

# IEEE 802.16 Mesh 模式下基于 QoS 的时隙分配算法

是元吉, 刘富强, 钱业青, 汪丽萍

(同济大学电子与信息工程学院, 上海 201804)

**摘要:** IEEE 802.16-2004 标准定义了 Mesh 模式下的调度机制, 基于这些调度机制下的数据信道资源分配算法, 在标准中没有详细规定。该文提出了两种基于优先级的时隙分配算法, 以实现 MAC 层的 QoS, 通过仿真分析了平均分组延迟、请求失败率和时隙利用率。仿真结果表明这些算法实现了对不同需求的业务流的 QoS 保障。

**关键词:** Mesh; MAC; QoS; 自适应时隙预留算法

## Slot Allocation Algorithm Based on QoS for IEEE 802.16 in Mesh Mode

SHI Yuanji, LIU Fuqiang, QIAN Yeqing, WANG Liping

(School of Electronics and Information Engineering, Tongji University, Shanghai 201804)

**【Abstract】** The MAC protocol of IEEE 802.16-2004 standard specifies schedule mechanisms about mesh mode, while its channel resource allocation algorithms are not specified. This paper proposes two slot allocation algorithms based on priority to achieve QoS on MAC layer. And conducts a simulation to study average delay, request reject rate, and minislot utilization based on proposed algorithm. The simulation results show that the algorithm achieves QoS of different services.

**【Key words】** Mesh; MAC; QoS; Adaptive slot reserved algorithm

### 1 概述

近年来, 随着人们对于带宽需求的增加, 无线宽带接入技术已经成为人们关注的热点问题。IEEE 802 委员会于 1999 年成立了 802.16 工作组来专门制定宽带无线接入标准, 以解决最后一公里的无线宽带接入。2004 年颁布的 IEEE 802.16-2004 标准<sup>[1]</sup>(又称 802.16d 标准)支持 2-11GHz 非视距(NLOS)传输和 10-66 GHz 视距(LOS)传输, 其媒质接入层(MAC)层支持 PMP 和 Mesh 两种模式。其中 Mesh 模式具有建网成本低、频谱利用率高、系统容量大、覆盖率高、可扩展性强、可靠性高以及在多跳环境下用户吞吐量大等优点<sup>[2]</sup>, 因此具有广泛的应用前景。

MAC 层的核心就是通过调度解决相互竞争的多个用户之间信道资源的分配问题。在 IEEE 802.16-2004 标准的 Mesh 模式下, MAC 层的调度机制可以分为集中式调度和分布式调度 2 种类型。在集中式调度中, 由基站统一调度信道资源的分配。在分布式调度中, 无须基站的许可, 用户站间可直接基于分布式调度算法进行通信。分布式调度又可分为协同和非协同方式。“协同”是指在控制子帧中使用无碰撞方式来调度传输包;“非协同”是指部分的基于竞争的方式来传输调度包。二者的主要区别在于, 协同情况下调度是完全无碰撞的, 而非协同方式可能发生碰撞。

目前, 很多文献研究了在 PMP 模式下支持 QoS 的调度算法, 而对于协议中没有规定的基于 Mesh 模式下调度机制的信道资源分配算法研究得很少。Zeng Zhihui<sup>[3]</sup>提出了一种可以实现 MAC 层 QoS 的基于优先级的分组调度算法。本文介绍了 IEEE 802.16-2004 Mesh 模式下协同分布式调度消息和三次握手的过程。提出了在 Mesh 模式协同分布式调度机制下的基于优先级的时隙分配算法。通过仿真结果分析了算法的性能,

并与目前已有的算法进行了比较。

### 2 Mesh 模式下协同分布式调度消息及 3 次握手过程

IEEE 802.16 MAC 层是基于“连接”的, 每一个“连接”均由一个标识符(CID)进行唯一标识。在 IEEE 802.16 标准中, MAC 层在 PMP 模式下定义了较为完整的 QoS 机制。MAC 层针对每个连接可以分别设置不同的 QoS 参数, 包括速率、延时等指标。为了更好地控制上行数据的带宽分配, 标准还定义了 4 种不同的业务, 即主动授权业务(UGS)、实时轮询业务(rtPS)、非实时轮询业务(nrtPS)、尽力而为业务(BE)。而在 Mesh 模式下, 却没有定义这 4 种业务。但是 Mesh 模式下的 CID 结构中包括了 3 项服务参数 Reliability、Priority/Class 和 Drop Precedence。Reliability 表示数据包是否重发; Priority/Class 表示数据包的优先级; Drop Precedence 表示当网络拥塞时, 数据包丢失的可能性。这为我们在研究实现 Mesh 模式下的 QoS 提供了协议保证。

IEEE 802.16 MAC CPS 子层控制多用户共享的空中资源, 其 Mesh 模式只支持时分双工(TDD)方式。其 MAC CPS 帧结构包括控制子帧和数据子帧。Mesh 模式下协同分布式调度时, 所有的节点通过控制子帧中规定的 slot 时隙传输 MSH-DSCH 消息通知所有的邻居节点它的传输调度信息。MSH-DSCH 不仅可以用来传送该节点的资源请求信息以及对于邻居节点的授权信息, 还可以传送该节点向邻居节点授

**基金项目:** 科技部国际合作基金资助重点项目(2005DFA10100); 国家“973”计划基金资助项目(2004CB719802)

**作者简介:** 是元吉(1982 -), 男, 硕士生, 主研方向: 无线城域网宽带接入; 刘富强, 教授、博导; 钱业青, 讲师、博士; 汪丽萍, 硕士生

**收稿日期:** 2006-07-04

**E-mail:** feiguohai1982127@163.com

权的空闲资源的信息（这里的资源指的是 Mesh 帧结构里的 256 个传输数据的 minislot，minislot 是一个时间单元）。

## 2.1 MSH-DSCH 消息结构

MSH-DSCH 消息在 Mesh 模式下的协同分布式调度过程中起着重要的作用。MSH-DSCH 消息由下面几个部分构成：

(1)Scheduling IE：存放本节点和邻居节点的下次发送 MSH-DSCH 的时间和 Holdoff Exponent 参数。

(2)Request IE：存放本节点的资源请求信息。包括 Link ID、Demand Level、Demand Persistence，其中 Demand Level 指请求的 minislot 总数，Demand Persistence 指请求的所有 minislot 连续分布的帧数。

(3)Availability IE：存放本节点当前的可利用信道资源。包括 Start Frame number（起始帧号）、Minislot start（起始位置）、Minislot range（一帧中 minislot 范围，即从起始位置开始 minislot 的数目）、Direction（0-不可利用，1-可被用于发送，2-可被用于接收，3-可被用于发送或接收）、Persistence（经历的连续帧数）、Channel（逻辑信道号）。

(4)Grants IE：存放本节点的资源授权信息或确认授权信息。包括 Link ID、Start Frame number、Minislot start、Minislot range、Direction、Persistence、Channel。参数的意义同上，其中 Direction=0 表示确认授权，Direction=1 表示授权。

## 2.2 三次握手过程

IEEE 802.16-2004 标准的 Mesh 模式下协同分布式调度时，两个节点通过“请求-授权-授权确认”的三次握手过程来建立发送数据前的连接，其过程如下：

(1)请求节点广播 MSH-DSCH: Request IE 请求信息，同时包含 MSH-DSCH: Availability IE 信息指明其可用的 minislot。

(2)授权节点通过本文算法检查在 Availability 范围内 Request 能否被接受，如果可以就广播 MSH-DSCH: Grants IE 授权信息，所有授权节点的邻居节点都必须认同这个 Grant 授权，以免发生冲突。

(3)请求节点复制接收到的 Grant 授权，并广播 MSH-DSCH: Grants IE 确认信息，所有请求节点的邻居节点都必须认同这个 Grant 确认，以免发生冲突。

这样就完成了三次握手过程。请求节点可以在三次握手过程协商的 minislot 范围里与授权节点进行数据通信。

## 3 基于优先级的时隙分配算法

在 FDMA 中经常采用一种信道资源预留的思想。本文引用了这种思想，并且提出了两类算法：预留时隙固定的时隙分配算法和预留时隙自适应改变的时隙分配算法。

### 3.1 预留时隙固定的时隙分配算法

本文设计的第 1 种时隙分配算法，称为预留时隙固定的时隙分配算法，记为 A1。该算法主要步骤如下：

(1)根据 Request IE 中的 Demand Level 和 Demand Persistence 计算每帧中占用的 minislot 个数，即  $Minislot\ range = Demand\ Level / Demand\ Persistence$ （上取整）。

(2)从邻居表中查询请求节点下一次 MSH-DSCH 消息的时间（Next Xmt Time），令  $Start = Next\ Xmt\ Time$  对应的帧。

(3)在数据子帧中设置一个固定点(Pt)，预留第 Pt+1 个 minislot 到第 256 个 minislot 分配给高优先级的分组调度请求。

(4)若分组调度请求是高优先级，则从 Start 帧开始，在每帧 256 个 minislot 范围内，查找连续 Demand Persistence 帧中相同位置开始的 Minislot range 个空闲的 minislot。如果查找成功，则生成相应的 Grants IE 并返回请求节点。如果查找失败，则返回失败信息。

(5)若分组调度请求是低优先级，则从 Start 帧开始，在每帧前 Pt 个 minislot 范围内，查找连续 Demand Persistence 帧中相同位置开始的 Minislot range 个空闲 minislot。如果查找成功，则生成相应的 Grant IE 并返回请求节点。如果查找失败，则返回失败信息。

### 3.2 预留时隙自适应改变的时隙分配算法

本文设计的第 2 种时隙分配算法，称为预留时隙自适应

改变的时隙分配算法，记为 A2。该算法主要分为 5 个步骤，步骤 1~步骤 5 和上述相同，其中步骤 3 改进为：在数据子帧中设置一个固定点(Pt)，预留第 Pt+1 个 minislot 到第 256 个 minislot 分配给高优先级的分组调度请求。设置检查点(CP)、阈值(TH)和步长(St)来根据网络的信道资源利用情况，动态改变 Pt 值。CP 指向当前帧后第 CP 帧，当 CP 指向帧占用的 minislot 数目(use)大于 TH 时，网络拥挤，根据公式

$$Pt = Pt - (\text{int})\left(\frac{\text{use}}{\text{TH}} \times \text{St}\right) \quad (1)$$

动态改变 Pt 值，当  $Pt < 128$  时， $Pt = 128$ ；当 CP 指向帧占用的 minislot 数目(use)小于 TH 时，网络正常，根据公式

$$Pt = Pt + (\text{int})\left(\frac{256 - \text{use}}{\text{TH}} \times \text{St}\right) \quad (2)$$

动态改变 Pt 值，当  $Pt > 188$  时， $Pt = 188$ 。

## 4 时隙分配算法在 NS 中的仿真分析

在 NS-2 中，仿真了上述两种时隙分配算法，并对仿真结果进行了分析。另外与 Zeng Zhihui<sup>[3]</sup>的算法（称之为 A0）进行了比较。为了与 A0 算法进行比较，仿真中参数设置如表 1 所示。仿真中使用 3 种流，分别是数据流、实时 CBR 流和实时 VBR 流。数据分组的到达率符合参数为的 Poisson 分布，取 30。分组大小为 825B，8 帧传完。固定数据节点数目为 15。CBR 流以恒定的 30ms 间隔时间产生。分组大小为 240B，2 帧传完。VBR 流采用符合指数分布的 ON/OFF 模型。分组大小为 240B，2 帧传完。在仿真中设定数据分组为低优先级(CID 中的 Priority=0)，实时 CBR 分组和实时 VBR 分组都是高优先级(CID 中的 Priority=1)。仿真时间设置为 10s。算法 A0 中参数设置：CP1=10；CP2=20；TH=150。算法 A1 中参数设置：Pt=158。算法 A2 中参数设置：Pt=158；St=10；CP=5；TH=158。

表 1 仿真网络参数配置

参数	值
Channel rate	50 Mbps
Frame time	1 ms
Holdoff Exp	0
Slot time	6 250 ns
Minislot time	3 515 625 ps

在图 1 中，在使用每种不同算法时，随着实时 VBR 节点数目的增加，实时 VBR 分组的平均分组延迟和数据分组的平均分组延迟的差值越来越大。体现了区分服务，高优先级业务流得到低延迟服务。

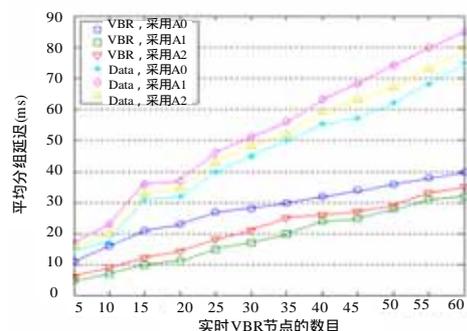


图 1 平均分组延迟与实时 VBR 节点的数目之间的关系

在图 2 中，随着实时 VBR 节点数目的增加，数据分组的请求失败率增大。当实时 VBR 节点数目达到 35 时，A2 算法数据分组的请求失败率比 A0 算法的数据分组的请求失败率小。实时 VBR 分组的请求失败率均为 0。

（下转第 113 页）