 等算法,本算法在保证吞吐量的同时提高了网络公平性,具有一定优势. 今后可研究根据网络负载动态调整区间分段阈值 h_1 和 h_2 的方法,以更灵活地适应负载变化情况,进一步减少冲突,改善吞吐量和公平性.

参考文献:

[1] LAN/MAN Standards Committee of IEEE Computer Society. IEEE Std 802.11-2007 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications [S]. Piscataway: IEEE, 2007.

[2] Gafur M A, Upadhayaya N, Sattar S A. Enhancing the performance of Ad Hoc networks using new adaptively varying backoff technique [J]. **International Journal of Engineering Science and Technology**, 2010, **2**(12):7078-7086.

[3] 吴传霞,范平志,冯军焕. 一种 Ad Hoc 网络信道接入排队退避公平算法[J]. 系统仿真学报, 2004, **16**(5):1111-1114.
WU Chuan-xia, FAN Ping-zhi, FENG Jun-huan. On a queue backoff fair algorithm for channel access in Ad Hoc network [J]. **Journal of System Simulation**, 2004, **16**(5):1111-1114. (in Chinese)

[4] Bharghavan V, Demers A, Shenker S, *et al*. MACAW: A media access protocol for wireless LANs [J]. **Computer Communications Review**, 1994, **24**(4):212.

[5] WU Hai-tao, CHENG Shi-duan, PENG Yong, *et al*. IEEE 802.11 distributed coordination function(DCF): analysis and enhancement [C] // **2002 IEEE International Conference on Communications**. Piscataway: IEEE, 2002:605-609.

[6] CUI Hai-xia, WEI Gang. A novel backoff algorithm based on the tradeoff of efficiency and fairness for Ad Hoc networks [C] // **Proceedings-2009 WRI International Conference on Communications and Mobile Computing, CMC 2009**. Piscataway: IEEE Computer Society, 2009:81-86.

[7] 赵志峰,郑少仁. Ad Hoc 网络信道接入技术研究[J]. 解放军理工大学学报(自然科学版), 2001, **2**(3):47-51.
ZHAO Zhi-feng, ZHENG Shao-ren. Research about Ad Hoc network channel access technology [J]. **Journal of PLA University of Science and Technology (Natural Science Edition)**, 2001, **2**(3): 47-51. (in Chinese)

[8] Chekka R T, Miao T, Kim K H. Implementation of adaptive binary exponential backoff (ABEB) algorithm with dynamical sizing buffer for load-balanced RPL [C] // **2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)**. Piscataway: IEEE, 2014:562-564.

[9] Lee S Y, Shin Y S, Lee K W, *et al*. Performance analysis of extended non-overlapping binary exponential backoff algorithm over IEEE 802.15.4 [J]. **Telecommunication Systems**, 2014, **55**(1):39-46.

[10] Chin H H, Lin C C, Deng D J. E-BEB: Enhanced binary exponential backoff algorithm for multi-hop wireless ad-hoc networks [J]. **Wireless Personal Communications**, 2014, **76**(2):193-207.

[11] Ozugur T, Naghshineh M, Kermani P, *et al*. Fair media access for wireless LANs [C] // **Conference Record/IEEE Global Telecommunications Conference**. Piscataway: IEEE, 1999:570-579.

[12] 李瑞芳,李仁发. Ad Hoc 网络信道接入退避算法研究[J]. 科学技术与工程, 2006, **6**(15):2358-2363.
LI Rui-fang, LI Ren-fa. The backoff algorithm for channel access in Ad Hoc networks [J]. **Science Technology and Engineering**, 2006, **6**(15): 2358-2363. (in Chinese)

[13] 李　广. Ad Hoc 网络多信道 MAC 协议研究与实现[D]. 武汉:武汉理工大学, 2009.
LI Guang. Research and realization of the multi-channel MAC protocol in Ad Hoc network [D]. Wuhan: Wuhan University of Technology, 2009. (in Chinese)

Channel occupancy rate-based adaptive fairness algorithm for Ad Hoc network

LAI Xiao-chen[1,2], LIU Quan-li[*1], REN Zhi-lei[2], ZHAO Ying[2]

(1. Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China;
2. Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software Technology, Dalian University of Technology, Dalian 116620, China)

Abstract: In Ad Hoc network, nodes compete channels to communicate. It is likely to cause fairness problems if the selection strategy of collision window value is unreasonable. With the analyses of the implementation mechanisms of existing typical fairness algorithms, a concept of channel occupancy rate is raised. Based on this concept, an adaptive fairness algorithm for Ad Hoc network is proposed. The network correspondence is classified into four types according to the history of correspondence and the relationship between the current channel occupancy rate and the ideal channel occupancy rate. In each type, the collision window value of node is set dynamically based on the current network load. Simulation results show that the proposed algorithm performs well under various load conditions. It can significantly improve fairness of channel access among nodes as well as throughput, so it is better than BEB, MILD, MIMD and NAVB algorithms.

Key words: Ad Hoc network; fairness; collision window; throughput