

一种 IEEE 802.16 中快速有效的冲突解决算法

肖峻峰 邹仕洪 程时端

(北京邮电大学网络与交换国家重点实验室 北京 100876)

摘要 IEEE 802.16 宽带无线接入系统在其上行媒体接入控制层采用时, 分复用和资源竞争与预留的方式进行接入, 各用户站(SS)之间存在着竞争与冲突, 这将导致系统性能下降。该文针对 SS 在碰撞以后的冲突解决过程提出了一种快速有效的冲突解决算法——动态退避控制(Dynamical Backoff Control, DBC)算法。在该算法中, 基站(BS)通过预测下一个上行帧中将会出现的带宽请求报文数, 动态地控制 SS 的退避范围, 以增大每帧中成功发送的带宽请求报文数, 从而提高 MAC 层的数据吞吐, 降低 SS 的平均接入延迟。利用 OPNET 对 DBC 和二进制指数退避两种算法进行了仿真。仿真结果显示 DBC 算法对系统性能有一定的改善和提高。

关键词 IEEE 802.16, 宽带无线接入, 退避, 冲突解决

中图分类号: TN915.65

文献标识码: A

文章编号: 1009-5896(2006)10-1921-05

A Fast Efficient Collision Resolution Algorithm in IEEE 802.16

Xiao Jun-feng Zou Shi-hong Cheng Shi-duan

(State Key Laboratory of Networking and Switching, Beijing University of Posts & Telecommunications, Beijing 100876, China)

Abstract IEEE 802.16 broadband wireless access system adopts TDMA (Time-Division Multiple Access) and contention-based reservation of resource in its uplink MAC layer. Thus there are contentions and collisions between SSs(Subscriber Stations), and the performance is inevitably deteriorated. A fast and efficient backoff algorithm DBC (Dynamical Backoff Control) is proposed to solve the problem. In DBC, by predicting the number of request messages in next uplink frame, BS controls the optimal backoff ranges dynamically. As a result, the successful rate of sending request messages in each frame is increased, the data throughput of MAC layer is promoted, and the mean access delay of SS is reduced. Simulation is conducted in OPNET to compare DBC with binary exponential backoff. Simulation results demonstrate that proposed DBC algorithm makes a great improvement in system performance.

Key words IEEE 802.16, Broadband wireless access, Backoff, Collision resolution

1 引言

Wi-Fi^[1]的成功让社会看到无线通信的魅力, 激发了人们进一步去追求功能更强大的宽带无线通信系统——宽带无线城域网。IEEE 802.16-REV D^[2]标准作为宽带无线城域网空中接口的规范于 2004 年年底被通过, 该标准是针对用户站 (Subscriber Station, SS) 固定的情况, 而针对 SS 移动的 IEEE 802.16e^[3]标准正在制订和完善中。同时, 为了改善和验证 802.16 设备的兼容性和互操作性, 由世界各大通信设备制造公司联合成立了 WiMax^[4]组织。IEEE 802.16 作为第二代宽带无线接入技术的标准, 具有速率高、配置快速、扩展性强、维护和更新费用更低等优势, 为网络运营商提供了在 Cable, xDSL 和以太网等宽带接入技术之外的一种选择^[5]。

IEEE 802.16 宽带无线接入系统在其上行方向采用时分复用(Time-Division Multiple Access, TDMA)和资源竞争与

预留的方式进行接入。协议一共定义了 4 种上行业务调度类型, 分别是主动授权业务(Unsolicited Grant Service, UGS)、实时轮询业务(Real-Time Polling Service, rtPS)、非实时轮询业务(not-Real-time Polling Service, nrtPS)和尽力而为业务(Best Effort Service, BE)。对于 UGS 和 rtPS 类型, 协议分别采用周期性预留带宽及定期轮询的方式, SS 之间不存在冲突。对于 nrtPS 和 BE 类型, 采用带宽请求与预留的方式, 各 SS 之间存在着竞争与冲突。对于冲突的解决, IEEE 802.16 标准采用简单的二进制指数退避算法, 该算法也应用在以太网中^[6]。尽管二进制指数退避算法实现起来比较简单, 但它也存在一些问题。例如, 退避延迟会随着连续冲突的次数增加而越来越大; 同时, 二进制指数退避窗口所描述的退避范围也不准确, 会造成部分竞争时隙的浪费; Kuo 等^[7]建议了一个多级竞争窗口机制去解决电缆调制解调器(Cable Modems, CMs)之间的冲突, 以使低优先级 CMs 不会影响高优先级 CMs 的带宽请求传输, 但他们的研究只考虑了高优先级 CMs 的性能, 而未考虑整个 DOCSIS^[8]系统的性能; Yin 等^[9]为冲突解决过程中的重传请求建议了一个时隙分配算法, 但是他们没有考虑退避窗口大小对冲突解决的影响。因

2005-01-17 收到, 2005-11-22 改回
国家自然科学基金(902040003, 60402012), 高等学校博士学科点专项科研基金(20010013003), 中国博士后科学基金(2003034111)和国家 973 计划(2003CB314806)资助课题

此,上面提到的这些退避机制都不太适合于解决 802.16 系统中的冲突问题。本文提出了一种快速有效的冲突解决方法——动态退避控制(Dynamical Backoff Control, DBC)算法。在该算法中,基站(Base Station, BS)动态地控制SS的退避范围,以减少SS的重传次数,增大每帧中成功发送的带宽请求报文数,减小时隙浪费,从而提高MAC层的数据吞吐,降低SS接入延迟。仿真结果显示本文提出的DBC算法对系统性能有一定的改善和提高。

本文的后续部分组织如下:第2节介绍 IEEE 802.16 的 MAC 层协议及相关的退避过程;第3节详细介绍本文提出的 DBC 算法;第4节详细描述对该算法的仿真以及和二进指数退避的性能比较;最后是结束语。

2 IEEE 802.16 MAC 协议介绍

在IEEE 802.16 协议中,上行信道被模型化为一连串的微时隙(minislot)流,这些时隙被所有的SS共享。上行信道的这种共享特性要求BS对这些时隙进行集中分配,以保证SS的有效接入。对上行nrTPS和BE业务类型的数据的发送,802.16 采用竞争与预留的方式。当SS有数据要发送时,它首先利用竞争时隙竞争发送带宽请求报文给BS,在BS成功收到带宽请求报文以后,再给该SS分配相应的数据时隙来传输数据。图1显示的是IEEE 802.16 中上行TDMA的帧结构。由于带宽请求报文是竞争发送的,因此在上行信道的竞争时隙中存在着冲突的现象。当冲突发生时,SS是不能直接检测到冲突的,而受冲突的带宽请求报文也会被BS丢弃。SS只有通过下一个收到的Uplink Map(UL-MAP)消息去判断自己发送的带宽请求报文是否受到冲突。如果请求报文没受到冲突,BS会在UL-MAP消息中对其进行确认;如果请求报文发生了冲突,由于BS没有成功接收该请求报文,因此不会在UL-MAP消息中描述任何与之相关的信息。SS在UL-MAP消息中得不到确认,从而进入冲突解决过程。对于在SS的冲突解决过程中采取的二进制指数退避算法,BS在上行信道描述符(Uplink Channel Descriptor, UCD)消息中指定SS的起始退避窗口值(Request Backoff Start, RBS)和最大退避窗口值(Request Backoff End, RBE),受冲突的SS首先从起始退避窗口中随机选择一个值进行退避。例如,当BS在UCD消息中指定Request Backoff Start=4, Request Backoff End=10 时,说明起始退避窗口大小为 2^4 , 最大退避窗口值为 2^{10} ; 在SS进入冲突解决过程时,它首先在起始退避窗口范围(0~15)内随机选择一个值(例如 5)作为退避值,这表明它需要退避 5 个传输机会以后才能在第 6 个传输机会进行带宽请求报文的发送。在协议中,传输机会被定义成 n 个 minislots, n 的大小由请求报文的大小所决定。当SS再次发生冲突时,它会把原来的退避窗口扩大 1 倍,然后再从中随机选择一个值进行退避;当窗口扩大到最大退避窗口值时,SS就一直保持该窗口的大小,并从中随机选择一个值进行退避,直到完成这次

冲突解决过程以后,SS才能发送新的带宽请求报文。若SS在该次冲突解决过程中的退避次数超过 16 次,则该带宽请求报文必须被丢弃。

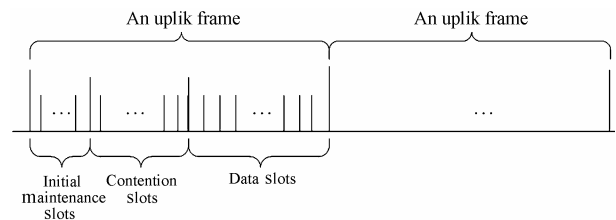


图 1 上行帧结构
Fig.1 The uplink frame structure

3 DBC 算法介绍

对于二进制指数退避算法,尽管其算法本身及实现比较简单,但存在一些问题。(1)起始退避窗口值和最大退避窗口值都是 2 的指数形式,不能准确描述 SS 的退避范围,会造成部分传输机会的浪费;(2)对于这两个值的选取,需要通过长期观察才能获得不同负载时的相应经验值;(3)二进制指数退避算法中的 SS 退避是分布式的,缺乏集中统一的控制;(4)不合适的起始退避窗口和最大退避窗口值会严重影响系统性能。

为了解决上述问题,本文提出了 DBC 算法。算法的基本思想是 BS 动态地控制 SS 的退避范围大小,以加速其冲突解决过程。BS 根据统计到的当前上行帧中受冲突的带宽请求报文数目,估计下一帧开始时发送这些受冲突的请求报文的 SS 需要在多大的范围内(亦即在退避窗口范围内)进行随机退避才能保证在这个退避范围内有较多的请求成功发送,同时又不浪费过多的传输机会。DBC 算法的详细描述如下。

首先,假设一个微时隙(minislot)恰好能传输一个带宽请求报文,亦即是一个微时隙代表一个传输机会;假设BS检测到第 i 个上行帧中有 n_c 个带宽请求报文受冲突,根据协议规定,这些受冲突的SS将根据退避范围大小在第 $i+1$ 个上行帧中开始进行随机退避。为了保证在这个退避窗口范围内的微时隙有较高的利用率(亦即吞吐),减少带宽请求报文的发送延迟,需要求出一个合适的退避范围大小值 w 。这里先定义几个符号: n_c 为一个上行帧中受冲突的带宽请求报文数; w 为退避范围的大小,亦即 w 个传输机会或竞争时隙; k 为表示恰好有 k 个带宽请求被成功发送,亦即恰好有 k 个竞争时隙或传输机会被成功利用; η 为退避范围大小 w 内的传输机会(或竞争时隙)的利用率。

下面我们先求出 w 个传输机会中恰好有 k 个传输机会或竞争时隙成功发送 k 个带宽请求报文的概率 P_k , 根据文献[10] 有

$$P_k = (-1)^k \frac{w! n_c!}{k! w^{n_c}} \sum_{j=k}^{\min(w, n_c)} (-1)^j \frac{(w-j)^{n_c-j}}{(j-k)!(w-j)!(n_c-j)!} \quad (1)$$

则退避范围内的这 w 个传输机会被成功利用的数学期望为

$$E[S] = \sum_{k=0}^w P_k \cdot k = \sum_{k=0}^w \left(k \cdot (-1)^k \frac{w! n_c!}{k! w^{n_c}} \cdot \sum_{j=k}^{\min(w, n_c)} (-1)^j \frac{(w-j)^{n_c-j}}{(j-k)!(w-j)!(n_c-j)!} \right) \quad (2)$$

当 w 较大时, 根据文献[11], 式(2)可以简化为

$$E[S] = \sum_{k=0}^w P_k \cdot k \cong \sum_{k=0}^w \left(k \cdot \frac{(n_c e^{-n_c/w})^k}{k!} \cdot \exp(-n_c e^{-n_c/w}) \right) \quad (3)$$

从而可以得出退避范围内的 w 个传输机会的利用率为

$$\eta = \frac{1}{w} \cdot E[S] = \frac{1}{w} \cdot \sum_{k=0}^w \left(k \cdot \frac{(n_c e^{-n_c/w})^k}{k!} \cdot \exp(-n_c e^{-n_c/w}) \right) \quad (4)$$

同样由于 w 较大, 式(4)可以简化为

$$\eta = \frac{1}{w} \cdot E[S] \cong \frac{1}{w} \cdot \exp(-n_c e^{-n_c/w}) \cdot n_c e^{-n_c/w} \cdot \exp(n_c e^{-n_c/w}) = \frac{n_c e^{-n_c/w}}{w} \quad (5)$$

为了使利用率 η 达到最大, 对式(5)中的 w 进行求导, 并令导数 $\frac{d\eta}{dw} = 0$, 从而可以得到

$$\frac{n_c^2 e^{-n_c/w} - n_c e^{-n_c/w} w}{w^3} = 0 \quad (6)$$

$$\Rightarrow w = n_c \quad (7)$$

另外, 在 w 不太大时, 退避范围内的 w 个传输机会的利用率为

$$\eta = \frac{1}{w} \sum_{k=0}^w \left(k \cdot (-1)^k \frac{w! n_c!}{k! w^{n_c}} \sum_{j=k}^{\min(w, n_c)} (-1)^j \frac{(w-j)^{n_c-j}}{(j-k)!(w-j)!(n_c-j)!} \right) \quad (8)$$

利用 Matlab 在不同 n_c 情况下对上式的 η 进行了计算, 得出当 $w \approx n_c$ 时, η 有最大值。从而, 可以得出当退避窗口范围大小等于受冲突的请求报文数, 并且受冲突的请求报文在这退避范围内重传时, 退避范围内的这些传输机会最大的吞吐或利用率。由于在实际系统中, BS 只能统计到每帧中受冲突的时隙数目, 根据文献[10]中提到一个冲突时隙最大可能(概率为 0.88)是由两个竞争请求报文造成的, 从而可以估算出每帧中受冲突的请求报文数 n_c 。

上面的推导是在没有考虑新到达的带宽请求报文对重传影响的情况下进行的, 现在把新到的请求报文与重传的请求报文一起考虑。只需对新到请求报文的发送稍做修改, 让其发送规则与重传的请求报文一样, 新到的请求报文也是在指定的退避窗口中随机选择一个值进行退避发送。这时, 只需估算每帧中会有多少个新到的请求报文和重传的请求报文就可以了, 亦即是让退避范围大小满足 $w = \hat{n}$, 其中 \hat{n} 为

预测的下一帧中会出现的请求报文数(包括新到和重传的)。对于 \hat{n} 的预测, 考虑到实现的复杂度, 同时相邻两帧中负载变化不大, 因此本算法利用当前帧中统计到的请求报文数来预测下一帧中可能会出现请求报文数。图 2 描述了 DBC 算法对整个退避范围的确定过程。在图 2 中, 每个时间间隔表示每个上行 TDMA 帧中的竞争时隙块, 把这些竞争时隙块从上行帧中提取出来组合成一个连续时间轴上的抽象描述, 以便于对 DBC 算法的解释。图 2 中 S_i 表示第 i 帧中 BS 分配的竞争传输机会数目, \hat{n}_i 表示第 i 帧中预计会出现的竞争请求数目(包括新到的和重传的), $w_i = \text{End}_i - \text{Start}_i$ 表示第 i 帧中出现的竞争请求报文的随机退避范围大小(亦即第 i 帧时各 SS 的退避范围大小), $[\text{Start}_i, \text{End}_i]$ 表示该随机退避的范围, Start_i 表示这个随机退避范围的下界, 是相对于第 i 帧的第 1 个传输机会的偏移量, End_i 表示这个随机退避范围的上界, 也是相对于第 i 帧的第 1 个传输机会的偏移量。如果 End_{i-1} 小于 S_{i-1} , 表示第 $i-1$ 帧中的退避范围在该帧中就结束了, 这时 $\text{Start}_i = 0$, 表示在第 i 帧时 SS 的随机退避范围从第 i 帧的第 1 个传输机会算起; 否则, $\text{Start}_i = \text{End}_{i-1} - S_{i-1}$ 。最后, $\text{End}_i = \text{Start}_i + w_i$ 。

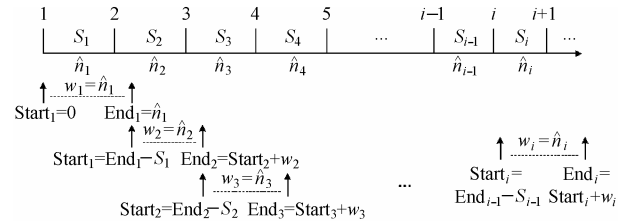


图 2 退避范围的计算过程

Fig.2 Computation process of backoff ranges

考虑到当 $\text{End}_i < S_i$ 时, 第 i 帧中会有 $S_i - \text{End}_i$ 个传输机会被浪费, 本文对上面的 DBC 算法进行了改进, 以扩大这时的退避范围。这将有助于减少该帧中的碰撞, 同时也能提高该帧中成功发送带宽请求报文的数量。修改后的 DBC 算法描述如表 1

表 1 DBC 算法
Tab.1 DBC algorithm

1	if($\text{End}_{i-1} \leq S_{i-1}$)	13	{
2	{	14	$\text{End}_i = \text{Start}_i + \text{remain_opportunity};$
3	$\text{Start}_i = 0;$	15	}
4	}	16	else
5	else	17	{
6	{	18	$\text{End}_i = \text{Start}_i + \hat{n}_i;$
7	$\text{Start}_i = \text{End}_{i-1} - S_{i-1};$	19	}
8	}	20	}
9	if($\text{Start}_i \leq S_i$)	21	else
10	{	22	{
11	$\text{remain_opportunity} = S_i - \text{Start}_i;$	23	$\text{End}_i = \text{Start}_i + \hat{n}_i;$
12	if($\text{remain_opportunity} \geq \hat{n}_i$)	24	}

为了实现上述算法, 并且与协议中规定的二进制指数退

避算法兼容,本文对 IEEE 802.16 协议中的 UCD 和 UL-MAP 消息内容做了一点改动。首先,利用 UCD 管理报文中 Request Backoff Start 和 Request Backoff End 两个 8bit 长度的字段中最高 bit 位来区分是采用二进制指数退避算法还是采用 DBC 算法。当最高比特位为“0”时,表示采用二进制指数退避算法;当为“1”时,表示采用 DBC 算法。由于描述每个上行帧中时隙的使用规则的 UL-MAP 消息的发送周期是以上行帧长为单位的,因此我们利用其描述信息单元 (Information Element, IE)的保留字段来传递每帧中 SS 的退避范围。其中 4bit 的上行间隔使用代码(Uplink Interval Usage Code, UIUC)等于 0 表示描述该 IE 的连接标识符(Connection Identifier, CID)和 Offset 字段是保留的, 12bit 的 Offset 字段来传递退避范围的下界, 16bit 的 CID 字段来传递退避范围的上界。图 3 是修改后的 UL-MAP 消息格式。这样,可描述的范围至少可达 4 千个 slot,这在实际环境中显然已经足够了。另外,由于 SS 的数量有限,且协议规定 SS 发送请求报文以后在没收到确认时是不能继续发送请求消息的,因此计算退避范围时,其上界和下界不会无限增大,它们只会随激活的 SS 的数量在某一范围内波动。

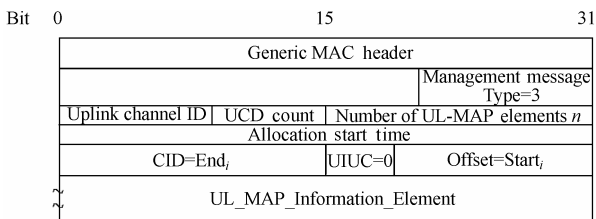


图 3 修改后的 UL-MAP 消息格式
Fig.3 Modified UL-MAP message format

4 仿真及讨论

由于 IEEE 802.16 的 MAC 层协议主要来自于 DOCSIS 协议^[12],因此本文对 OPNET 和 Cablelabs 联合开发的 DOCSIS 仿真模块进行了修改。在仿真中,一个带宽请求报文的长度定义为 128byte;为了使一个 minislot 恰好代表一个传输机会,正好能传输一个带宽请求报文,一个 minislot 所描述的时间长度被定义为 200μs。上行 TDMA 帧长为 0.01s 和 0.02s 时的两种场景被分别仿真。在统计受冲突的请求报文数时,我们让每个受冲突的时隙以概率 0.88 乘以 2 或者以概率 0.12 乘以 3,这样就可统计到每帧中受冲突的请求报文数。业务源为语音和数据流。各 SS 到 BS 的距离都完全相等,仿真时长为 120s,二进制指数退避算法中的最大退避窗口值都设为 10,亦即 2¹⁰。仿真实验对二进制指数退避算法一共比较了 5 种不同起始退避窗口值的性能。

在仿真中,主要比较本文提出的 DBC 算法与二进制指数退避算法对整个网络性能影响的差异。首先,图 4 显示的是上行帧长为 0.01s 时,DBC 算法与不同起始退避窗口下的二进制指数退避算法的 MAC 层数据吞吐的比较。从图 4 中

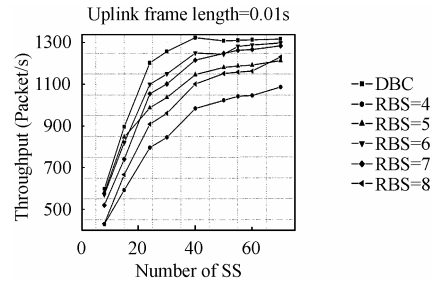


图 4 MAC 层数据吞吐
Fig.4 Throughput of MAC

可以看到,使用 DBC 算法时可以获得比二进制算法更好的吞吐性能。在 SS 数量较少,亦即是负载较轻的时候,DBC 算法与起始退避为 5 的二进制指数退避算法的 MAC 层数据吞吐相近,DBC 的略为高一点;同时,起始退避为 5 的数据吞吐高于其他起始退避窗口值的数据吞吐。随着 SS 数量的增加,起始退避窗口值为 6 的二进制指数退避算法下的数据吞吐逐渐高于其他 4 种起始退避窗口值的数据吞吐,而 DBC 算法始终使 MAC 层数据吞吐保持在最大值。当 SS 数量大于 50,亦即负载较重时,DBC 算法的数据吞吐仍能稳定在一个较高值附近。这时,DBC 算法与起始退避窗口为 6 的二进制指数退避算法的吞吐性能差距不大,因为此时它们都已接近系统上行的吞吐饱和。

图 5 显示的是上行帧长为 0.01s 时,DBC 算法和二进制指数退避算法下的 SS 平均接入延迟的比较。其中,SS 的平均接入延迟定义为 SS 发送一个带宽请求报文到成功接收 BS 对该请求报文的确认信息所花费的平均时间。平均接入延迟反映了 SS 冲突解决过程的快慢程度。该延迟主要受 SS 重传次数以及退避值大小的影响。从图 5 中可以看到,起始退避值为 8 时,由于退避范围较大,SS 发送碰撞的概率较小,因此其平均接入延迟也较小;但退避范围过大,造成大量传输机会浪费,从而导致起始退避值为 8 时的 MAC 层数据吞吐很低。起始退避窗口值为 4 时,由于退避范围较小,使得带宽请求报文碰撞的概率加大,从而增加了 SS 重传的次数,造成 SS 的平均接入延迟加大。而 DBC 算法中由于每帧的传输机会利用率大,带宽请求报文重传的次数和数量较少,使得该算法控制下的 SS 的平均接入延迟明显小于二进制指数退避算法控制下的平均接入延迟。从图 5 中还可以看到,当 SS 数目较大时,DBC 算法的平均接入延迟较二进制指数退避算法改进了 10%左右。

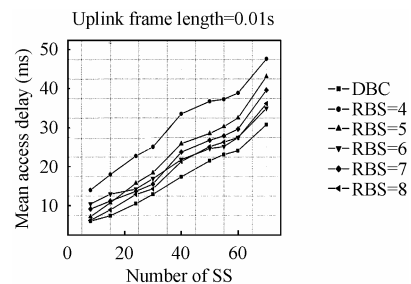


图 5 SSs 平均接入延迟
Fig.5 Mean access delay of SSs

图 6 和图 7 分别显示了上行帧长为 0.02s 时的系统 MAC 层数据吞吐性能和 SS 的平均接入延迟性能。从这两幅图中我们可以看到, DBC 算法在帧长为 0.02s 时仍能获得优于二进制指数退避算法的吞吐及延迟性能, 且其延迟性能改进最高达 30%左右(SS 数量为 40)。对于二进制指数退避算法而言, 起始退避窗口值为 7 和 8 时的系统性能好于其它几种起始退避窗口值的性能, 但在图 4 中我们看到起始退避窗口值为 6 时有更好的系统性能。这说明上行 TDMA 帧的长度对二进制指数退避算法的性能有较大影响。综合图 4—图 7, 可以发现 DBC 算法不受上行 TDMA 帧的长度以及网络中 SS 数量的影响, 始终能获得最好的吞吐性能和延迟性能。

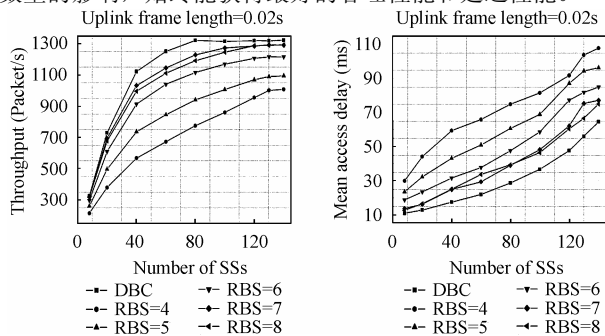


图 6 MAC 层数据吞吐
Fig.6 Throughput of MAC

图 7 SSs 平均接入延迟
Fig.7 Mean access delay of SSs

5 结束语

IEEE 802.16 的上行 MAC 层采用 TDMA 的接入方式, 同一扇区内的所有 SS 共享一个上行信道资源, 它们之间存在着碰撞与冲突。为了解决 SS 之间的冲突, 本文针对 SS 在碰撞以后的冲突解决过程提出了一种快速有效的 DBC 算法。在该算法中, BS 通过预测下一上行帧中将会出现的带宽请求报文数, 来动态、有效地控制 SS 的退避范围, 以提高 MAC 层数据吞吐和降低 SS 的平均接入延迟。我们利用 OPNET, 通过修改其中的 DOCSIS 模块对两种算法进行了仿真比较。通过仿真结果发现二进制指数退避算法的不同起始退避窗口值对系统性能有较大影响, 同一起始退避窗口值在不同负载(或 SS 数量)时不能使系统性能一直保持最优; 同时, 二进制指数退避算法性能也受上行 TDMA 帧长的影响。对于 DBC 算法, 仿真结果验证了其不受负载(或 SS 数量)及上行帧长的影响, 系统性能优于二进制指数退避算法, 并始终保持在最好状态。

参考文献

- [1] <http://www.wi-fi.org>
- [2] IEEE P802.16-REVd/D5-2004: Draft IEEE Standard for Local and Metropolitan Area Networks—Part 16: Air Interface for Fixed Broadband Wireless Access Systems. May 13, 2004.
- [3] IEEE Std 802.16e/D2-2004, Draft Amendment to IEEE Standard for Local and Metropolitan Area Networks—Part 16: Air Interface for Fixed Broadband Wireless Access Systems—Amendment for Physical and Medium Access Control Layer for Combined Fixed and Mobile Operation in Licensed Bands. Apr. 2, 2004.
- [4] <http://www.wimaxforum.org>
- [5] Eklund C, Marks R B, Stanwood K L, Wang S. IEEE Standard 802.16: A technical overview of the wireless MAN air interface for broadband wireless access. *IEEE Communications Magazine*, 2002, 6 (40): 98-107.
- [6] Bertsekas D, Gallager R. *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987: 35-36.
- [7] Kuo W K, Kumar S, Kuo C C J. Dynamic collision resolution and traffic scheduling for DOCSIS systems with QoS support. In *Proceedings of IEEE GLOBECOM'03*, San Francisco, California, Dec. 2003: 3894-3898.
- [8] SCTE DSS 00-05, *Data-Over-Cable Service Interface Specification (DOCSIS) SP-RF1v1.1-I05-000714*, Radio Frequency Interface 1.1 Specification. July 2000.
- [9] Yin W M, Lin Y D. Statistically optimized minislot allocation for initial and collision resolution in hybrid fiber coaxial networks. *IEEE Journal on Selected Areas in Communications*, 2000, 9 (18): 1764-1773.
- [10] Feller W. *Introduction to Probability Theory and Its Applications*. New York: Wiley, 1957: 91-103.
- [11] Raychaudhuri D. Announced retransmission random access. *IEEE Trans. on Communications*, 1985, COM-33(11): 1183-1190.
- [12] IEEE 802.16.1mc-00/01. *Media Access Control Protocol based on DOCSIS 1.1*. 1999.

肖峻峰: 男, 1978 年生, 博士生, 研究方向为宽带无线网络、IP 网络服务质量、网络测量等。
邹仕洪: 男, 1978 年生, 博士, 讲师, 主要研究方向为无线局域网、移动自组织网等。
程时端: 女, 1940 年生, 教授, 博士生导师, 主要研究方向为服务质量、网络性能分析等。