

## 基于冲突分类模型的冲突解析算法\*

张棋飞<sup>1+</sup>, 刘威<sup>2</sup>, 孙宝林<sup>1</sup>, 桂超<sup>1</sup>, 严冰<sup>1</sup>

<sup>1</sup>(湖北经济学院 计算机学院,湖北 武汉 430205)

<sup>2</sup>(华中科技大学 电子与信息工程系,湖北 武汉 430074)

### Collision Classification Model-Based Collision Resolution Algorithm

ZHANG Qi-Fei<sup>1+</sup>, LIU Wei<sup>2</sup>, SUN Bao-Lin<sup>1</sup>, GUI Chao<sup>1</sup>, YAN Bing<sup>1</sup>

<sup>1</sup>(School of Computing, Hubei University of Economics, Wuhan 430205, China)

<sup>2</sup>(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

+ Corresponding author: E-mail: cheffly@gmail.com

**Zhang QF, Liu W, Sun BL, Gui C, Yan B. Collision classification model-based collision resolution algorithm.**

*Journal of Software*, 2010,21(3):548-563. <http://www.jos.org.cn/1000-9825/3452.htm>

**Abstract:** Traditional backoff algorithms in IEEE 802.11 networks adopt contention window scheme for collision resolution. Collided nodes are redistributed in extended window ranges to avoid further collisions. However, as long as these distribution windows intersect with each other, collision may still occur. To solve this problem, this paper puts forward a collision classification model to classify collisions into cross collision and intra collision and proposes to solve them with different policies. It utilizes sequential discrete window distribution (SDWD) scheme to resolve cross collision by distributing the collided nodes in a series of discrete Elementary Windows and the intra collision is resolved with an appropriate elementary window size to achieve a tradeoff between intra collision probability and packet latency. Based on this proposal, two algorithms are developed featuring cross collision resolution (CCR) and collision-free CCR (CF-CCR). The extensive simulations demonstrate that compared with IEEE 802.11 DCF protocol, the CCR and CF-CCR algorithms consistently excel, in terms of collision rate, throughput, delay, fairness and delay jitter. Moreover, CCR and CF-CCR exhibit their respective advantages in different scenarios.

**Key words:** IEEE 802.11; DCF; backoff; collision classification model; sequential discrete window distribution

**摘要:** IEEE 802.11 网络中的传统退避算法利用竞争窗口机制进行冲突解析,冲突节点被重新分布在一个更大的窗口范围内以避免冲突,然而,只要这些分布窗口之间存在着交集,就仍有可能引发冲突。为了解决一个问题,提出了一种冲突分类模型,将网络中的分组冲突分成交叉冲突和同级冲突,并且提出针对这两种不同类型的冲突应该采

\* Supported by the National Natural Science Foundation of China under Grant Nos.60602029, 60772088 (国家自然科学基金); the Young and Middle-Aged Elitists' Scientific and Technological Innovation Team Project of the Institutions of Higher Education in Hubei Province of China under Grant No.T200902 (湖北省高等学校优秀中青年科技创新团队资助计划项目); the Scientific Research Projects of Hubei Education Department of China under Grant Nos.D20081904, Q20091903, B20091904 (湖北省教育厅科研项目); the Young Scientific Research Project of Hubei University of Economics of China under Grant No.XJ2009011 (湖北经济学院校级青年项目)

Received 2008-04-16; Accepted 2008-08-07

取不同的策略进行解析.利用顺序离散窗口分布机制(sequential discrete window distribution,简称 SDWD)解析交叉冲突,通过节点分布窗口的离散化避免交叉冲突;同时,通过设置合适的分布窗口大小在同级冲突概率和分组延迟之间取得折衷.在此基础上,实现了两种冲突解析算法:冲突分类解析算法(cross collision resolution,简称 CR)和无冲突分类解析算法(collision-free CCR,简称 CF-CCR).仿真结果表明,同 IEEE 802.11 DCF 协议相比,CCR 和 CF-CCR 在冲突率、吞吐量、分组延迟、公平性以及延迟抖动方面均有提高.并且,这两个算法在不同场合表现出不同的优越性.

关键词: IEEE 802.11;分布协调函数;退避;冲突分类模型;顺序离散窗口分布

中图法分类号: TP393 文献标识码: A

IEEE 802.11 无线网络中的媒体接入控制(media access control,简称 MAC)协议负责解析节点间的竞争冲突,协调节点的信道接入,直接关系到信道的利用效率和网络性能<sup>[1]</sup>.具体来讲,MAC 协议负责解析由于竞争节点同时接入信道所导致的分组冲突问题.为了避免冲突,MAC 通常采用随机退避(random backoff)机制,使节点在重发之前随机等待一段时间,进入某个退避阶段(backoff stage),根据既定的退避规则调度分组的下一次发送.

图 1 是目前广泛应用的 IEEE 802.11 DCF(distributed coordination function)<sup>[2]</sup>协议基本接入模式示意图.节点在开始传输之前首先感知信道状态,只有信道连续空闲 DIFS(DCF interframe space)时间后才能发送数据.倘若信道忙,则节点在当前竞争窗口(contention window,简称 CW)范围 $[0, CW-1]$ 内随机选取一个整数,称为退避计数器(backoff counter,简称 BC),并以时隙为单位开始进行退避.每空闲一个时隙,退避计数器值减 1,直到变为 0 后节点发送数据.在退避期间,倘若信道由闲变忙,则退避进程暂时挂起,直到信道连续空闲 DIFS 后继续递减计数器值.DCF 采用二进制指数退避算法 BEB(binary exponential backoff)<sup>[3]</sup>.节点的竞争窗口初始值为  $CW_{min}$ .在每次发送失败后,将窗口加倍,直至最大值  $CW_{max}$ .一旦发送成功,竞争窗口立刻重设为  $CW_{min}$ .

$$\begin{cases} CW_i = \min(2^i CW_{min}, CW_{max}), & \text{upon collision} \\ CW_0 = CW_{min}, & \text{upon success} \end{cases}$$

其中,  $i$  代表节点的退避阶段,反映了节点的退避状态,取决于节点自最近一次成功发送以来所遭遇的冲突次数<sup>[4]</sup>.倘若收到发送方发来的数据分组,接收节点经过 SIFS(short interframe space)时间后回送一个确认信息 ACK(acknowledgement).如果发送节点没有及时收到 ACK,表明传输失败,于是重新选择计数器值进行新一轮退避.

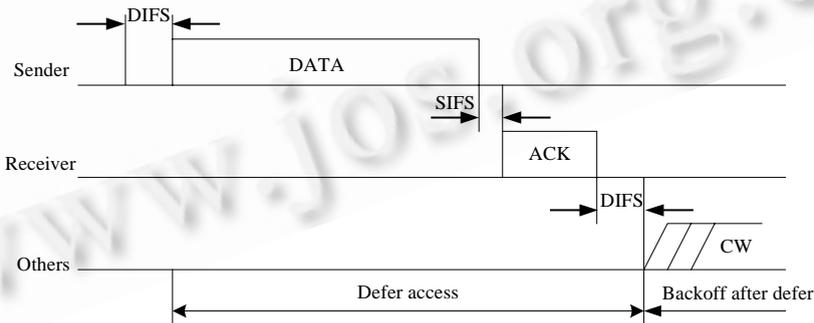


Fig.1 Basic access mode of IEEE 802.11 DCF

图 1 IEEE 802.11 DCF 协议中的基本接入模式

传统的退避算法只是利用退避过程的随机性来规避冲突,通过在不同大小的窗口范围内随机选取不同的退避计数器值来实现冲突的避免.据我们所知,现有退避算法并没有对网络中冲突的性质进行深入的分析 and 区分,而是对所有的冲突采取同样的方式进行处理.我们认为,网络中存在着两种不同性质的分组冲突,应该采取不同的方式进行解析.为此,我们提出了一种冲突分类模型,并基于该模型提出了两种冲突解析算法.下节首先会介绍几种典型的 MAC 层退避算法,接着通过分析传统退避算法存在的问题,提出我们的冲突分类模型.随后,

将依据该模型分别设计针对不同冲突类型的解析机制,并提出两种冲突解析算法:冲突分类解析算法 CCR(collision classification resolution)和无冲突分类解析算法 CF-CCR(collision-free CCR).仿真结果在第5节中给出.最后对全文进行总结.

## 1 相关工作

通常来讲,退避算法在节点发送成功后减小退避窗口,在遭遇冲突时增大退避窗口,以动态反映网络的竞争状况,关键就在于退避窗口变化的速度以及其他节点如何进行响应<sup>[5]</sup>.

BEB 在分组发送成功后直接重设竞争窗口为  $CW_{min}$ ,导致发送成功的节点有可能再次接入信道,影响了算法的公平性.因此,MILD(multiplicative increase and linear decrease)<sup>[6]</sup>在每次分组发送失败后都将竞争窗口增大 1.5 倍,并且在每次成功传输后将窗口值线性减小 1,避免了退避间隔大小变化过快,提高了公平性.但由于采用了退避计数器拷贝机制,MILD 会导致不同业务负载区域之间的退避窗口迁移问题<sup>[5]</sup>.

SD\_DCC<sup>[7]</sup>是一种基于慢退避思想的分布式接入控制退避算法,通过准确记录数据帧成功发送后高负荷网络的退避阶数来记录网络当前状态,降低冲突概率.

在 NSAD(new self-adaptive DCF)<sup>[8]</sup>中,节点成功发送一个报文后,并不直接重设  $CW$  为  $CW_{min}$ ,而是将其更新为初始竞争窗口  $CW_{init}$ .每个节点根据当前观察到的网络碰撞时长和空闲时长独立计算当前负载的大小,并通过与最优负载  $l_{opt}$  的比较结果来进一步调整  $CW_{init}$ .NSAD 通过在数据帧中携带优化后的窗口值来保证公平性,同样会带来窗口迁移问题.

竞争窗口慢速减小算法 SD(slow CW decrease)<sup>[9]</sup>在成功发送一个报文后,将节点的竞争窗口减小为原来的  $\delta$ 倍( $\delta=0.9$ ).而在 MIMD(multiplicative increase multiplicative decrease)<sup>[10]</sup>中,窗口减小速度更快,每次发送成功后都将  $CW$  减半.EIED(exponential increase exponential decrease)则采用了一种更为通用的算法,当报文发送成功后,用  $CW/r_D$  更新  $CW$ ;报文发送失败则用  $CW/r_I$  更新  $CW$ .

MIMLD(multiplicative increase multiplicative/linear decrease)<sup>[11]</sup>额外使用了一个参数  $CW_{basic}$  作为区分网络竞争程度的标准.当  $CW > CW_{basic}$  时,认为竞争较激烈,报文成功发送后将窗口减半;否则,线性递减  $CW$  值.

LMILD(linear/multiplicative increase linear decrease)<sup>[12]</sup>将发送失败节点的  $CW$  值增大为原来的  $m_c$  倍,同时,监听到冲突的其他节点将  $CW$  增大  $l_c$ .成功发送报文后,所有节点的  $CW$  值都线性减小.

GDCF(gentle DCF)<sup>[13]</sup>算法规定:只有当节点连续成功发送若干个报文后才将竞争窗口值减小一半.GDCF 可以支持基于优先级或其他一些有服务质量保证需求的业务.

## 2 冲突分类模型

### 2.1 动机

尽管采取了各种策略对竞争窗口的大小做出调整,传统退避算法仍然无法有效避免分组冲突的发生.当前的冲突解析机制只是利用退避过程的随机性来规避冲突,通过在不同大小的窗口范围内随机选取不同的退避计数器值来实现冲突的解析.退避后的节点有可能选择一个已经被其他节点提前占用的时隙进行传输,导致冲突的再次发生,不仅消耗了系统资源、浪费了带宽,而且增加了分组延迟,给系统性能带来很大影响.

下面我们以典型的 IEEE 802.11 协议为例,说明传统退避算法存在的问题.如图 2 所示,假设有 3 个节点  $a, b$  和  $c$  均处于退避阶段  $i$ ,其竞争窗口大小为  $[0, CW_i - 1]$ .3 个节点分别随机选取自己的退避计数器值为  $j, j$  和  $k$ .由于节点  $a$  和  $b$  选取了同一时隙进行发送,所以在  $j$  时刻,必然发生分组冲突.冲突节点  $a$  和  $b$  于是进入退避阶段  $(i+1)$ ,增大自己的竞争窗口为  $[0, 2CW_i - 1]$ ,并在新窗口上重新选择退避计数器值,确定自己下一次分组发送的时隙分别为  $l$  和  $k$ (分别用  $a'$  和  $b'$  表示).碰巧此时节点  $b'$  选取的发送时隙  $k$  已经被处于退避阶段  $i$  的  $c$  节点提前占用,于是在  $k$  时刻,处于退避阶段  $i$  的  $c$  节点和处于退避阶段  $(i+1)$  的  $b'$  节点同时发送分组,再次造成冲突.不难看出,这次冲突是发生在两个处于不同退避阶段节点之间的.也就是说,即便节点已经进行过退避(如  $b'$  节点),但是仍

然不能保证其分组的成功发送。

由于采用了基于随机性的退避规则,传统算法并不能取得理想的冲突解析效果,甚至还会对其他节点的正常发送造成干扰,带来新的分组冲突.例如在本例中,本来节点  $c$  是可以成功发送的,可是由于节点  $b$  退避后恰巧选取了  $c$  节点的发送时隙作为它的发送时刻,导致本来应该成功发送的分组发生了冲突,影响了系统性能。

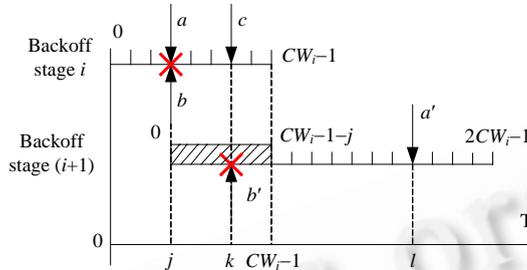


Fig.2 Illustration of collision resolution in IEEE 802.11 networks of traditional backoff algorithms

图 2 IEEE 802.11 网络传统退避算法冲突解析示意图

通过观察不难看出,在以上退避过程中,节点  $b$  所经历的两冲突性质是不一样的.第 1 次冲突是发生在两个处于同一退避阶段的节点之间(同处退避阶段  $i$  的节点  $a$  和节点  $b$ );而第 2 次冲突则是发生在两个处于不同退避阶段的节点之间(处于退避阶段  $i$  的节点  $c$  和处于退避阶段  $(i+1)$  的节点  $b'$ ).根据冲突节点所处的退避阶段的不同,我们将网络中的分组冲突划分成两类:交叉冲突(cross collision)和同级冲突(intra collision).所谓交叉冲突,是指产生冲突的节点处于不同的退避阶段;而在同级冲突中,冲突节点都处于同一退避阶段.如图 2 中,发生在节点  $a$  和节点  $b$  之间的冲突称为同级冲突,而节点  $b'$  和节点  $c$  之间的冲突则称为交叉冲突。

通过进一步的分析我们发现,之所以会产生这种冲突解析效果得不到保障的情况,其根本原因在于退避节点选取的退避空间与其他节点的退避空间存在某种程度的重叠,导致二者有可能选取同一时隙作为发送时刻,从而引入新的分组冲突.那么到底有没有一种策略能够有效避免冲突,使得节点重新选择的发送时隙与其他节点互不干扰呢?如果存在这样一种方法,那么对系统性能的提升无疑是有帮助的.下面我们就来针对该问题进行具体分析。

### 2.2 交叉冲突概率分析

假定在 IEEE 802.11 网络中,节点处于退避阶段  $i$ ,其竞争窗口大小为  $CW_i(0 \leq i < i_{\max}, CW_{\min} \leq CW_i < CW_{\max})$ .倘若在第  $j(0 \leq j \leq CW_i - 1)$  个时隙发生冲突,则节点加倍自己的退避窗口,进入退避阶段  $(i+1)$ .如图 2 所示,退避阶段  $(i+1)$  的竞争窗口大小为  $[0, 2CW_i - 1]$ .图中的阴影部分表示两个退避窗口的交集.我们定义重叠率(superposition rate,简称 SR)为竞争窗口重叠的部分占其组成部分的比例,即

$$SR_i = \frac{\text{sizeof}(\text{Intersection of Contention Windows})}{\text{sizeof}(\text{Contention Window for Backoff Stage } i)} \tag{1}$$

其中,函数  $\text{sizeof}(\cdot)$  表示窗口大小.于是,可以得到两个退避阶段的重叠率分别为

$$SR_i = \frac{CW_i - 1 - j + 1}{CW_i} = 1 - \frac{j}{CW_i} \tag{2}$$

$$SR_{i+1} = \frac{CW_i - 1 - j + 1}{2CW_i} = \frac{1}{2} - \frac{j}{2CW_i} \tag{3}$$

同时,竞争窗口大小  $CW_i$  满足:

$$CW_i = 2^i \times CW_{\min} \tag{4}$$

因此有

$$SR_i = 1 - \frac{j}{2^i \times CW_{\min}} \quad (5)$$

$$SR_{i+1} = \frac{1}{2} - \frac{j}{2^{i+1} \times CW_{\min}} \quad (6)$$

下面利用重叠率来评估网络中交叉冲突概率的大小.为了便于分析,首先我们只考虑两个连续的退避阶段: $i$ 和 $(i+1)$ .在这两个连续的退避阶段之间发生交叉冲突的必要条件满足:

- 处于退避阶段 $(i+1)$ 的节点选取的退避计数器值位于重叠区域内,即 $BC_{i+1} \in [0, CW_i - 1 - j]$ ,其中, $BC$ 表示退避计数器值;
- 在退避阶段 $i$ ,仍然有节点的退避计数器值位于交集之内,即 $BC_i \in [j, CW_i - 1]$ .

因此,在退避阶段 $i$ 和 $(i+1)$ 之间发生交叉冲突的概率 $Pr_{cr(i,i+1)}(j)$ 满足:

$$Pr_{cr(i,i+1)}(j) \leq Pr\{BC_i \in [j, CW_i - 1]\} \times Pr\{BC_{i+1} \in [0, CW_i - 1 - j]\} \quad (7)$$

其中, $Pr\{BC_i \in [j, CW_i - 1]\}$ 表示选取的退避计数器值位于给定区域内的概率.由于退避计数器值在竞争窗口内均匀分布,于是有

$$Pr\{BC_i \in [j, CW_i - 1]\} = SR_i \quad (8)$$

$$Pr\{BC_{i+1} \in [0, CW_i - 1 - j]\} = SR_{i+1} \quad (9)$$

联立方程(5)~方程(9),可以得到:

$$Pr_{cr(i,i+1)}(j) \leq \frac{(CW_i - j)^2}{2CW_i^2} = \frac{(2^i CW_{\min} - j)^2}{2^{2i+1} CW_{\min}^2} \quad (10)$$

显然,最大交叉冲突概率函数 $\max Pr_{cr(i,i+1)}(j)$ 满足:

$$\max Pr_{cr(i,i+1)}(j) = \frac{(2^i CW_{\min} - j)^2}{2^{2i+1} CW_{\min}^2} \quad (11)$$

这是一个连续单调递减函数,在取值范围 $j \in [0, 2^i CW_{\min} - 1]$ 内的最大值为

$$\max Pr_{cr(i,i+1)}(0) = 1/2 \quad (12)$$

最小值为

$$\max Pr_{cr(i,i+1)}(2^i CW_{\min} - 1) = \frac{1}{2^{2i+1} CW_{\min}^2} \quad (13)$$

显然,冲突发生得越早,引发交叉冲突的概率越大.尤其是当退避阶段 $i$ 达到其上限值 $i_{\max}$ 时,有

$$CW_i = CW_{i+1} = CW_{\max} \quad (14)$$

此时,最大交叉冲突概率函数为

$$\max Pr_{cr(i_{\max}, i_{\max}+1)}(j) = \frac{(2^i CW_{\min} - j)^2}{(2^i CW_{\min})^2} \quad (15)$$

由此可见,当满足公式(16)时,交叉冲突概率甚至可以无穷逼近1.

$$2^i CW_{\min} \gg j \quad (16)$$

虽然以上我们推导了两个连续退避阶段之间的最大交叉冲突概率,但是应该强调的是:交叉冲突不仅仅发生在两个连续的退避阶段之间.只要节点的退避窗口存在重叠,就有可能引发交叉冲突.因此,对于一个当前处于退避阶段 $s$ 的节点来说,其交叉冲突概率函数 $Pr_{cr(s)}(j)$ 应满足:

$$Pr_{cr(s)}(j) = \sum_{t=0}^{s-1} Pr_{cr(t,s)}(j) \geq Pr_{cr(s-1,s)}(j) \quad (17)$$

由此可见,网络中的交叉冲突绝对不是一个小问题,在进行算法设计时应当予以充分考虑.

### 3 冲突分类解析策略

从冲突分类模型的角度,不难发现,传统退避算法并没有对网络中的分组冲突进行区分,而是对所有的冲突采取相同的策略进行解析.以IEEE 802.11 DCF协议为例,在每次冲突后,节点进入下一退避阶段,同时加倍自己

的竞争窗口,重新选取退避计数器值.事实上,该策略仅适用于同级冲突的解析,可以通过窗口的增大获得更多的随机性,在从一定程度上缓解同级冲突.然而,由于竞争窗口的加大,不可避免地要引入更多的空闲时隙,对分组延迟造成了影响.更为不利的是,只要不同节点的竞争窗口之间存在着交集,就有可能引发交叉冲突.特别地,当节点的竞争窗口到达其上限值  $CW_{max}$  时,交叉冲突的概率更是可以无穷逼近 1.

我们认为,对分组冲突进行合理的分类是实现冲突有效解析的关键,并且冲突产生原因不同,其解析方式也应该有所区别,以体现冲突的不同性质.因此,对于不同的冲突类型,应该采取适合各自特点的冲突解析策略.

根据冲突分类模型,一个合格的退避算法应该能够同时解析这两种不同类型的冲突,而且还要保证引入的开销要尽可能小.下面,我们将分别讨论交叉冲突和同级冲突的解析策略.

### 3.1 交叉冲突解析

交叉冲突是网络中的一类重要冲突,它的解析效果直接影响着整个网络的性能.

#### 3.1.1 顺序离散窗口分布机制 SDWD

我们从交叉冲突的定义中受到启发:由于退避过程的随机性,只要节点的退避窗口之间存在着交集,就有可能引发交叉冲突.节点只有分布在独立的窗口范围内才能够完全消除重叠区域,从根本上杜绝交叉冲突的发生.基于此,我们提出了顺序离散窗口分布机制(sequential discrete window distribution,简称 SDWD).其根本思想是,通过分布窗口的离散化达到消除交叉冲突的目的.

具体策略为:将节点的竞争窗口范围划分成一个初始竞争窗口  $CW_0$  和若干个动态分配的基本窗口 (elementary window,简称 EW).系统初始化时,节点均匀分布在初始竞争窗口上.一旦发生分组冲突,立刻为冲突节点分配一个独立的基本窗口.由于基本窗口是动态创建的,并且每次创建后都添加到当前竞争窗口(current contention window)的末端,这样就能保证各基本窗口之间的独立性,实现了退避空间的不重叠,从源头上避免了交叉冲突的发生.

图 3 是顺序离散窗口分布策略示意图,图中的实线箭头代表一次分组发送.可以看出,系统初始化时,所有的节点都随机分布在初始竞争窗口  $CW_0$  上.由于节点 1 和节点 2 选取了同样的退避计数器值,导致它们在同一时刻发送分组,于是产生冲突.根据 SDWD 机制,紧接着初始竞争窗口分配一个基本窗口  $EW1$ ,同时将冲突节点 1 和节点 2 的竞争窗口范围限定在该基本窗口内.这样,节点 1 和节点 2 将会在  $EW1$  上重新选取自己的退避计数器值为  $1'$  和  $2'$ ,完成了本次退避.对于节点 3 和节点 4 来讲,经过一次退避后,两个节点重新在为它们分配的基本窗口  $EW2$  上随机选取自己的退避计数器值.碰巧他们又选取了同一时隙作为自己的发送时刻,于是这两个节点在  $EW2$  上将再次产生冲突.SDWD 将继续为它们分配另一个独立的基本窗口  $EW4$ .依此类推,直至最终实现冲突解析.

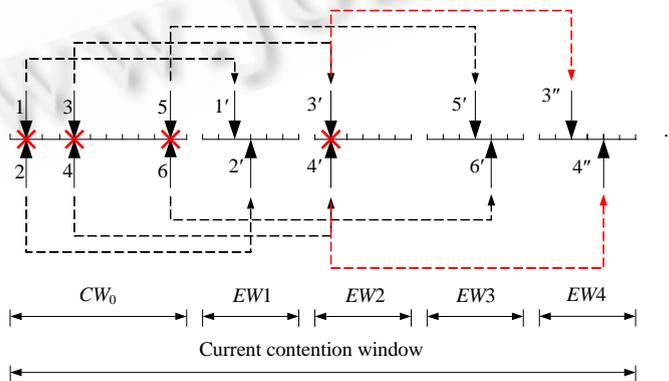


Fig.3 Illustration of sequential discrete window distribution scheme

图 3 顺序离散窗口分布策略示意图

3.1.2 冲突概率分析

我们以 IEEE 802.11 DCF 协议为例,对比分析 SDWD 机制的冲突概率。

如果至少两个节点同时发送分组,就会导致冲突.假定共有  $n$  个节点随机分布在竞争窗口  $[0, m-1]$  上( $m$  代表竞争窗口大小),则冲突概率  $Pr_m^n$  满足:

$$Pr_m^n = \begin{cases} 1, & n > m \\ 1 - \frac{m!}{m^n(m-n)!}, & n \leq m \end{cases} \quad (18)$$

我们定义节点的分布密度  $\rho$  为

$$\rho = n/m \quad (19)$$

于是可以得到分组发送成功的概率  $Pr_m^\rho\{Succeed\}$  满足:

$$Pr_m^\rho\{Succeed\} = \begin{cases} 0, & \rho > 1 \\ \frac{m!}{m^{\rho m}(m-\rho m)!}, & 0 < \rho \leq 1 \end{cases} \quad (20)$$

考虑一个典型的 IEEE 802.11 DCF 退避场景,如图 4 所示.假定在退避阶段  $i$  共有  $n(n < m)$  个节点均匀分布在竞争窗口  $[0, m-1]$  上.在第  $j$  个时隙,共有  $k(k \in [2, \min(n, 2m-2n)])$  个节点同时发送分组,产生冲突.于是,这  $k$  个节点全部进入退避阶段  $(i+1)$ ,并且在扩大的竞争窗口  $[0, 2m-1]$  上重新分布.此时,对处于退避阶段  $(i+1)$  的节点来讲,其分布密度不再是均匀的,新的竞争窗口也因此被分成两部分:重叠部分(intersection part)和剩余部分(rest part),分别对应交叉冲突区域和同级冲突区域.这两个区域的分布密度分别满足:

$$\rho_{Rst} = k/2m \quad (21)$$

$$\rho_{Int} = \rho_i + \rho_{Rst} = n/m + k/2m \quad (22)$$

其中,  $\rho_i$  是退避阶段  $i$  的原始分布密度,  $\rho_{Rst}$  和  $\rho_{Int}$  分别代表退避阶段  $(i+1)$  的剩余部分及重叠部分的分布密度.因此,对 DCF 协议而言,在新的竞争窗口上成功发送分组的概率  $Pr_{DCF}\{Succeed\}$  为

$$Pr_{DCF}\{Succeed\} = Pr_{DCF}\{Succeed \text{ in Intersection Part}\} \times Pr_{DCF}\{Succeed \text{ in Rest Part}\} \quad (23)$$

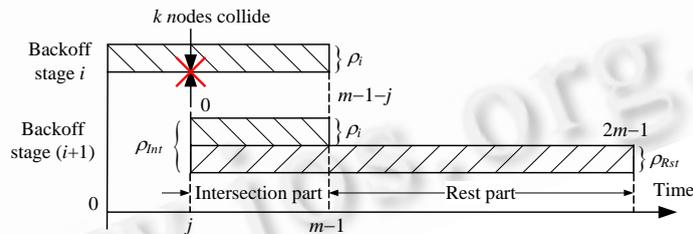


Fig.4 Illustration of distribution density

图 4 节点分布密度示意图

联立方程(20)~方程(23)可以得到 DCF 协议的冲突概率函数  $Pr_{DCF}(j)$  满足:

$$\begin{aligned} Pr_{DCF}(j) &= 1 - Pr_{DCF}\{Succeed\} \\ &= 1 - \frac{(m-j)!}{(m-j)^{\frac{2n+k}{2m}(m-j)} \left[ m-j - \frac{2n+k}{2m}(m-j) \right]!} \times \frac{(m+j)!}{(m+j)^{\frac{k}{2m}(m+j)} \left[ m+j - \frac{k}{2m}(m+j) \right]!} \end{aligned} \quad (24)$$

再来分析 SDWD 机制.  $k$  个冲突节点重新在一个独立的基本窗口上均匀分布,避免了交叉冲突,此时网络中只有同级冲突存在.假设:

$$sizeof(EW) = m \quad (25)$$

可以得到在 SDWD 机制调度下,算法的冲突概率  $Pr_{SDWD}(j)$  为

$$Pr_{SDWD}(j) = 1 - \frac{(m-j)!}{(m-j)^{\frac{m}{m}} \left[ m-j - \frac{n}{m}(m-j) \right]!} \times \frac{m!}{m^k (m-k)!} \quad (26)$$

图 5 对比了 DCF 协议和 SDWD 机制的冲突概率(活跃节点数  $n$  变化,竞争窗口大小  $m=64$ ,冲突节点数  $k=2$ ).我们固定竞争窗口大小  $m$  以及冲突节点数  $k$ ,仅仅改变活跃节点数  $n$ ,观察冲突概率的变化.不难想象,活跃节点数的增加势必加剧信道竞争,从而导致更高的冲突概率.从图中可以看出,在活跃节点数相同的条件下,SDWD 的冲突概率几乎始终低于 DCF.

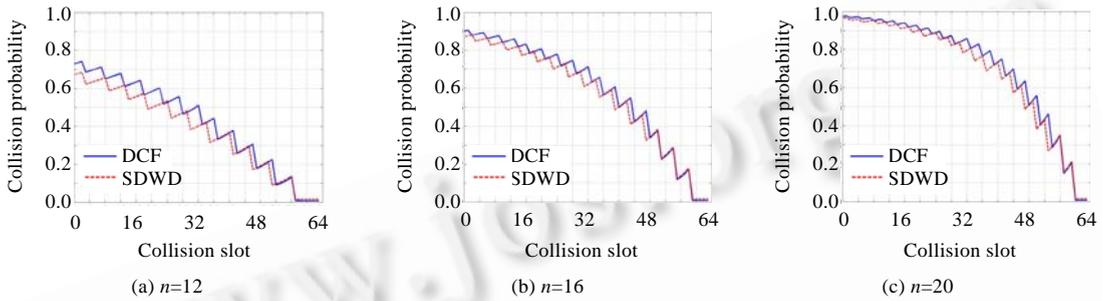


Fig.5 Collision probability of DCF and SDWD

图 5 DCF 和 SDWD 冲突概率比较

图 6 显示了冲突节点数对冲突概率的影响规律(冲突节点数  $k$  改变,竞争窗口大小  $m=64$ ,活跃节点数  $n=20$ ).SDWD 机制避免了交叉冲突,网络中只有同级冲突存在.在这种情况下,冲突节点数越多,发生同级冲突的概率就越大.所以同 DCF 协议相比,SDWD 机制冲突概率的下降相对缓慢.特别在冲突节点数较多时,随着冲突时隙的增大,SDWD 机制的冲突概率逐渐高于 DCF.也就是说,冲突节点数越多,SDWD 机制的冲突解析效果越得不到保障.然而, $k$  个节点在同一时刻发送分组,从而导致分组冲突的概率满足:

$$Pr\{k \text{ Nodes Collided}\} = \frac{\binom{n}{k} (m-1)^{n-k}}{m^n} \quad (27)$$

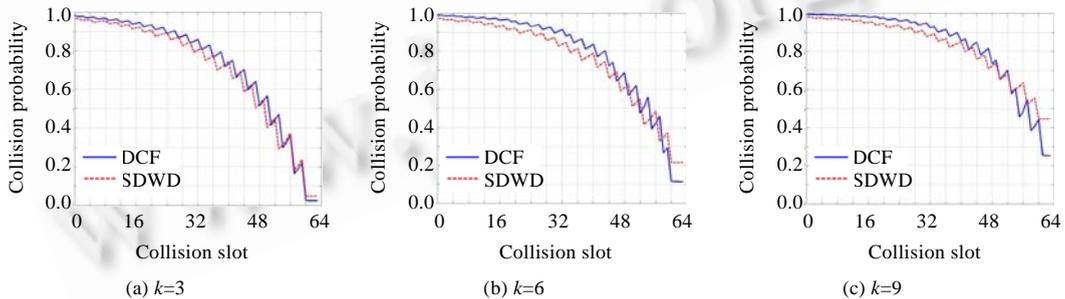


Fig.6 Collision probability of DCF and SDWD

图 6 DCF 和 SDWD 冲突概率比较

图 7 给出了冲突节点数与冲突发生概率之间的关系.从图中可以明显看出,随着冲突节点数目的增大,冲突发生概率急剧下降并趋于 0.根据实际推断原理<sup>[14]</sup>:概率很小的事件在一次实验中几乎是不发生的.这也就意味着:多个节点在同一时刻发生冲突的概率几乎为 0.因此,我们完全可以忽略 SDWD 机制中由多个节点同时发送分组带来的高冲突概率的影响.

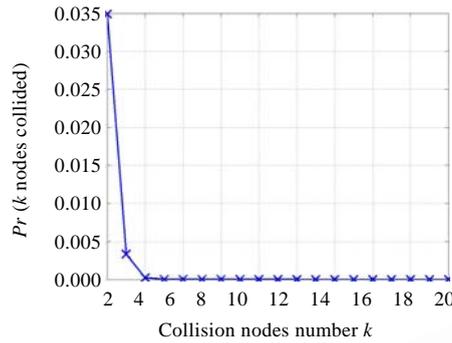


Fig.7 Relation between collided nodes number and collision probability

图 7 冲突节点数目与冲突发生概率关系

### 3.2 同级冲突解析

前面已经提到,SDWD 机制通过分布窗口的离散化消除了交叉冲突,将交叉冲突的解析转化为同级冲突解析,因此,我们有必要在此处对同级冲突解析中的问题进行分析。

同级冲突的解析策略非常简单,可以采取传统算法中的解析模式,通过扩大分布窗口范围,增加退避机制的随机性来降低冲突概率。

在 SDWD 机制调度下,同级冲突主要发生在基本窗口上(发生在初始竞争窗口范围内的同级冲突相对来说很少,可以忽略)。我们知道,如果至少两个节点同时发送分组,就会导致冲突。考虑到方程(27):多个节点同时发生冲突的概率可以小到忽略不计,这里我们仅分析两个节点发生冲突时的状况,于是可以得到同级冲突概率为

$$Pr\{\text{Collision in Elementary Window}\} = \frac{1}{\text{sizeof}(\text{Elementary Window})} = \frac{1}{m} \quad (28)$$

此处, $m$  表示基本窗口大小。由此可见,基本窗口越小,发生同级冲突的概率越大。

同时,空闲时隙的数学期望  $E(T_{idle})$  可以表示为

$$E(T_{idle}) = \sum_{i=0}^{m-1} i \times Pr\{\text{Backoff Counter} = i\} = \sum_{i=0}^{m-1} i \times \frac{\binom{2}{1}^{m-i}}{m^2} = \frac{2}{m^2} \sum_{i=1}^{m-1} (m-i)i \quad (29)$$

这是一个连续单调递增函数,窗口越大,平均空闲时间越长。对比方程(28)和方程(29)可知,基本窗口的大小同时控制着同级冲突的概率以及空闲时隙的开销。

## 4 基于分类模型的冲突解析算法

在分别确定了针对交叉冲突和同级冲突的解析策略后,接下来设计冲突解析算法。

首先考虑交叉冲突解析。为了避免交叉冲突,如何保证各个基本窗口之间的独立性是一个亟待解决的问题。为了协调基本窗口的分配,保障各分布窗口的离散性,我们使用参数竞争级别(contention level,简称 CL)来指导基本窗口的动态分配。竞争级别定义为发生在公共信道上的分组冲突次数。由于无线信道的共享特性,一旦发生分组碰撞,处于该竞争区域内的所有节点都能够及时监测到该冲突事件,从而保证了竞争节点间的同步。我们利用顺序离散窗口分布机制 SDWD 将冲突节点分布在一系列离散的基本窗口上,将交叉冲突解析转变为同级冲突解析,并采用传统的基于概率的冲突解析模式处理同级冲突。

在算法实现过程中,针对发送成功的节点可以采取不同的处理方式。基于此,我们提出了两种不同的冲突解析算法:冲突分类解析算法 CCR 和无冲突分类解析算法 CF-CCR。

### 4.1 冲突分类解析算法 CCR

CCR 算法的实现过程是:

(1) 所有节点监测信道.在信道连续空闲 *DIFS* 后,如果再空闲一个时隙,节点将其退避计数器值减 1.一旦退避计数器值到达 0,节点立刻发送分组.倘若信道由闲变忙,退避进程挂起,直到信道再次空闲 *DIFS* 后继续递减计数器值;

(2) 阻塞节点监测信道状态.每当节点监测到网络中发生了一次冲突,则将 *CL* 值加 1:

$$CL=CL+1 \quad \text{upon collision.}$$

(3) 发送节点完成数据分组的发送后,如果在规定的时间内没有收到接收方发来的确认信息,则认为发生了分组冲突(我们假定信道是无差错的,所有的分组丢失都是由冲突造成).于是,增加 *CL* 值以反映信道的冲突状况,同时进行分组重传次数检查:

- a) 如果分组重传次数超过了预设的最大门限值 *Max\_Retrans\_Times*,节点将该分组丢弃,并且重设竞争窗口为 $[0, CW_0-1]$ ,同时保留 *CL* 值以继续指导后续的窗口分配;
- b) 否则,节点设置其新的竞争窗口范围为 $[CW_0+(CL-1) \times EW, CW_0+CL \times EW-1]$ .

显然,在每次重传之前,*CL* 值都不小于 1.通过这种分布策略,冲突节点将会在一个紧接当前竞争窗口的的基本窗口上重新分布,从而实现了退避空间的独立性,避免了交叉冲突,体现了 *SDWD* 机制的思想.

(4) 在每次成功传输后(数据分组传输加上 *ACK* 分组传输),发送节点设置其竞争窗口为 $[0, CW_0+CL \times EW-1]$ ,从而与分布在当前竞争窗口范围内的所有邻居节点一起竞争信道,为其他节点提供了信道接入机会,保障了协议的公平性.

*CCR* 算法为每个冲突节点分配一个独立的基本窗口,以此避免交叉冲突的发生.同时,节点一旦发送成功,则将自己的竞争窗口设为当前竞争窗口,从而同所有阻塞节点一起重新竞争信道的使用,保障了公平性.

### 4.2 无冲突分类解析算法CF-CCR

*CCR* 仅仅为冲突节点分配独立的竞争窗口,而成功节点只能在当前竞争窗口范围内重新分布.尽管这样可以节约成功节点的信道接入时间,但同时使得其必须与所有的阻塞节点一起重新竞争信道的使用,由此可能引发新的分组冲突.

如图 8 所示,节点 1 和节点 2 在时刻 *h* 发生冲突后,重新在基本窗口 *EW1* 上分布,选取的发送时刻分别为 *k*(节点 1')和 *l*(节点 2').节点 5 发送成功后,根据 *CCR* 算法规定,设置自己的竞争窗口范围为 $[j, CW_0+2 \times EW]$ ,并随机选取退避计数器值.节点 5 刚好选取 *l* 时刻作为自己下次发送的时隙(用 5'表示),而该时隙已经被节点 2'提前占用,所以导致系统在 *l* 时刻再次发生冲突.可以看到,原本节点 2 经过退避以后选择在时刻 *l* 的发送是成功的,可是由于节点 5 的介入才导致了冲突.由此可见,*CCR* 算法的这种解析策略并不能完全消除交叉冲突,影响了冲突解析效率.

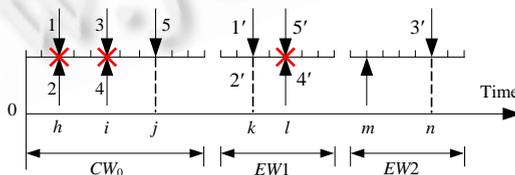


Fig.8 Collisions in collision classification resolution algorithm

图 8 CCR 算法中的分组冲突

实际上,*SDWD* 机制只是规定了对于冲突节点应该采取的处理方式,并没有提出针对成功节点的解析策略,这给予了我们一定程度上的灵活性.我们完全可以利用 *SDWD* 机制,采取严格的分时策略来彻底消除网络中的分组冲突(包括交叉冲突和同级冲突).其基本思想是:无论分组的发送结果如何(发送成功或冲突),都为发送节点分配一个独立的分布窗口,使得节点依次顺序分布在当前竞争窗口上,从而实现冲突的完全避免.我们称其为

无冲突分类解析算法 CF-CCR.

CF-CCR 同样基于 SDWD 机制,只是修改了竞争级别的定义,将其看成是在竞争区域内发生的信道接入次数.每当节点监测到信道上开始一个新的传输时,不论传输成功与否,都相应地将  $CL$  值加 1.CCR 和 CF-CCR 的主要区别还体现在对成功节点的处理上.CCR 将成功节点在当前竞争窗口范围内重新分布,而 CF-CCR 则为其单独分配一个独立的基本窗口,对其采取与冲突节点相同的处理方式.在 CF-CCR 调度下,节点在分组发送成功后增加自己的  $CL$  值,并相应设置竞争窗口范围为  $[CW_0+(CL-1)\times EW, CW_0+CL\times EW-1]$ .这样,节点就能够完全单独(exclusive)占有第  $CL$  个基本窗口.根据 SDWD 机制的窗口离散化原则,只要节点成功发送一次,则其后续分组发送都能够在无干扰的情况下进行.在 CF-CCR 调度下,节点依次接入信道,完成分组发送任务后(不论发送是否成功),主动将自己的退避窗口设置在当前竞争窗口的末端,从而将信道接入机会移交给其他节点,自己则等待一轮周期过后再次接入信道.一旦某个节点成功完成一次分组发送,则由于 CF-CCR 调度策略的独占性,该节点的后续传输都可以顺利进行.这样,就能够彻底杜绝网络中的交叉冲突.可以想象,在没有其他节点加入网络的情况下\*\*,系统最终会收敛到一个稳定状态.此时,节点依次顺序接入信道,整个网络中没有任何冲突(包括交叉冲突和同级冲突)发生.

实际上,两个算法的区别一个体现在对基本窗口的分配原则上,另一个则表现为对发送成功节点的处理方式不同.CF-CCR 调度节点依次顺序接入信道,一旦完成分组发送,不论发送成功与否,节点都将自己安排在当前竞争窗口的末端,等待一轮周期过后,再次接入信道.显然,与 CCR 相比,CF-CCR 能够获得更少的冲突次数和更多的公平性.

## 5 性能评估

### 5.1 实验模型

我们利用 ns-2<sup>[15]</sup>仿真平台进行实验,从多个方面对 CCR,CF-CCR 以及 IEEE 802.11 DCF 协议的性能进行比较.仿真基于直接序列扩频 DSSS 规范,采用 IEEE 802.11 无线网络标准参数<sup>[2]</sup>,见表 1.

Table 1 Experimental parameters

表 1 实验参数

| Parameters       | Value      | Parameters  | Value      |
|------------------|------------|-------------|------------|
| SIFS             | 10 $\mu$ s | $CW_{min}$  | 32         |
| DIFS             | 50 $\mu$ s | $CW_{max}$  | 2 048      |
| Slot interval    | 20 $\mu$ s | $CW_0$      | 32         |
| Preamble length  | 144 bits   | $EW$        | 16         |
| Channel bit rate | 2 Mbps     | Packet size | 1 024 Byte |

实验在一个 WLAN 环境中进行,只有一个目的节点,即接入点 AP,所有的源节点随机分布在一个  $100m\times 100m$  区域内,节点的传输半径为 150m.采用基于 CBR 的 UDP 数据流进行通信,并且假设系统具有饱和的流量.也就是说,在任意时刻,每个节点上总有至少一个分组等待发送<sup>[16,17]</sup>.

### 5.2 性能比较和分析

由于分组传输失败对系统吞吐量和延迟造成很大影响<sup>[18]</sup>,因此我们需要考察算法对分组冲突的解析能力.另外,吞吐量、延迟和公平性是考量退避算法性能的主要指标<sup>[5,19]</sup>.同时,为了体现算法对服务质量的支持,我们对算法的延迟抖动特性进行了分析.最后,还考察了基本窗口大小对算法性能的影响.

#### 1) 冲突率

为了考察算法对分组冲突的解析能力,我们定义冲突率(collision rate,简称 CR)为

\*\* 考虑节点动态加入或离开网络的情况,可以在节点间采取某种通信手段,以实现新加入节点的  $CL$  值与网络其他节点的同步.该算法将在后续研究中实现.

$$CR = \frac{T_{col}}{Pkt_{Rx}},$$

其中,  $T_{col}$  表示传输期间发生的冲突次数,  $Pkt_{Rx}$  代表接收端成功接收的分组数目. 冲突率越小, 冲突解析能力越强. 冲突的存在, 不仅浪费了系统带宽, 消耗了节点能量, 而且降低了网络吞吐量, 增加了分组延迟<sup>[20]</sup>, 对系统性能造成了很大影响. 一个合格的退避算法应该在对分组冲突进行有效解析的同时, 尽可能降低系统开销<sup>[21]</sup> (例如控制分组延迟等).

图 9 所示为改变活跃节点数目和计算冲突率的结果. 我们知道, 随着竞争节点数目的增加, 信道竞争状况必然加剧, 从而导致更多的冲突. 从图中可以看出, DCF 协议的冲突率明显高于 CCR 和 CF-CCR. 由此可以说明: 作为这两种算法的基础, SDWD 机制在冲突解析方面效果显著. 从图中还可以看出, CF-CCR 的冲突率始终比 CCR 算法要低. 这是由于 CF-CCR 通过严格的分时策略完全消除了交叉冲突, 同时保障了成功节点的后续传输.

## 2) 吞吐量

通过对比算法的归一化吞吐量(normalized throughput, 简称 NT)<sup>[22]</sup> 来衡量算法的吞吐量性能. 从图 10 中可以看出, IEEE 802.11 网络的吞吐量随着竞争节点数目的增加而显著下降. 由于采取了 SDWD 机制对冲突节点的分布窗口进行离散化处理, 避免了交叉冲突, 因此, 与 DCF 相比, CCR 和 CF-CCR 提高了分组成功发送的概率, 从而获得更高的系统吞吐量.

另一方面, 由于当前基本窗口尺寸设为 16, 从而可以将网络中的同级冲突概率控制在某一水平以下. 在这种情况下, CCR 将成功节点的分布范围设为整个竞争窗口, 而不是像 CF-CCR 那样将其分布在当前竞争窗口末端的一个基本窗口上, 因此, CCR 获得的吞吐量要稍高些.

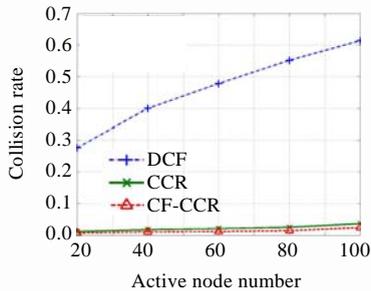


Fig.9 Collision rate comparison results

图 9 冲突率比较结果

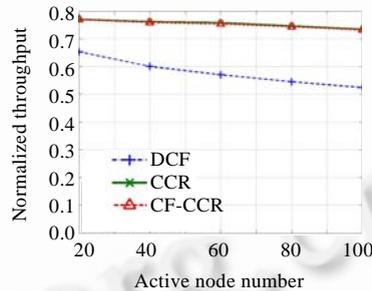


Fig.10 Throughput comparison results

图 10 吞吐量比较结果

## 3) 分组延迟

分组延迟(delay)是指分组从到达发送队列头部开始直到被接收方正确接收为止所经历的平均时延, 能够反映算法的调度效率<sup>[23]</sup>. 从图 11 中可以看出, CCR 和 CF-CCR 的分组延迟很接近, 延迟增加幅度比 DCF 协议更加平缓. 由于 DCF 协议的冲突解析效率不高, 因此节点需要耗费更多的时间从冲突中恢复, 无疑延长了分组正确接收的时间, 增加了延迟. 而在 CCR 和 CF-CCR 中, 虽然竞争窗口以基本窗口大小为步长持续增加, 会在一定程度上增加分组延迟, 然而与此同时, 分组冲突的概率却大为降低, 最终获得更低的分组延迟.

与吞吐量的分析结果一样, 当基本窗口大小设为 16 时, 作为两种算法的基础, SDWD 机制的冲突解析效率较高, 发生冲突的次数在可以容忍的范围内. 在这种情况下, CCR 将发送成功的节点随机分布在当前竞争窗口范围内, 于是节点就有机会抢占信道, 节省接入时间, 从而获得比 CF-CCR 更低的分组延迟.

## 4) 公平性

我们采取文献[24]中定义的公平性指数(fairness index, 简称 FI)来衡量协议的公平性. 该公平指数定义如下:

$$FI = \left( \frac{\sum_{i=1}^n T_i / \phi_i}{n \times \sum_{i=1}^n (T_i / \phi_i)^2} \right)^2,$$

其中,  $n$  表示数据流的数目,  $T_i$  代表流  $i$  的吞吐量,  $\phi_i$  是流  $i$  的权重(我们假定所有节点权重相同).由柯西-苏瓦茨不等式<sup>[14]</sup>可知,公平指数  $FI$  满足  $FI \leq 1$ , 当且仅当所有的  $T_i/\phi_i$  都相等时取等号.因此,  $FI$  越趋近于 1, 说明公平性越好.

从图 12 中可以清楚地看到,随着节点数目的增多,DCF 的公平指数曲线大幅跌落,而 CCR 和 CF-CCR 的只是稍有下降,并且 CF-CCR 的下降幅度最为平缓.由此充分说明了两种算法都能提供比 DCF 协议更好的公平性,并且 CF-CCR 算法的公平性更高.这是由于两种算法通过分布窗口的顺序排列,避免了 DCF 中窗口重设带来的信道长期被占用的问题,保障了公平性.

与 CCR 算法相比,CF-CCR 执行更加严格的分时调度策略,按照节点的发送顺序依次将其分布在时间轴上.一旦完成本次分组发送(不论发送的结果如何),信道接入机会立刻发生转移,节点需要等待一轮周期过后才能再次接入信道.因此,在 CF-CCR 算法中,并不存在 CCR 中的成功节点需要同阻塞节点共同竞争信道的问题,从而能够获得更多的公平性.

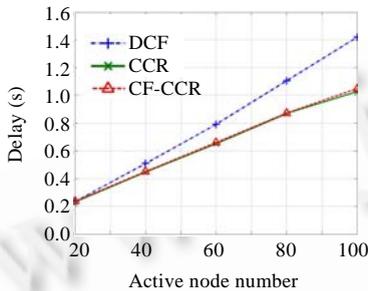


Fig.11 Delay comparison results

图 11 分组延迟比较结果

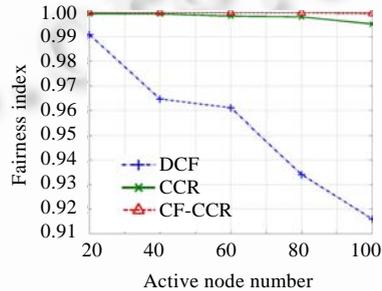


Fig.12 Fairness comparison results

图 12 公平性比较结果

5) 延迟抖动

延迟抖动(delay jitter)是指分组通过网络连接到目的地过程中分组延迟的变化程度,是考察网络服务质量的一个重要性能指标.在理想情况下,分组以同样的延迟穿过网络,到达目的地.然而由于各种原因,包括网络阻塞、流量动态变化、分组冲突等,各个分组所经历的延迟往往不尽相同<sup>[25]</sup>,从而导致延迟抖动的差异.通常来讲,延迟抖动越小,获得的流量越稳定.

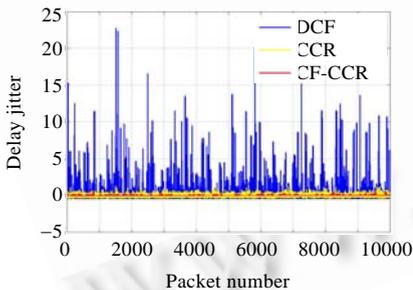


Fig.13 Delay jitter comparison results

图 13 延迟抖动比较结果

为了比较算法的延迟抖动结果,我们选取 40 个节点进行仿真实验,分别记录下 10 000 个分组的延迟大小,统计得出分组延迟抖动值.其中,基本窗口尺寸设为 16,分组大小为 1024 Byte,结果如图 13 所示.

从图 13 可以明显看出,DCF 协议的延迟抖动最厉害,CF-CCR 算法的抖动最小.这是由于同 DCF 协议相比,两个算法均采用了竞争窗口离散化的方式解析冲突,有效降低了分组冲突概率,同时避免了竞争窗口设置过大给延迟抖动带来的影响,从而获得更加稳定的分组延迟.另外,由于 CF-CCR 完全消除了交叉冲突,并且保障了成功节点的后续发送,使得系统可以收敛到完全没有冲突的状态,因此可以获得最小的延迟抖动.

从图 13 可以明显看出,DCF 协议的延迟抖动最厉害,CF-CCR 算法的抖动最小.这是由于同 DCF 协议相比,两个算法均采用了竞争窗口离散化的方式解析冲突,有效降低了分组冲突概率,同时避免了竞争窗口设置过大给延迟抖动带来的影响,从而获得更加稳定的分组延迟.另外,由于 CF-CCR 完全消除了交叉冲突,并且保障了成功节点的后续发送,使得系统可以收敛到完全没有冲突的状态,因此可以获得最小的延迟抖动.

6) 基本窗口尺寸对算法性能的影响

正如我们前面提到的,基本窗口的大小对系统性能有较大影响,它同时决定着同级冲突的概率和分组延迟的大小.我们选择 40 个节点进行实验,通过改变基本窗口的大小观察协议性能的变化,结果如图 14 所示.

与 IEEE 802.11 DCF 协议相比,CCR 算法能够有效解析网络中的交叉冲突,但却无法保证完全清除.然而,CF-CCR 却能够通过严格的调度策略消除交叉冲突,系统中仅剩同级冲突.同时,同级冲突概率可以随着基本窗口尺寸的增大而有效降低.此外,在 CF-CCR 算法的调度下,系统最终能够收敛到一个无冲突状态.因此,如图 14 (a)所示,CF-CCR 的冲突率始终低于 CCR.另外,对同一算法而言,分组尺寸越大,其发送持续时间越长,邻居节点

的阻塞时间相应越久,单位时间内节点的接入次数减小,所以冲突率更小。

增大的窗口尺寸必然引入更多的空闲等待时隙,造成分组延迟的增加.从图 14(b)中可以看出,CCR 算法延迟增加的幅度比 CF-CCR 平缓,而拐点就在  $EW=16$  处.当基本窗口尺寸小于该门限值时,由于 CF-CCR 算法冲突解析效果更好,因此分组延迟小于 CCR;一旦基本窗口尺寸变得足够大,分组冲突已经不是影响系统性能的主要因素,那么冗余的窗口空间就只能造成分组延迟的增加。

如图 14(c)所示,随着窗口尺寸的增大,两个算法的吞吐量都在下降,而 CF-CCR 下降得更快一些.当基本窗口很小时,引入的空闲等待时隙数相对较少,分组冲突就成了决定系统吞吐量的主导因素.此时,算法的冲突解析效率越高,获得的系统吞吐量也就越大.随着基本窗口尺寸的增加,分组延迟的影响逐渐突出.而 CCR 算法可以提供更少的延迟,对于饱和流量而言,能够有效提高信道利用率,从而获得更高的系统吞吐量。

因此,我们可以根据实际需求来调整基本窗口大小,提供相应的服务质量保证.例如,在可以忽略冲突率影响的前提下,通常可以选取较小的基本窗口尺寸以获得较低的分组延迟和较大的系统吞吐量。

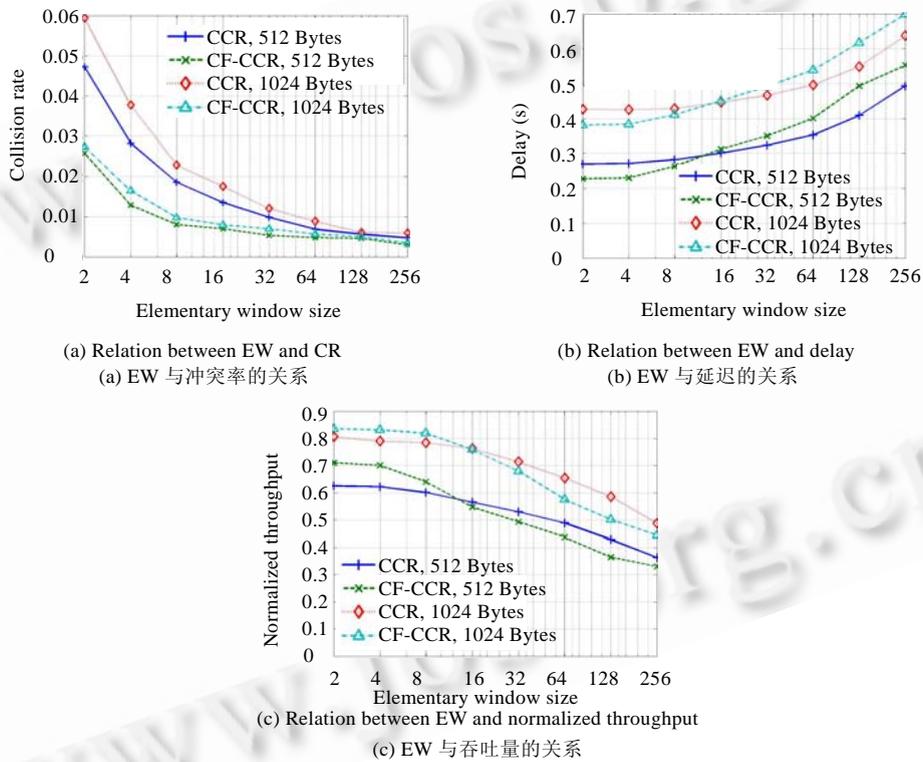


Fig.14 Impact of elementary window size on CCR & CF-CCR algorithms

图 14 基本窗口大小对 CCR 和 CF-CCR 算法性能的影响

### 6 结论

传统的退避算法主要依靠退避过程的随机性来实现冲突的解析.通常,这种基于随机性的策略并不能有效地保证冲突的解析效果,有时甚至会导致更多的冲突.此外,现有的退避算法并没有对冲突的类型进行区分,而是对所有的冲突采取相同的解析策略,并没有考虑到不同冲突的不同特点.我们认为,对冲突的类型进行有效合理的区分是实现冲突可靠解析的前提。

本文首次提出一种冲突分类模型,根据冲突节点所处退避阶段的不同,将冲突分成两类:交叉冲突和同级冲突,通过分析证明:网络中的交叉冲突是普遍存在的.随后,提出了针对交叉冲突和同级冲突的不同解析策略:利

用顺序离散窗口分布机制 SDWD,通过竞争窗口的离散化来消除交叉冲突.同时,通过设置合适的基本窗口尺寸减小同级冲突概率,保证分组延迟.在此基础上,我们提出了两种算法:冲突分类解析算法 CCR 和无冲突分类解析算法 CF-CCR.CCR 将成功节点随机分布在当前竞争窗口上以节约接入时间,可是却有可能导致新的冲突.为了完全消除冲突,CF-CCR 对于发送成功的节点采取和冲突节点同样的处理方式,即为它们分配独立的基本窗口.这样不但避免了产生新的冲突,而且还保障了成功节点的后续传输.在 CF-CCR 算法调度下,系统最终会收敛到一个无冲突的稳定状态.

仿真结果表明,CCR 和 CF-CCR 在冲突解析、吞吐量、分组延迟、公平性以及延迟抖动方面均优于 IEEE 802.11 DCF 协议.同时,这两个算法的适用场合不同.同 CCR 相比,CF-CCR 能够获得更低的冲突率、更多的公平性以及更小的延迟抖动.如果基本窗口尺寸足够大,能够有效完成同级冲突的解析,则采用 CCR 算法能够获得更高的吞吐量和更少的分组延迟;否则,采用 CF-CCR 效果更好.

## References:

- [1] Sun BL, Li LY. A QoS based multicast routing protocol in ad hoc networks. Chinese Journal of Computers, 2004,27(10): 1402–1407 (in Chinese with English abstract).
- [2] LAN MAN Standards Committee of the IEEE Computer Society. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. The Institute of Electrical and Electronic Engineers, 1999. <http://www.csse.uwa.edu.au/adhocnets/802.11-1999.pdf>
- [3] Kwak BJ, Song NO, Miller LE. Performance analysis of exponential backoff. IEEE/ACM Trans. on Networking, 2005,13(2): 343–355.
- [4] Bianchi G. Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications, 2000,18(3):535–547.
- [5] Haas ZJ, Deng J. On optimizing the backoff interval for random access schemes. IEEE Trans. on Communications, 2003,51(12): 2081–2090.
- [6] Bharghavan V, Demers A, Shenker S, Zhang LX. MACAW: A media access protocol for wireless LAN's. In: Proc. of the ACM SIGCOMM. London, 1994. 212–225. [http://portal.acm.org/ft\\_gateway.cfm?id=190334&type=pdf](http://portal.acm.org/ft_gateway.cfm?id=190334&type=pdf)
- [7] He H, Li JD, Sheng M, Li WY. A novel SD\_DCC backoff algorithm for wireless LAN based on slow decrease. Chinese Journal of Computers, 2005,28(11):1907–1914 (in Chinese with English abstract).
- [8] Peng Y, Cheng SD. A self-adaptive wireless LAN protocol. Journal of Software, 2004,15(4):604–615 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/15/604.htm>
- [9] Aad I, Ni Q, Barakat C, Turetli T. Enhancing IEEE 802.11 MAC in congested environments. In: Proc. of the 4th Workshop on Applications and Services in Wireless Networks. 2004. 82–91. <http://sciencestage.com/d/1510350/enhancing-ieee-802-11-mac-in-congested-environments-1.html>
- [10] Wu HT, Cheng SD, Peng Y, Long KP, Ma J. IEEE 802.11 distributed coordination function (DCF): Analysis and enhancement. In: Proc. of the IEEE ICC. 2002. 605–609. <http://ieeexplore.ieee.org/iel5/7828/21514/00996924.pdf>
- [11] Pang QX, Liew SC, Lee JYB, Leung VCM. Performance evaluation of an adaptive backoff scheme for WLAN. Wireless Communications and Mobile Computing, 2004,4:867–879.
- [12] Deng J, Varshney PK, Haas ZJ. A new backoff algorithm for the IEEE 802.11 distributed coordination function. In: Proc. of the CNDS. 2004. 215–225. <http://citeseer.ist.psu.edu/631198.html>
- [13] Wang CQ, Li B, Li LM. A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF. IEEE Trans. on Vehicular Technology, 2004,53(4):1235–1246.
- [14] Sheng Z. Probability Theory and Statistics. 2nd ed., Shanghai: Shanghai Jiaotong University Press, 2006 (in Chinese).
- [15] Fall K, Varadhan K. The ns Manual. UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2009. <http://www.isi.edu/nsnam/ns/ns-documentation.html>
- [16] Garetto M, Chiasserini CF. Performance analysis of 802.11 WLANs under sporadic traffic. In: Proc. of the IFIP Networking. 2005. 1343–1347. <http://www.springerlink.com/index/xfaj3xma78f7u3hk.pdf>
- [17] Foh CH, Tantra JW. Comments on IEEE 802.11 saturation throughput analysis with freezing of backoff counters. IEEE Communications Letters, 2005,9(2):130–132.

- [18] Xiao YK, Shan XM, Ren Y. Game theory models for IEEE 802.11 DCF in wireless ad hoc networks. *IEEE Radio Communications*, 2005:S22–S26.
- [19] Barowski Y, Biaz S, Agrawal P. Towards the performance analysis of IEEE 802.11 in multi-hop ad-hoc networks. In: *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*. 2005. 100–106. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1424483](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1424483)
- [20] Lin IH, Pan JY. Throughput analysis of a novel backoff algorithm for IEEE 802.11 WLANs. In: *Proc. of the Wireless Telecommunications Symp.* 2005. 85–90. <http://ieeexplore.ieee.org/iel5/10227/32603/01524769.pdf?arnumber=1524769>
- [21] Vu HL, Sakurai T. Accurate delay distribution for IEEE 802.11 DCF. *IEEE Communications Letters*, 2006,10(4):317–319.
- [22] Yang X, Vaidya NH. Explicit and implicit pipelining for wireless medium access control (invited paper). In: *Proc. of the IEEE Vehicular Technology Conf. (VTC)*. 2003. 1427–1431. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1285260](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1285260)
- [23] Kwon YG, Fang YG, Latchman H. Fast collision resolution (FCR) MAC algorithm for wireless local area networks. In: *Proc. of the IEEE Globecom*. 2002. 2250–2254. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1189032](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1189032)
- [24] Jain R, Durrresi A, Babic G. Throughput fairness index: an explanation. *ATM Forum*, 1999. <http://www.cse.wustl.edu/~jain/atmf/a99-0045.htm>
- [25] Zhou YK, Sethu H. A simulation study of relationships between delay jitter and properties of a switching network. In: *Proc. of the Applied Telecommunication Symp.* 2000. <http://citeseer.ist.psu.edu/559819.html>

#### 附中文参考文献:

- [1] 孙宝林,李腊元. Ad Hoc 网络 QoS 多播路由协议. *计算机学报*, 2004, 27(10): 1402–1407.
- [7] 何宏,李建东,盛敏,李维英. 一种基于慢退避思想的 SD\_DCC 退避算法及其性能分析. *计算机学报*, 2005, 28(11): 1907–1914.
- [8] 彭泳,程时端. 一种自适应无线局域网协议. *软件学报*, 2004, 15(4): 604–615. <http://www.jos.org.cn/1000-9825/15/604.htm>
- [14] 盛骤. *概率论与数理统计*. 第 2 版, 上海: 上海交通大学出版社, 2006.



张棋飞(1977—),男,湖北武汉人,博士,讲师,主要研究领域为无线通信网络,网络优化技术.



桂超(1965—),男,副教授,主要研究领域为无线自组织网络.



刘威(1977—),男,博士,副教授,主要研究领域为无线通信网络,认知无线电技术.



严冰(1978—),女,讲师,主要研究领域为无线自组织网络.



孙宝林(1963—),男,博士,教授,主要研究领域为移动通信技术,通信协议技术.