宽带无线城域网 MAC 层
CS 分类器和 ARQ 机制的研究与实现

摘 要

嵌入式设备和无线通信技术的进步大大推动了无线网络的发展。WiMAX 技术是当前热点的无线网络技术，它是一种宽带无线接入技术，用以替代传统的有线连接方式，解决“最后一公里”接入问题。本论文的课题来源于当前最热门的 WiMAX 研究项目“宽带无线城域网络多媒体实时通信系统”，该项目的研发目标是根据 IEEE 802.16 标准进行 WiMAX 系统硬件平台设计和软件系统开发，实现 WiMAX 系统，并实现 WiMAX 系统的多媒体实时通信。工作重点是 WiMAX 协议软件 MAC 层的设计框架和模块划分；CS 子层的设计与实现以及 ARQ 功能的研究与实现。

本文结合项目开发过程，对 WiMAX 基站整体架构设计进行了论述，并针对工作重点及相关测试进行了分析和讨论。论文的主要研究工作和创新点体现在下面几个方面：

1. 根据协议，分析了 IEEE802.16 MAC 层中的关键机制。
2. 基于 picoChip 平台的 802.16 MAC 协议软件的实现。文中介绍了 picoChip 的 DVK102 嵌入式开发平台，给出了在上述平台上实现 IEEE802.16 MAC 层的总体设计框架，模块划分，模块接口定义。
3. 802.16 MAC CS 子层分类器的设计和实现。文中提出了 CS 子层分类器的一种实现方案。本文给出了该方案的设计框架，模块划分，分类算法，实现方式。该方案对宽带无线网络的实现工作有着非常重要的作用。
4. 802.16 MAC 软件的 ARQ 功能的设计和实现。文中介绍了一种为宽带无线接入网络实现 ARQ 功能的设计方案，给出了该方案的算法和实现方式。该设计方案对无线城域网络的实现工作有着重要的作用。
5. 802.16 MAC 分类器的测试方案和测试结果。文中给出了测试的相关流程，测试结果，并对测试结果进行了分析。

关键词：无线城域网　ARQ　CS 分类器　基站

DESIGN AND IMPLEMENTATION OF CS CLASSIFIER AND ARQ MECHANISM IN MAC LAYER IN WMAN

ABSTRACT

The achievements on mobile devices and wireless communication technology lead to the boom of wireless networks. WiMAX (IEEE 802.16 standard) is the hottest wireless networking technologies, which is expected to solve the last-mile problem instead of classic wired connections. The topic of this thesis is based on the project "Multimedia Real-time System of Broad Band WMAN ". The final aim of this project is to implement WiMAX system and transmit the audio and video on this system. My work in the project is focus on part of protocol software design and implements.

In this thesis, an overall analysis of WiMAX technology and WiMAX architecture design is given first, and following is the main part of the thesis, detailed description of my work, based on the design and implementation of the whole project. The main achievements of this thesis are listed as following:

1. Parse the key mechanisms of the IEEE802.16 MAC.
2. The implementation of IEEE 802.16 MAC protocols on picoChip development platform. We introduce the DVK102 embedded platform. The architecture, module partition, module interface and data structure to implement the IEEE 802.16 MAC on that platform are given in the thesis.
3. Design and implementation of CS sub-layer classifier in MAC layer of IEEE 802.16 networks. We introduce a method of CS sub-layer classifier. We introduce the architecture, module partition, and implementation. The proposed classifier method is critical for the future

research work on IEEE 802.16 wireless access systems.

4. Design and implementation of ARQ function in MAC layer of IEEE802.16 networks. We introduce a method of ARQ mechanism for the 802.16 network system. We introduce the design idea and implement mode. The proposed ARQ method has been applied in the IEEE 802.16 broadband wireless access systems.
5. Test and integration of 802.16 CS sub-layer; part of test results and related analysis are also provided.

KEY WORDS: Wireless MAN ARQ CS classifier Base station

英文缩略语索引

WiMAX	Worldwide Interoperability for Microwave Access，微波存取全球互通
MAC	Medium Access Control Layer，媒体接入控制层
PHY	Physical Layer，物理层
BS	Base Station，基站
SS	Subscriber Station，用户站
CS	Convergence Sublayer，会聚子层
CPS	Common Part Sublayer，公共部分子层
QoS	Quality of Service，服务质量
NLOS	Non-line-of-sight，非视距
UGS	Unsolicited Grant Service，主动授予业务
rtPS	Real-time Polling Service，实时轮询业务
nrtPS	Non-real-time Polling Service，非实时轮询业务
BE	Best Effort，尽力而为业务
AP	Access Point，接入点
SDU	Service Data Unit，服务数据单元
PDU	Protocol Data Unit，协议数据单元
ARQ	Automatic Repeat Request，自动重发请求
BWA	Broadband Wireless Access，宽带无线接入
SAP	ServiceAccessPoint，业务接入点
PMP	Point to Multipoint，点到多点
RS	RepeatStation，中继站

声 明

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：蔡彩云 日期：2007.3.31

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在__年解密后适用本授权书。非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名：蔡彩云 日期：2007.3.31

导师签名：崔晓春 日期：2007.4.2

第一章 课题简介

1.1 课题背景介绍

价格便宜的便携式计算机、移动电话和手持式设备的日趋流行，以及 Internet 应用和电子商务的快速发展促进了无线网络的发展。针对不同的网络通信环境产生了多种无线网络通信技术，包括 IEEE 802.11 无线局域网标准(WLAN)，IEEE 802.16 无线城域网标准（WMAN）等。目前 IEEE 802.11 无线局域网和 IEEE 802.16 无线城域网技术是发展势头非常强劲的无线网络技术。无线城域网技术是继无线局域网之后，覆盖范围更广、传输速率更高的新一代宽带无线接入技术。

本课题来源于项目“宽带无线城域网多媒体实时通信系统”，课题主要内容是进行 WiMAX 系统的硬件平台设计和软件系统开发，最终实现高容量、多用户、性能稳定的 WiMAX 系统；并在该系统上，实现实时多媒体的传输。该系统开发遵循 IEEE 802.16 标准；开发主要内容是基于该标准的 MAC 层协议软件和信令协议软件开发以及嵌入式操作系统板级支持包（BSP）和设备驱动程序开发；硬件平台和基于标准的物理层开发；软硬件开发完成后进行系统集成和各项测试。

目前已经完成第一代的设计开发和测试，达到预期目标。

1.2 本文论述的主要内容

本文主要是对项目“宽带无线城域网络多媒体实时通信系统”中的关键技术进行论述，主要内容是：

1) 对 IEEE 802.16 系列标准进行研究，分析基于该标准的 WiMAX 体系结构；
2) 分析基于该标准的 WiMAX 系统中的关键机制；
3) 根据客户功能需求，结合 IEEE 802.16 协议，完成 MAC 层整体框架的设计，模块的划分；
4) 学习使用 VxWorks 操作系统及其开发工具 Tornado；
5) 学习 VxWorks 的 MUX 层协议，基于该协议栈进行驱动程序和网络协议栈接口的设计与编码；
6) 完成 CS 子层（分类器）的设计与实现，包括整体设计，模块划分，各模块的详细设计和编码工作，并在 Tornado 开发环境下完成代码仿真和调试；

7) 完成 MAC 层中 ARQ 模块的设计和编码工作，在 Tornado 上完成仿真与调试；

8) 参与项目后期的子模块测试及系统测试工作。

限于篇幅，本文只给出了作者设计的系统的总体框架和作者负责设计和开发的子模块，对于其他模块则没有涉及。

很高兴在研究生期间有机会参与WiMAX的项目，在老师的精心指导、同学的热心帮助和共同努力下，顺利完成了科研任务。在完成整个项目的过程中，不仅深入研究了WiMAX技术、IEEE 802.16标准及无线链路资源管理机制，对宽带无线接入系统设计有了深刻的理解；对嵌入式操作系统、通信协议软件开发有了深刻的理解，更重要的是深刻体会到项目管理，有效沟通和团队精神对项目成败的至关重要作用。这些收获都将使作者终身受益！

第二章 概述

2.1 WiMAX 技术

WiMAX 的全名是微波存取全球互通(Worldwide Interoperability for Microwave Access)，将此技术与需要授权或免授权的微波设备相结合后，由于成本较低，将扩大宽带无线市场，改善企业与服务供应商的认知度。该技术以 IEEE 802.16 的系列宽频无线标准为基础。它可作为线缆和 DSL 的无线扩展技术，从而实现无线宽带接入，来提供最后一英里的无线宽带接入。WiMAX 将提供固定、移动、便携形式的无线宽带连接，并最终能够在不需要直接视距基站的情况下提供移动无线宽带连接。

作为致力于宽带无线接入(BWA)产品的互操作性认证的唯一专业组织，WiMAX 论坛将阐释技术标准，并进行相关的一致性和互操作性测试，确保不同供应商的系统能够实现无缝连接。那些通过一致性和互操作性测试的产品将获得"WiMAX 论坛认证(WiMAX Forum CertifiedTM)"标志。WiMAX 论坛是一个由业界领先的运营商、通信部件和设备公司共同成立的组织。WiMAX 论坛的宗旨是促进和认证符合 IEEE 802.16 和 ETSI HiperMAN 标准的宽带无线接入设备的兼容性和互操作性。由于标准本身不足以推动某种技术的大规模使用，而 WiMAX 论坛的建立则可以帮助消除大规模使用宽带无线接入(BWA)技术所面临的各种障碍[1]。

WiMAX 的技术优点：

1) 提供优良的最后一公里网络接入服务。作为一种无线城域网技术，它可以将 Wi－Fi 热点连接到互联网，也可作为 DSL 等有线接入方式的无线扩展，实现最后一公里的宽带接入。WiMAX 可为 50 公里线性区域内提供服务，用户无需线缆即可与基站建立宽带连接。
2) 实现更远的传输距离。WiMAX 所能实现的 50 公里的无线信号传输距离是无线局域网所不能比拟的，网络覆盖面积是 3G 发射塔的 10 倍，只要少数基站建设就能实现全城覆盖，大大扩展了无线网络应用的范围。
3) 提供更高速的宽带接入。WiMAX 所能提供的最高接入速度是 70M，是 3G 所能提供的宽带速度的 30 倍。

WiMAX 的市场优势：

1) WiMAX 为集成电路制造商提供了大规模量产的机会。

2) 基于一个标准的稳定平台来快速增加设备的新能力，使设备制造商能够更快地进行设备创新。

3) 为运营商提供一个能降低设备成本并加速性价比提升的公共平台，快速开通 T1/E1 级别的、按需分配大容量宽带业务。随着市场规模的扩大，设备将越来越便宜，从而降低与设备配置相关的资金风险。基站可与各厂家的终端(CPE)相兼容，不再出现被一家设备制造商垄断的现象。

4) 为用户提供更多的宽带接入选择，特别是在那些仍存在市场空间的地域，如接入困难的大城市市中心的建筑；用户距离局端很远的郊外区域；人口密度低，电信基础网络薄弱的农村[1]。

2.2 WiMAX 的标准体系

IEEE 802委员会于1999年成立了802.16工作组，专门开发宽带无线接入技术标准。802.16系列标准及主要内容如表2-1所示[1]。

2001年12月颁布的802.16标准，对使用10～66GHz频段的固定宽带无线接入系统的空中接口物理层和MAC层进行了规范，由于其使用的频段较高，因此仅能应用于视距范围内。

2003年1月颁布的802.16a标准对之前颁布的802.16标准进行了扩展，对使用2～11GHz许可和免许可频段的固定宽带无线接入系统的空中接口物理层和MAC层进行了规范，该频段具有非视距传输的特点，覆盖范围最远可达50km，通常小区半径为6～10km。另外，802.16a的MAC层提供了QoS保证机制，可支持语音和视频等实时性业务。这些特点使得802.16a标准与802.16标准相比更具有市场应用价值，真正成为适合应用于城域网的无线接入技术。

表2-1 802.16系列各标准负责的技术领域

标准号	负责的技术领域
802.16	10－66GHz 固定宽带无线接入系统空中接口标准
802.16a	2－11GHz 固定宽带无线接入系统空中接口标准
802.16c	10－66GHz 固定宽带无线接入系统关于兼容性的增补文件
802.16d	2－66GHz 固定无线接入系统空中接口标准
802.16e	2－66GHz 固定和移动宽带无线接入系统空中接口标准
802.16f	固定宽带无线接入系统空中接口 MIB 要求
802.16g	固定和移动宽带无线接入系统空中接口管理平面流程和服务要求

802.16d标准是802.16标准系列的一个修订版本，是相对比较成熟并且最具有实用性的一个标准版本，在2004年下半年正式发布。802.16d对2～66GHz频段的空中接口

物理层和MAC层进行了详细规定，定义了支持多种业务类型的固定宽带无线接入系统的MAC层和相对应的多个物理层。该标准对前几个802.16标准进行了整合和修订，但仍属于固定宽带无线接入规范。它保持了802.16、16a等标准中的所有模式和主要特点，并未增加新的模式；增加或修改的内容用来提高系统性能和简化部属，或者用来更正错误、不明确或不完整的描述，其中包括对部分系统信息的增补和修订。同时，为了能够后向平滑过渡到支持用户站以车辆速度移动的802.16e标准，802.16d增加了部分功能以支持用户的移动性。

802.16e标准是802.16标准的增强版本，于2006年1月发布。该标准规定了可同时支持固定和移动宽带无线接入的系统，工作在2～66GHz之间适宜于移动性的许可频段，可支持用户站以车辆速度移动，同时802.16a规定的固定无线接入用户能力并不因此受到影响；同时该标准也规定了支持基站或扇区间高层切换的功能。802.16e标准面向更长范围的无线点到多点城域网系统。该系统可提供核心公共网接入。

除了以上几个标准外，另外还有3个重要标准：

2002年正式发布的IEEE 802.16c，它是对IEEE 802.16的增补，是使用10～66 GHz频段IEEE 802.16系统的兼容性标准，它详细规定了工作于10～66GHz频段的IEEE 802.16系统在实现上的一系列特性和功能。

IEEE 802.16f，它定义了IEEE 802.16系统MAC层和物理层的管理信息库(MIB)以及相关的管理流程。

IEEE 802.16g，制订它的目的是为了规定标准的IEEE 802.16系统管理流程和接口，从而实现IEEE 802.16设备的互操作性和对网络资源、移动性和频谱的有效管理。IEEE 802.16f和IEEE 802.16g这两个标准是的目的是在网络管理层面形成新标准。

2.3 WiMAX 的应用

WiMAX 的技术特性和应用特点决定了其能适应各种的应用环境，具有不同的应用模式。

1. PMP 应用模式

如图 2-2 所示，PMP 应用模式以 BS 为核心，采用点到多点的连接方式，构建星形结构的 WiMAX 接入网络。BS 扮演业务接入点（SAP）的角色，通过动态带宽分配技术，BS 可以根据覆盖区用户的情况，灵活选用定向天线、全向天线以及多扇区技术满足大量的用户站（SS）设备接入核心网的需求。必要时，可以通过中继站（RS）扩大无线覆盖范围。还可以根据用户群数量的变化，灵活划分信道带宽，对网络扩容，

实现效益与成本的协调。

PMP 应用模式是一种常用的接入网应用形式，其特点在于网络结构简洁，应用模式与 xDSL 等线缆接入形式相似，因此，是一种线缆替代的理想选用方案[1]。

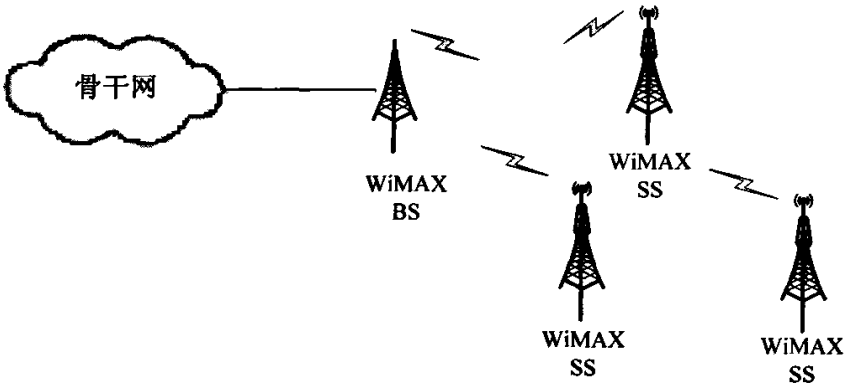

图 2-2 PMP 应用模式

2. Mesh 应用模式

如图 2-3 所示，Mesh 应用模式采用多个基站（BS）以网状网方式扩大无线覆盖区。其中，有一个基站作为业务接入点（SAP）与核心网相连，其余基站（BS）以无线链路与该 SAP 相连。因此，作为 SAP 的基站既是业务的接入点又是接入的汇聚点，而其余基站并非简单的中继站（RS）功能，而是业务的接入点[1]。

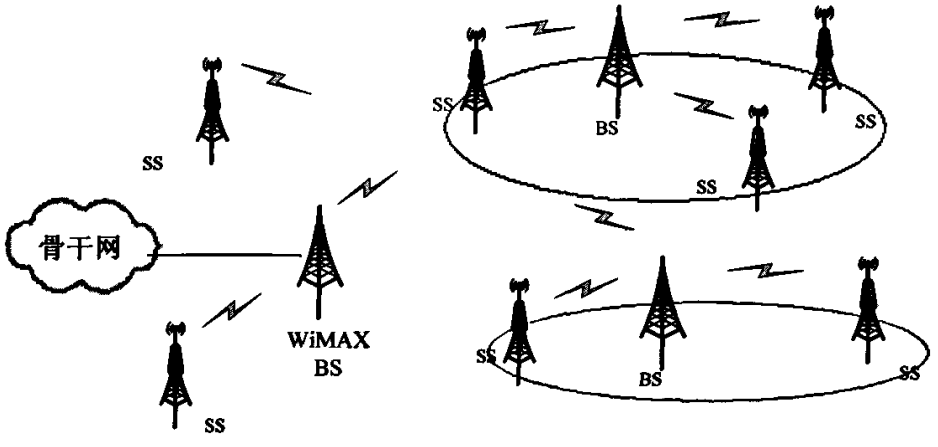

图 2-3 Mesh 应用模式

Mesh 应用模式的特点在于网状网结构，可以根据实际情况灵活部署，实现网络的弹性延伸。对于市郊等远离骨干网，有线不易覆盖的地区，可以采用该模式扩大覆

盖范围，其规模取决于基站半径、覆盖区大小等因素。

3. 综合应用模式

无线城域网的推出是为了满足日益增长的宽带无线接入（BWA, Broadband Wireless Access）市场需求。符合 802.16 标准的设备可以在“最后一英里”宽带接入领域替代 Cable Modem、DSL 和 T1/E1，也可以为 802.11 热点提供回传。新标准规范了一个支持诸如话音和视频等低时延业务的协议，在用户站和基站之间允许非视距的宽带连接，一个基站可支持数百上千个用户，在可靠性和 QoS 方面提供电信级的性能。总之，它充分考虑了为全世界通信公司和服务提供商设计一个可扩展、长距离、大容量“最后一英里”无线通信系统的需要，可支持一整套的服务，从而使服务提供商能够在降低设备成本和投资风险的同时提高系统性能和可靠性，有助于加速无线宽带设备向市场的投放以及“最后一英里”宽带在世界各地的部署。BWA 应用包括住宅宽带接入、用于 SOHO 和小企业的 DSL 级业务、用于企业的 T1/E1 级业务（所有这些不仅支持数据，而且还支持话音和视频），还包括用于热点的无线回传和蜂窝小区基站回传业务等，如图 2-4 所示[1]。

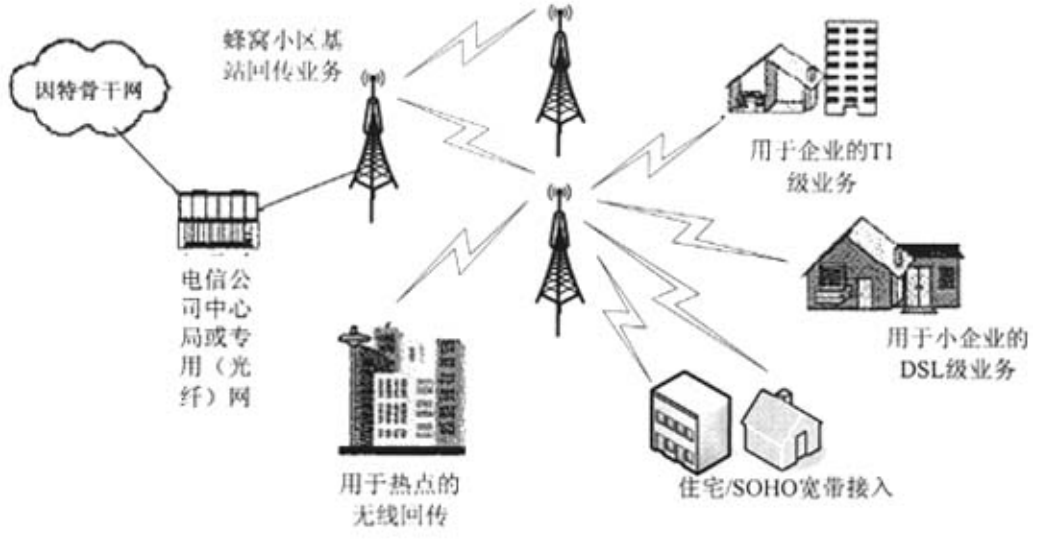

图 2-4 802.16 的 BWA 应用

4，校园网 WiMAX 应用实例

图 2-5 是以校园应用为例的 WiMAX 系统典型应用方案在实际当中的网络拓扑结构。WiMAX 基站既能够通过用户站为图书馆、教学楼等设备密集地区提供大容量的接入服务，也能够通过 AP 为笔记本等无线设备提供到核心网的连接，同时还能为 PDA、智能手机等移动设备提供高速率数据传输和服务[2]。

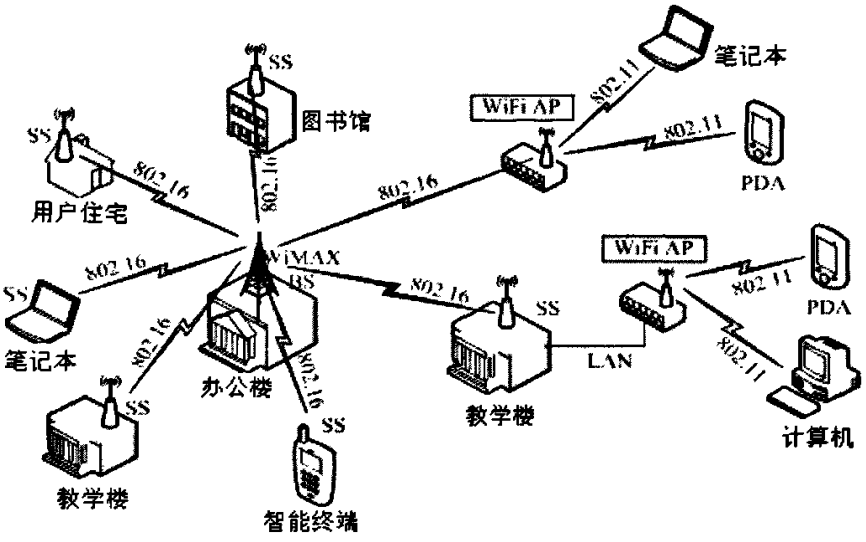

图2-5 WiMAX系统应用方案

第三章 IEEE802.16 MAC 层的协议分析

本章对 IEEE802.16 宽带无线接入网络中 MAC 层的技术进行讨论。本文中对于 802.16 MAC 层的讨论是基于 802.16－2004 标准，未考虑宽带无线接入网络中对于移动性的支持。

IEEE802.16 标准负责对于固定的宽带无线接入技术的空中接口制订标准，该标准涉及到网络的物理层和媒体访问控制层。物理层负责制订空中接口采用的底层通信技术，频带选择等。MAC 层分成三个子层：特定服务会聚子层（Service Specific Convergence Sublayer）、公共部分子层（Common Part Sublayer）、安全子层（Privacy Sublayer），各子层通过服务接入点进行通信。802.16 协议模型[3]如图 3-1 所示。

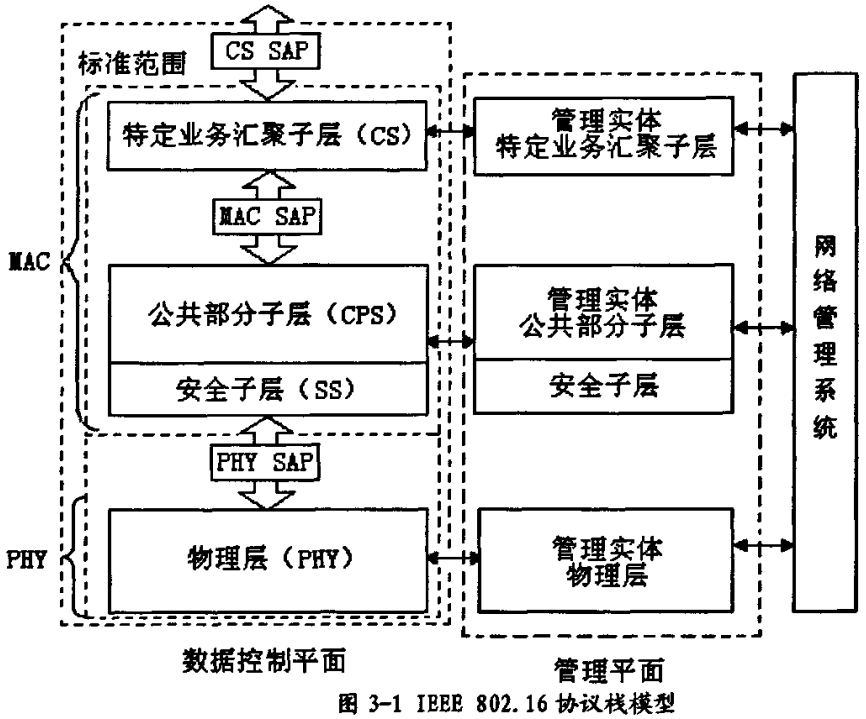

图 3-1 IEEE 802.16 协议栈模型

3.1 IEEE 802.16 MAC 机制

为了提供高效的信道访问控制机制，802.16 MAC 层定义了完善的控制机制，主要包括服务流会聚部分，网络初始化部分，测距部分，MAC 层协议数据单元构建和传

输部分，服务调度部分，带宽分配和请求部分，竞争解决部分以及自动重传机制等。

3.1.1 特定服务会聚子层

802.16 协议中定义了两种不同协议的会聚子层，即 ATM 协议下的会聚子层和分组协议下的会聚子层。特定的服务会聚子层位于 MAC 公共部分子层之上，而且通过 MAC 的服务接入点调用 MAC 公共部分子层提供的服务。服务会聚子层有下面几种功能[4]：

1) 接收高层的协议数据单元（PDU，Protocol Data Unit）。
2) 完成高层 PDU 的定义。
3) 按规则对高层 PDU 进行分类处理。
4) 完成服务会聚子层中的 PDU 到特定 MAC 服务接入点的数据传送。
5) 接收来自对等实体的服务会聚子层的 PDU。

3.1.2 网络初始化过程

在网络初始化过程中，上下行参数获得，初始化测距过程，基本能力协商，注册和建立 IP 连接是核心部分。

1) 上下行链路同步和参数获得

为了获得网络的上下行链路通信参数，SS 通过接收 UCD，UL—MAP，DCD，DL—MAP 消息获得网络的上下行参数和同步信息。

2) 初始化测距过程

测距过程是 SS 和 BS 之间保证无线通信质量的一系列过程。其中包括下行链路 Burst Profile 管理，上行链路周期性测距和针对 OFDMA 物理层模式的测距机制。测距过程交互的参数包括链路时序，功率级别等。

3) 基本通信能力协商

SS 和基站协商对基本通信能力的支持。其中包括带宽分配支持参数，PDU 构建、传输参数和物理层支持参数。

4) 节点注册

注册是SS被允许进入网络的过程，一个被管理的SS在此时接收到它的Secondary CID。注册过程中采用 REG—REQ 和 REG—RSP 消息交互。在注册过程还包含 IP 版本协商。

5) 网络配置（IP）

SS 发起 DHCP 过程以获得一个 IP 地址和建立 IP 连接需要的其他参数。建立 IP 连接的参数在 Secondary CID 上传输[7]。

6) 获得系统时间

SS 和 BS 都需要获得当前的系统时间，以完成一些需要时间戳的控制事件。

3.1.3 MAC 层协议数据单元的构建和传输

在802.16 标准对MAC 层协议数据单元（PDU，Protocol Data Unit）的格式和构建方式做了详细的规定，这部分是MAC层传输控制功能的基础。802.16 MAC层数据和控制消息包括下面几种：

1) 两种PDU 头格式：通用包头，带宽请求包头。
2) 6 种PDU 子头：分段子头，拼接子头，授予管理子头，ARQ 反馈，Mesh子头和快速反馈子头。其中比较关键的是分段子头，拼接子头和授予管理子头。
3) 47 种管理消息：16 标准中规定了UCD，DCD，UL-MAP，DL-MAP，RNG-REQ 等47种控制消息。

3.1.4 测距过程

1) 下行Burst Profile 管理

SS 可以根据接收到的信号的信噪比信息确定目前信道情况是否满足当前通信需要。如果发现不行，则启动改变下行Burst Profile 的过程。有两种途径，如果SS 有BS 授予的上行传输机会，在这个传输机会中利用DBPC-REQ 消息发起更改下行Burst Profile 的过程。如果没有可用传输机会，则SS 在初始化测距过程发起RNG-REQ 消息，利用新的测距过程获得更好的下行Burst Profile。

2) 上行周期性测距

上行测距包含两个过程，初始化测距和周期性测距。初始化测距过程中，刚进入网络的SS 获得正确的传输参数，包括传输时间偏移和传输功率级别等。周期性测距过程使得SS 可以在初始化测距之后再对传输参数进行调整，以保证与BS 的链路通信质量。

3.1.5 信道分配和请求机制

带宽分配和请求机制是802.16MAC 层规定的为SS 静态和动态分配带宽的机制。带宽分配的机制包括：

1) 带宽请求

发送带宽请求有三种方式[3]：

a) 在竞争请求时隙中发送带宽请求（Bandwidth Request）包。

b) 在授予（Grant）时隙中发送带宽请求消息。

c）搭乘（Piggyback）方式。

2） 带宽授予

带宽授予是BS 向SS 分配上行链路时隙的动作。分配的时隙是利用UL-MAP消息通过Basic CID 连接发送给SS 的。因此这里需要SS 做上行调度，从自己的多个连接中选择一个连接发送数据。

3） 轮询

轮询是BS 轮流给SS 授予带宽请求机会的过程。轮询过程中BS 给SS 轮询的是带宽请求机会，不是发送数据的带宽。轮询分为单播轮询和多播轮询。

3.1.6 MAC 层竞争解决

MAC 在上行链路MAP 消息中定义了一系列的消息元素（IE，Information Element），从而实现上行链路时隙的分配并确定哪个时隙发生了碰撞。BS 可以允许碰撞发生在REQ 或者数据PDU 中。主要的竞争解决方法是截断的二进制指数回退。初始回退窗和最大回退窗的值由BS 控制，该数值在上行链路描述UCD消息中规定。当SS 要传送消息进入竞争解决过程时，它将起始回退窗设为与当前有效的上行链路描述消息中规定的起始值相等，然后随机选择一个回退窗口范围内的数字。该随机数指出了该SS 在传送消息之前所要等待的竞争传输机会的个数。在竞争要求发送之后，SS 要等待来一个数据认可（ACK）消息，收到后该竞争解决过程就完成了。若没有收到ACK 消息，则表明传送消息丢失，SS 将回退窗口加倍，选择新的随机数，重复上述过程。

3.2 IEEE 802.16 MAC 层的 QoS 机制

802.16 MAC 层是基于“连接”的，每一个“连接”均由一个标识符(CID) 来唯一进行标识。在802.16 标准中，MAC 层定义了较为完整的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，标准还定义了四种不同的业务，分别为：非请求的带宽分配业务（UGS）、实时轮询业务（rtPS）、非实时轮询业务（nrtPS）、尽力而为业务（BE）。802.16 可以根据业务的需要提供实时、非实时的不同速率要求的数据传输服务。802.16 目前主要面向宽带数据业务，也可以提供话音业务。

802.16 系统的QoS 机制可以根据业务的实际需要来动态分配带宽，具有较大的灵活性。802.16 可以在无线接入网部分为不同业务提供不同质量的服务，16MAC层QoS支持相关的机制主要包括下面几个部分。

3.2.1 SS 和 BS 之间的连接建立

SS 是宽带无线接入系统中的终端（接入单元），BS 是宽带无线接入系统中为接入单元提供接入服务的控制实体。一个BS 可以控制多个SS，并且它们提供无线的接入服务。SS 和BS 的连接建立在网络初始化过程中完成。首先SS 需要获得一个BS 信道同步以及上下行链路信息，通过交换上下行链路信息建立物理连接。SS 的网络初始化过程还包括注册，测距等过程。通过上述过程，SS 可以找到一个BS 并且完成与BS 的通信能力协商，认证和建立控制连接等过程。

IEEE 802.16 的MAC 协议支持TDD 和FDD 工作方式。在TDD 方式中，信道被分成一系列的微时隙（mini-slot）。SS 和BS 已经建立了同步，所以BS 可以通过控制把信道中的某些时隙分配给制定的SS。SS 通过上行链路向BS 发送数据和信道请求消息。BS 在下行链路上给SS 返回下行数据和相关的控制信息。

3.2.2 服务流

为了支持QoS， IEEE 802.16 定义了4 种服务流[5]：非请求授予服务（UGS，Unsolicited Grant Service），实时轮询服务流（rtPS, Real-time Polling Service），非实时轮询流（nrtPS，Non-realtime Polling Service）和尽力而为服务（BE, Best Effort）。

(1) 非请求授予服务流（UGS）：这种服务流是为了支持恒定比特率（CBR）的实时数据流，例如无静音抑制的VOIP（Voice Over IP）数据。 这种业务需要固定的网络带宽保证。

(2) 实时轮询服务流（rtPS）：这种服务流是为了支持变化比特率（VBR）的实时数据流，例如MPEG 视频数据。这些业务有具体的网络带宽要求并且有最大延迟时间的限制。

(3) 非实时轮询服务流（nrtPS）：这种服务流是为了支持非实时数据流，但是这些数据流要求比尽力而为服务更好的服务质量，例如带宽敏感的文件传输服务。这些业务对延迟时间不敏感但是需要保证最小的网络带宽。

(4) 尽力而为服务（BE）：这种服务流是针对无特定服务质量要求的数据。例如HTTP 数据。

针对上述的几种服务流，16MAC 层定义了详细的服务流管理机制。该管理机制中定义了详细的QoS 操作理论，服务流特性，对象模型，服务分类和授权模型等。通过前面介绍的动态服务流改变，添加和删除消息可以完成服务流的动态管理和更新。

3.2.3 带宽分配和请求机制

在宽带无线接入系统中，存在着上行和下行信道，上行信道指SS 到BS 的通信信道，下行信道指BS 到SS 的通信信道。下行信道由BS 集中控制，而多个SS 共享上行信道，这就涉及到网络带宽在多个SS 之间的分配问题。IEEE 802.16规定了带宽请求和请求机制。请求指的是SS 在需要获得网络带宽发送数据时向BS 发出请求，要求获得上行信道带宽的过程。SS 的带宽请求有三种方式，一种是在上行信道的竞争请求时隙中发送带宽请求消息；另一种在BS 分配的请求时隙中发送请求消息；第三种是在数据时隙中采用搭乘方式发送请求。BS 采用多种方式完成网络带宽的分配，其中包括授予机制（Grant）和轮询（Polling）机制。授予机制就是BS 直接在上行信道的MAP 消息中给SS 分配数据发送时间；轮询机制是BS 采用轮询的方式给各个SS 带宽请求机会，轮询机制包括单播轮询和多播轮询机制。需要特别指出的是，IEEE 802.16-2004 中规定，SS 向BS 发送带宽请求时是为具体的数据连接请求带宽，而BS 向SS 分配带宽时却不是针对每个连接的，而是针对某个SS 分配带宽，这一点会影响调度策略的设计。

第四章 基于 picoChip 平台的 WiMAX 系统设计

WiMAX系统的总体设计包括硬件平台设计、嵌入式系统设计和协议软件设计三个方面。硬件设计包括CPU芯片和相关开发板选择、外围设备芯片选择、原理图和PCB设计及绘制；嵌入式系统设计包括嵌入式操作系统选择和相关BSP及外围设备驱动程序开发；协议软件开发包括基于IEEE 802.16标准的协议分析、模块划分和结合嵌入式操作系统的软件实现。

硬件设计是整个系统中最基础也最具决定性的部分，操作系统和协议软件开发都要在硬件平台上进行。本项目采用的开发板是picoChip的开发板。

嵌入式操作系统选择了美国 WindRiver System 公司推出的VxWorks实时操作系统。VxWorks是一个高性能、可裁减的嵌入式实时操作系统；另外，风河公司从1995年起推出了针对VxWorks的集成开发环境－Tornado，它是一整套强有力的交叉开发工具，支持多种形式的在线仿真和跟踪调试。

WiMAX系统需要实现的协议功能包括物理层、MAC层和MAC层与网络层的接口。其中物理层和MAC层涵盖在IEEE 802.16标准中定义的空中接口的规范；MAC层与网络层的接口用于实现WiMAX系统和现有网络的互连互通，此处的网络层与OSI七层模型中的网络层相当。

4.1 picoChip 嵌入式开发平台

PC102是picoChip公司开发的高性能通信处理器阵列，该处理器针对无线数字信号处理的应用做了专门的性能优化。利用PC102处理器可以高效率的用软件实现所有通信系统中的物理层信号处理和控制过程。PC102处理器芯片包括一组RISC处理器阵列，专用的协处理器，外部处理器接口，外部存储器接口，同步和异步IO[7]。

基于 PC102 处理器芯片构建的无线通信系统的软件开发平台，该平台命名为DVK102。DVK102 是一个嵌入式的开发平台，包括一个子板和一个母板。子板是一个 PC102 芯片的开发环境，采用 PC102 芯片完成无线信号处理过程；母板作为整个开发平台的控制板，负责子板的驱动和系统测试的功能。DVK102 有下面的特点：

a) PC102 芯片的硬件开发环境；

b) 母板集成的 IBM 266MHz 405GP PowerPC 处理器控制 PC102 设备，处理

DMA 数据和 IO 接口。母板支持 QNK 和 VxWorks 操作系统；

c) 外部 PC 主机可以通过 10/100M 以太网口与 DVK102 平台连接；

d) 128Mbyte 的 PC133 数据 SDRAM 和最高 32Mbyte 的本地 Flash，用来存放 PC102 镜像和配置数据，实时操作系统和开发平台的设置信息；

e) 与 DVK102 配套的主机开发软件包括：基于以太网的 DVK102 驱动，DVK102 硬件测试程序，DVK102 功能演示程序；

f) 母板支持 JTAG 口的调试；

g) 母板支持两个 RS232 串行接口。

DVK102的子板体系结构如图4-1所示。

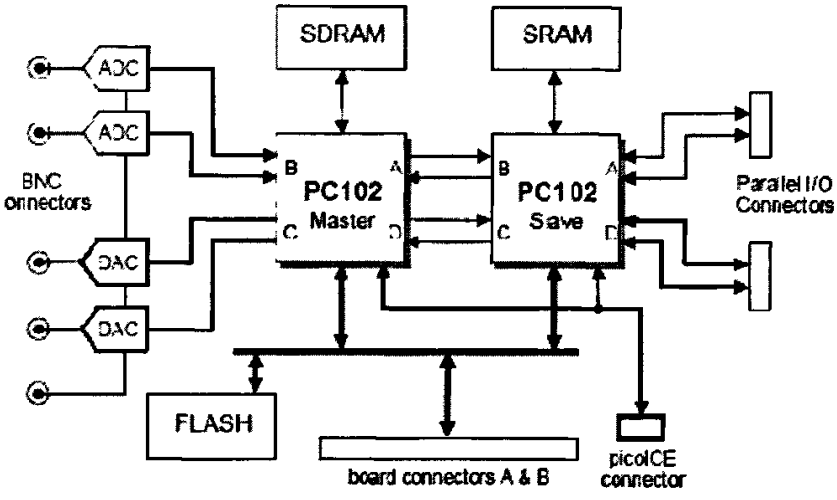

图 4-1 DVK102 子板系统结构图

DVK102 的子板是一个 PC102 芯片的硬件开发环境，通过在子板上运行不同的软件模块，可以使得该子板完成不同的通信信号处理功能。目前子板上可以完成 3G 和 WiMAX 信号处理。子板上包括 PC102 阵列处理器芯片（2 个），数据存储器，异步数据接口，数据转换器，FLASH 等部件。

DVK102 的母板是一个常见的嵌入式开发板，母板上包括 IBM405GP PowerPC 的处理器（处理器工作频率 266MHz），SDRAM，Flash 和 E2PROM 存储器，以太网接口，专用子板接口（该接口专门用于子板和母板之间的连接），RS232 串行接口，状态部件和电源部件等。DVK102 的母板系统结构如图 4-2 所示。

运行在母板上的软件系统包括 DVK102 服务器软件，VxWorks 操作系统，BSP，网络协议栈和 picoAPI；运行在子板上的软件包括信号处理软件和 picoAPI。

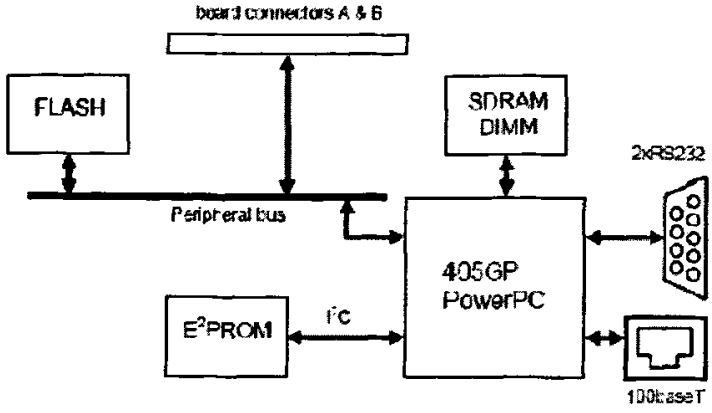

图 4-2 DVK102 母板系统结构图

运行在母板上的软件系统包括 DVK102 服务器软件，VxWorks 操作系统，BSP，网络协议栈和 picoAPI；运行在子板上的软件包括信号处理软件和 picoAPI。

4.2 802.16 MAC 的实现框架

基于 picoChip 公司开发的 DVK102 开发平台和 PC8520 OFDM 物理层软件参考设计实现 802.16 MAC CS 子层的上层业务接入和数据分类过程，MAC CPS 子层的管理和数据处理过程，从而实现完整的 802.16 基站设备。基站 OFDM 物理层和 MAC 底层的控制功能，由 8520 软件参考设计在 DVK102 的子板中实现，MAC 上层的控制功能在 DVK102 母板上通过软件实现，系统功能组件结构如图 4-3 所示。

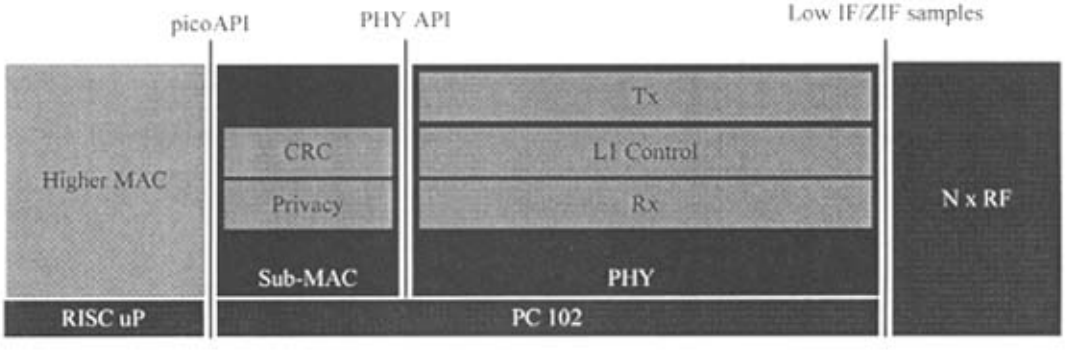

图 4-3 802.16 基站设备软件结构图

802.16 MAC CS 子层主要是数据分类和分类映射表管理部分；CPS 子层分为六个部分实现：公共支持部分、物理层管理部分、基站初始化部分、无线台初始化处理部分、MAC 控制按需处理部分、数据处理部分。系统总体设计框图如图 4-4 所示。

下面描述 CPS 子层各部分包括的子模块和主要功能，关于 CS 子层的模块划分和功能描述以及实现将在下面的章节中介绍。

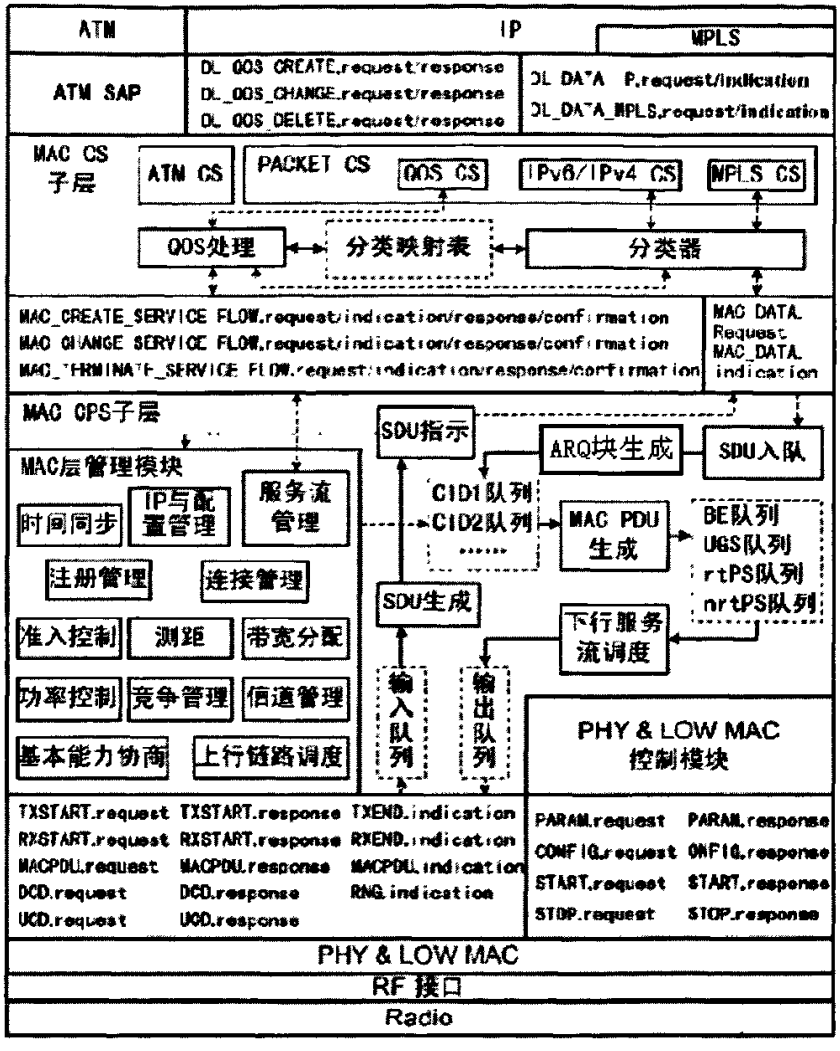

图 4-4 802.16 基站 MAC 层的总体设计框图

1） 公共支持部分

本部分提供 MAC CPS 子层可使用的通用支持函数，包括如下几个模块。

a) 队列管理模块

提供对缓冲区、链表、队列、堆栈等数据结构进行管理的基本操作函数；提供增加、修改、删除、查询服务数据单元（SDU，Service Data Unit）队列、服务流队列、带宽请求队列、MAC PDU 输出队列的基本函数。

b) 定时器管理模块

提供操作（增加、修改、删除等）定时器的基本函数。

2) 物理层管理部分

a) 物理层管理模块

提供启动、配置、停止物理层网络设备的接口；对物理层的信道检测结果进行分析与记录，为其他模块提供查询接口。

3) 基站初始化部分

a) 基站配置模块

完成基站的初始配置，设置基本参数，启动物理层网络设备。

4) 无线台初始化处理部分

a) 信道描述模块

确定上下行信道描述消息里的相关参数，生成 UCD/DCD 消息，用于周期性发送；处理信道描述符更新过程。

b) 测距模块

完成初始测距和周期性测距过程；处理终端发来的 Ranging 消息，从连接管理模块处获得 Basic CID 和 Primary CID。

c) 基本能力协商模块

根据物理层管理模块提供的信息，对终端发来的 SBC－REQ 进行处理；将基本能力信息录入连接管理模块维护的全局数据结构。

d) 准入控制模块

对终端进行设备认证；处理密钥交换。

e) 注册管理模块

处理终端发来的 REG－REQ 消息，从连接管理模块处获得 Secondary CID；协助终端完成 TFTP 过程；将注册信息录入连接管理模块维护的全局数据结构。。

f) 网络信息配置模块

为终端提供 DHCP 服务器地址（DHCP）服务器中包含了时间服务器地址、配置服务器地址），辅助终端完成过程：建立 IP 连接，网络时间同步，可操作性参数配置，预连接建立，（ID，IP）注册。

5) MAC 控制按需处理部分

a) 上行链路调度模块

处理终端发来的带宽请求，通过调度算法合理分配带宽，组成并且发送 UL－MAP 消息。

b) 连接管理模块

CID 的管理与生成，维护 CID/SDU 队列。

c) 服务流管理模块

对服务流控制消息进行处理，维护服务流队列，向 CS 层提供服务流管理服务。

d) 管理消息分类模块

解析 MAC 层管理消息，根据其类型调用相应模块作进一步处理。

6) 数据处理部分

a) 接收与预处理模块

接收 MAC PDU，SDU 重组，分类出 MAC 管理消息和数据 SDU（ARQ）；将 SDU（ARQ 的话传给 ARQ 块）传给 SDU 输入输出模块处理，将管理消息分类传递到 MAC 层模块进行处理。

b) SDU 入队模块

向 CS 层提供数据传送服务，将 CS 传来的 SDU 置入相应的 CID/SDU 队列中。

c) ARQ 模块

将 CS 传来的 SDU 分成 ARQ 块，放入到发送队列中，对 ARQ 块进行管理，重发或删除；接收对端的 ARQ 块，进行块的重组，发送给 SDU 输入输出模块。

d) MAC PDU 生成模块

根据 CID、SDU 和其他头信息生成 MAC PDU。

e) 下行服务流调度模块

利用合适的调度算法对 PDU 进行调度，确定 DL－MAP，管理 DL－MAP 队列。

f) 数据输出模块

读取 DL－MAP 队列，根据 DL－MAP 中的信息组织各数据单元进入输出队列。

系统内部接口根据各个功能模块功能定义了 MAC 层中各个模块之间的数据关系和控制关系，如图 4-5 所示。

4.3 802.16 MAC CPS 层的主要模块接口设计

在本节中，只是介绍 MAC CPS 层的模块接口设计，MAC CS 子层的模块和接口将在下一章中进行详细的论述。

1) 数据接收与预处理模块

接收来自Low MAC的MAC PDU，根据其头部信息判断类型。PDU 中可能包括数据SDU（ARQ块）、数据片Fragment 和MAC 层控制消息。MAC PDU经数据接收

与预处理模块后分离出数据SDU（ARQ块）、SDU中的带宽请求和MAC 消息，数据SDU（ARQ块）根据所属的CID号被发送到连接管理模块中的数据SDU缓存模块中，SDU中的带宽请求被发送到上行链路调度模块中的带宽请求缓存队列中，MAC消息被发送到管理消息分类模块做进一步的分类。数据接收与预处理模块与其他模块的接口如图4-6所示。

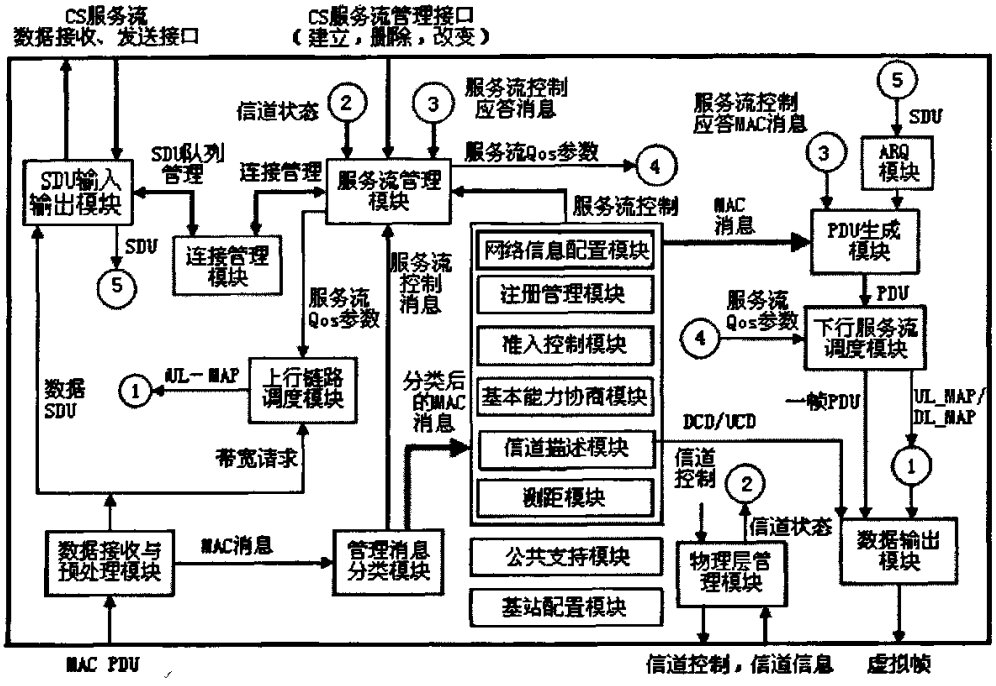

图 4-5 802.16 MACCPS 子层模块结构图

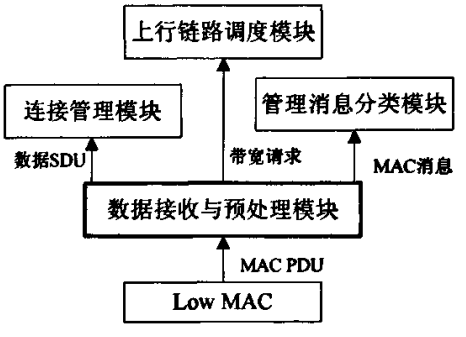

图4-6 数据接收与预处理模块接口

2） 上行链路调度模块

上行链路调度模块输入接口包括接收来自数据接收与预处理模块和管理消息分

类模块的带宽请求，从服务流管理模块中读取服务流的QoS参数。输出接口为代表带宽分配状况的UL_MAP消息，UL_MAP被发送到数据输出模块，填充到物理层帧中进行发送。上行链路调度模块接口定义如图4-7所示。

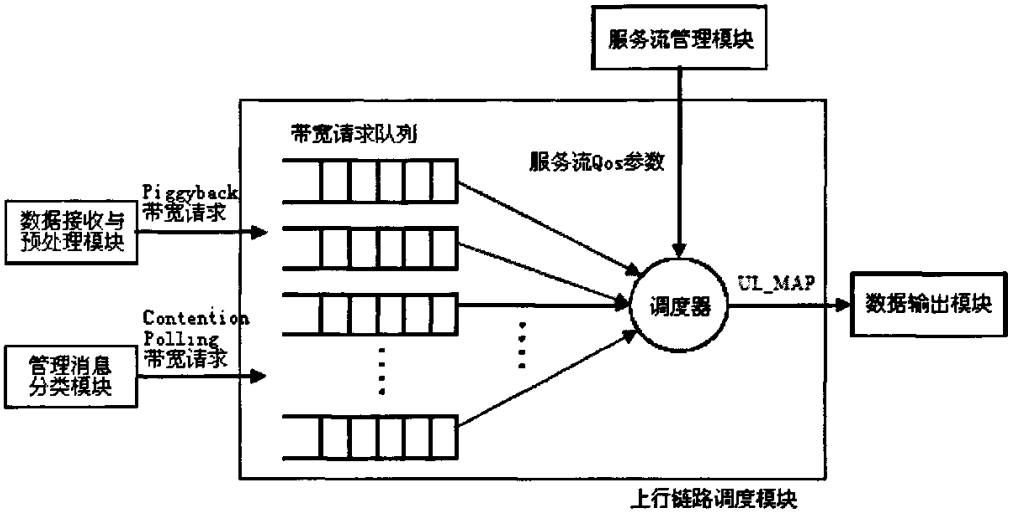

图4-7 上行链路调度模块接口

3) 服务流管理模块

服务流管理模块中保存有服务流的管理信息表，管理信息表中包括服务流的QoS参数（不同类型的服务流具有不同的表项）、代表物理信道通信质量的统计参数、服务流数据传输的Profile类型等。CS 层通过服务流管理接口建立、删除和改变服务流的QoS 参数。当新建一个服务流时，在服务流管理模块中的服务流管理信息表中添加对应的管理表项，服务流的添加可以是CS调用服务流管理接口发起，也可以由SS向BS发送DSx 消息发起。服务流管理模块在建立、删除和改变服务流时产生的服务流控制应答消息（DSx.response）通过PDU生成模块发送到对等MAC。下行服务流调度模块和下行链路调度模块根据当前调度的服务流ID从服务流管理模块中读取服务流QoS参数等信息进行调度。服务流管理模块接收来自物理层管理模块中物理信道状态的信息，包括物理层支持的工作模式（调制解调方式、编解码方式、数据传输速率等）、当前信道通信质量等，并由服务流管理模块中的管理信息表来统一管理。模块接口设计如图4-8所示。

4) 管理消息分类模块

管理消息分类模块解析MAC消息中的头部信息（消息类型字段），对MAC消息进行分类，并将消息发送到相应的消息处理模块中。管理消息分类模块设计以及接口定义如图4-9所示。

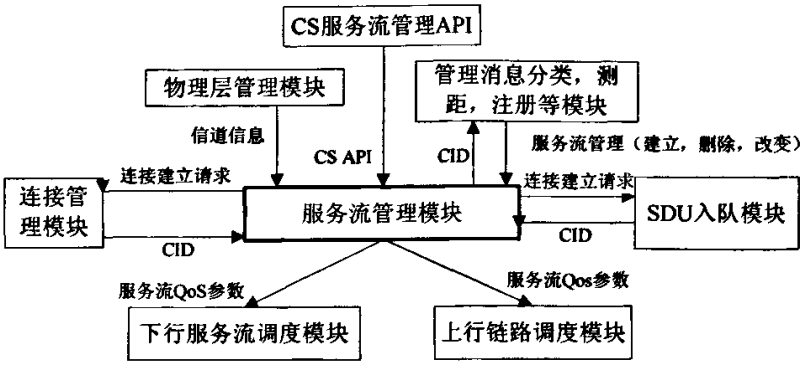

图4-8 服务流管理模块接口

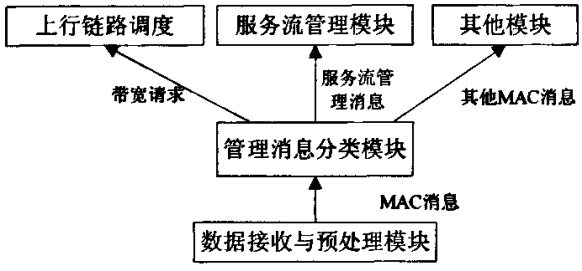

图4-9 管理消息分类模块接口

5) 连接管理模块

连接管理模块负责建立、删除和管理连接的ID号，每个连接对应一个数据缓存队列。连接管理模块接收来自数据接收与预处理模块中的数据SDU，将其缓存到对应CID的队列中。连接管理模块还负责将SDU数据发送到CS层。连接的建立由服务流管理模块发起，连接管理模块在分配CID之后将其返回给服务流管理模块。连接管理模块与其它模块的接口如图4-10所示。

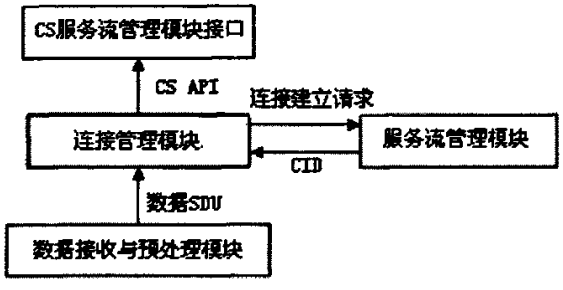

图4-10 连接管理模块接口

6) PDU 生成模块

PDU生成模块完成SDU（ARQ块）和MAC消息到MAC PDU的转换，包括对SDU

进行分段操作。如果对应的服务流定义了负载头部压缩（PHS，Payload Head Suppression），则根据PHS 的参数进行包头压缩。是否进行PHS 以及PHS 的参数在服务流管理模块中定义。PDU生成模块还负责将生成的PDU发送给数据输出模块。MAC PDU生成模块接口设计如图4-11所示。

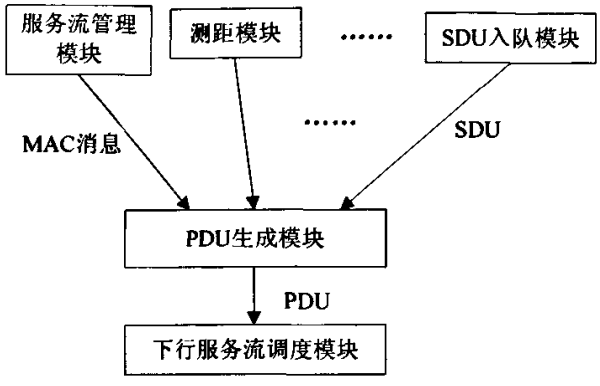

图4-11 MAC PDU 生成模块接口

7) 下行服务流调度

下行服务流调度模块根据服务流管理模块中对应服务流的QoS参数生成代表下行带宽分配状况的DL_MAP消息，DL_MAP消息发送给数据输出模块。同时，下行服务流调度模块根据调度结果，从SDU入队模块中取出本次调度对应的SDU数据，并将对应的SDU 数据发送给PDU 生成模块。模块的接口设计如图4-12所示。

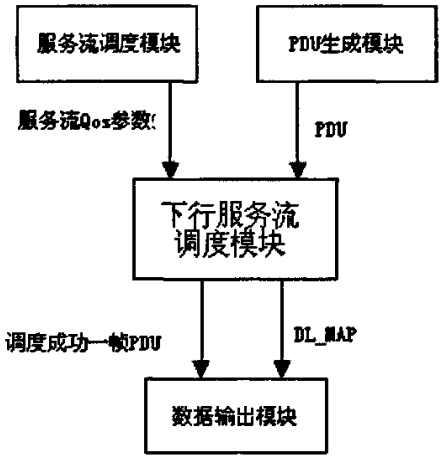

图4-12 下行服务流调度模块接口

4.4 本章小结

本章介绍了基于picoChip DVK102嵌入式系统开发平台和8520 OFDM PHY软件参考设计实现802.16 MAC层协议的系统设计方法。802.16MAC层软件实现运行在DVK102的母板上，通过调用子板上运行的OFDM物理层实现一个完整的802.16协议功能。802.16MAC分为CS子层和CPS子层，CS子层分为分类器和分类规则管理两部分，CPS子层分为六个部分实现：公共支持部分、物理层管理部分、基站初始化部分、无线台初始化处理部分、MAC 控制按需处理部分、数据处理部分。本文中给出了802.16MAC CPS子层的总体设计框架，详细模块划分和模块接口定义。CS子层的实现将在下一章进行论述。该方案被应用在802.16无线城域话音通信系统的基站设备软件设计和实现中。

第五章 会聚子层(CS)的设计与实现

IEEE 802.16 MAC 是面向连接的，协议定义了两种 CS 子层将上层业务映射成连接。ATM CS 子层和包(Packet)CS 子层。ATM CS 子层提供对 ATM 的业务支持；包(Packet)CS 提供对 IEEE 802.3(Ethernet)、802.1Q(VLAN)、IP(IPv4、IPv6)等基于包的业务的映射。由于目前通信网络中最大的数据业务是基于 IP 的分组业务，而且 WiMAX 组织仅认证与 IP 相关的 IEEE 802.16 设备，因此本系统中主要研究 Packet CS 的特点和应用场景以及实现。

5.1 面向分组业务的会聚子层(CS)

5.1.1 会聚子层（CS）的功能

WiMAX/802.16 标准中定义了面向分组业务的会聚子层(Packet CS)需要完成以下功能[4]：

1) 对高层到达的 PDU 进行分类，并将其分配给相应的连接；
2) 对净荷(payload)的报头信息进行压缩，该功能是可选的；
3) 将重组以后的 CS PDU 送往 MAC CPS SAP，进而发往目的节点的对应实体；
4) 接收来自对等实体的 CS PDU；
5) 对于进行了报头压缩的 PDU，重建其报头包含的所有信息，该功能可选。

总的来说，发送节点的 CS 子层负责把 MAC SDU 送到对应的 MAC CPS SAP。之后，MAC CPS 子层将 SDU 送到目的节点的对等 MAC SAP。在分组的发送过程中，发送节点将通过一切手段来满足连接或服务流预先约定的 QoS 要求。相应的传输功能包括：QoS 机制、分片、串联。接收的对等 CS 子层收到该 SDU 后，最终送往高层应用实体。

对到达的SDU进行分类的过程实际是将接收到的SDU映射到某一个连接上的过程。这一映射的过程将该 SDU 与某一个特定的连接相关联。同时，这个映射的过程也将 SDU 与连接所承载服务流的特征建立起对应关系。所以，这个映射过程最终的结果是使得 SDU 在传递的过程中，加入了一些服务质量 QoS 的控制。

Packet CS 子层定义了分类器(Classifier)的概念。分类器就是一组匹配法则的集合。这一组法则将被应用于每一个进入 WiMAX/802.16 网络的分组。通常来说，分类

器包括了针对特殊协议定义的分组匹配方法(比如，根据目标 IP 地址，协议类型等)、分类器本身的优先级以及到某一个连接标识符的应用。当一个分组符合所定义的匹配规则时，分类器就将其送到对应的服务访问点，并通过由 CID 标识的连接发往目的节点。而对于该分组的 QoS 的限制则由对应服务流的特征决定。在 WiMAX/802.16 网络中，每个分类器的实现都是可选的。

WiMAX/802.16 系统还允许下面的情况出现：多个分类器指向同一个服务流。当存在多个不同的分类器时，它们的优先级决定了应用的顺序。因为不同分类器所使用的映射模式可能会有重叠，所以有必要进行显示的排序。虽然 WiMAX/802.16 系统不要求每个优先级只有一个分类器，但是一定要避免在分类的过程中出现含糊不清的情况，从而引发错误的映射。

5.1.2 会聚子层(CS)的处理流程

在基站侧的会聚子层 CS 的处理流程如图 5-1 所示。

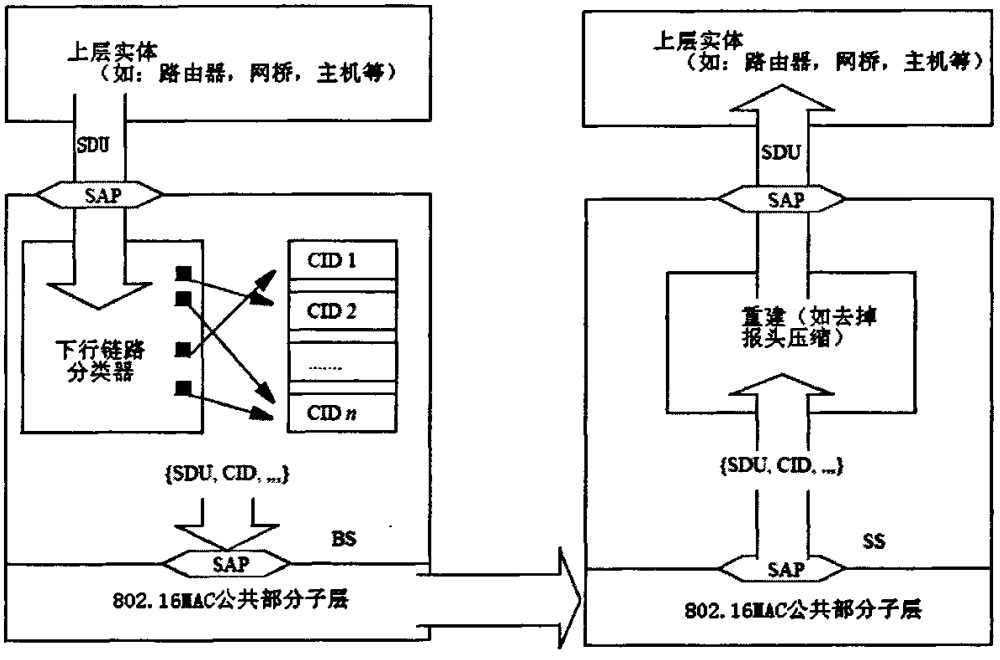

图 5-1 分类与 CID 的映射（BS 到 SS）

在用户侧的会聚子层 CS 的处理流程如图 5-2 所示。

在这个处理流程中，主要包括了分类(Classification)和CID对应(CID mapping)两个过程。

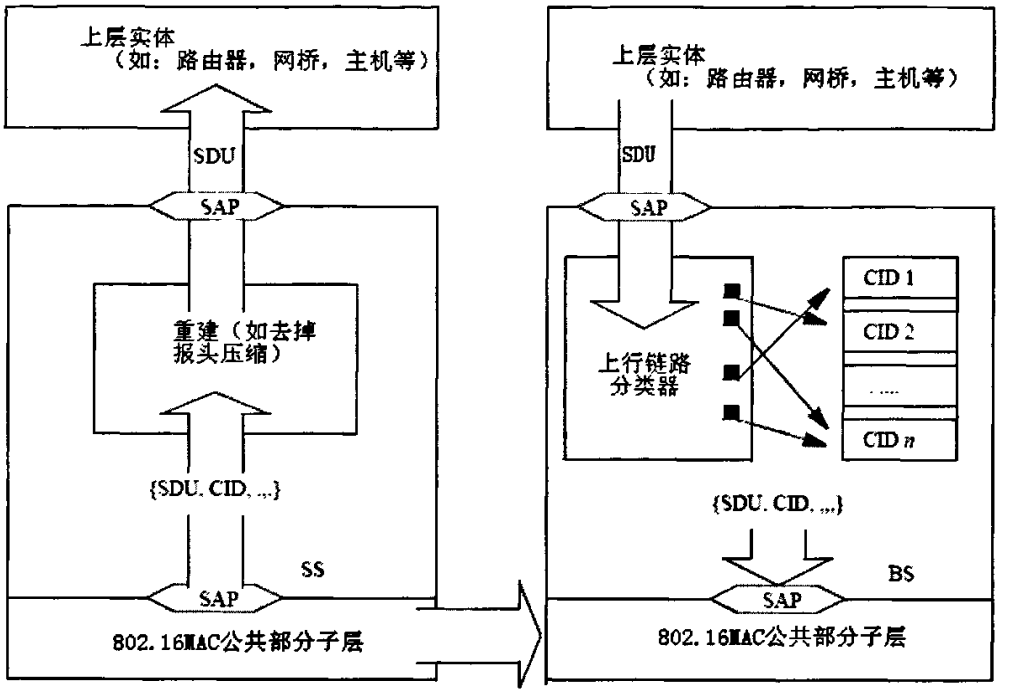

图5-2 分类与CID的映射（SS到BS）

5.1.3 会聚子层(CS)的实现框架

会聚子层(CS)的驱动程序主要由以下几个模块组成：MUX接口模块、分类器模块、服务流管理模块、数据接收及分类模块、数据包/服务流请求缓存模块、MAC接口模块、地址转换模块。模块关系如图5-3所示。

各模块的功能如下

1) MUX接口模块

该模块主要完成与 VxWorks 中 MUX 层的交互，作为驱动程序和网络协议栈的接口。主要功能是装载、启动设备，缓冲区管理，接收、发送数据接口，IP 地址和 MAC 地址转换。把应用程序的一系列原语转化为 CS 驱动程序的动作，这些原语包括两类：服务流管理原语、数据原语。

2) 分类器模块

该模块根据从 MUX 接口模块中发下来的 SDU 包，从映射表中查询到该包所属的服务流，然后通过 MAC 接口模块，将该包传送到 MAC 程序的 SDU 输入输出模块中。如果没有相匹配的服务流，对于 IP 数据包，则根据包的 IP 地址，向 MAC 层发送 MAC_CREATE_SERVICEFLOW.request 原语建立服务流，并将该数据包和请求缓

存到数据、请求缓存表中。如果在地址转换表中没有查到相应的 MAC 地址则丢弃该包（此处，不对没有找到 MAC 地址的数据包进行处理）。

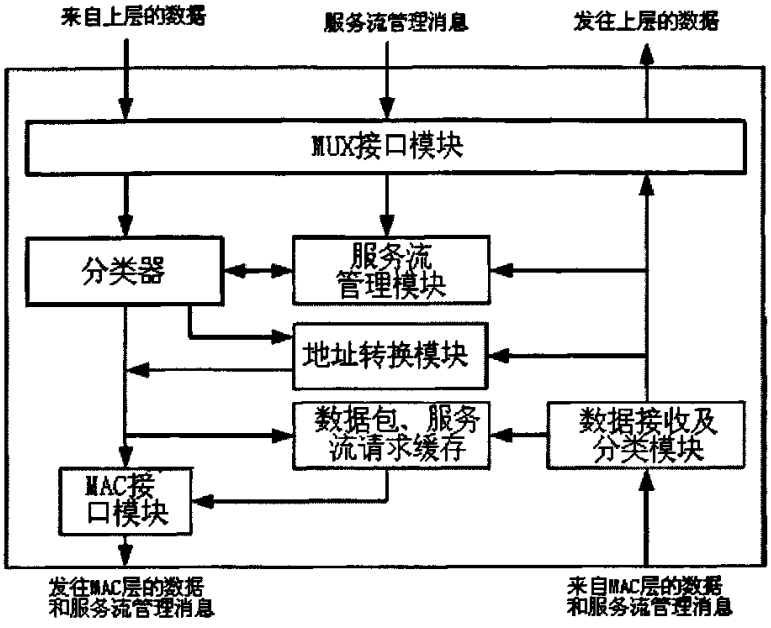

图5-3 CS子层的模块划分

3) 服务流管理模块

维护分类器映射表，提供基本的映射表维护、查询函数。保存分类器规则表。对所有服务流管理消息进行响应。

4) 数据接收及分类模块

该模块作为一个任务运行，接收来自 MAC 层 SDU 输入输出模块和服务流管理模块发来的消息，并对消息进行判断。主要包括两类消息：服务流管理消息、数据消息。数据消息通过 mux 接口传到协议栈上层。如果是服务流管理消息则进行解析并更新服务流管理模块数据。如果是 confirmation 消息，则对缓存表中的请求消息进行处理。

5) MAC接口模块

将其他模块发来的包封装成合适的格式，然后发送到 MAC 程序。需要发送的消息主要包括两类：服务流管理消息和服务流数据消息。

6) 数据包、服务流请求缓存

对于尚未映射 cid 的数据包进行缓存。对其他模块发出的等待回应的服务流管理消息进行缓存。用 seq_num 做为索引。

7) 地址转换模块

提供 MAC 地址、basic cid 和 ip 地址之间的映射。对地址消息进行响应。

根据图 4-1 和图 4-2 可知，会聚子层 CS 的作用就是查找分类规则库，将数据映射到相应的 CID 进行发送；没有相匹配的规则，则向 MAC CPS 子层发送服务流建立请求，并将该数据包和请求缓存。在整个会聚子层中，主要的功能是分类器功能，其他模块都是为实现分类器功能进行服务的。因此，在下面的章节中，重点讨论分类器的设计和实现，为了节省篇幅，对其他模块的设计和实现只是略述。

5.2 分类器的设计与实现

分类器是一系列映射标准的集合，每个进入 IEEE 802.16 网络的数据包根据分类器定义的规则映射成为连接。分类器可以通过配置得到或动态建立，SS 进入网络时也可以通过空中接口从基站(BS)获得。MAC 层的每个连接由长度为 16bit 的连接标识(CID)唯一标识，这种基于连接的机制是提供 QoS 保障的基础。同时 CS 子层对于特定业务还可以进行进一步处理。

5.2.1 现有的几种 IP 分类方法

1) RFC 算法(Recursive Flow Classification)

RFC 是由 Pankaj Gupta 和 Nick McKeown 提出的一种适合多域流分类问题的算法，具有流分类速度快，直接支持范围和前缀匹配等优点。但所需存储空间太大。如果该算法所基于的特征在所用的分类器中不具有或不明显，每一维长度的压缩量将很小，这将严重影响流分类的性能。该算法的另一个缺点是动态性差，添加一条新规则在最坏的情况下需要重建整个数据结构，因而不适合规则频繁变化的流分类器。

2) Linear Search

这种算法采用的数据结构最简单，规则以链表的方式降序存储。分类时数据包从表头开始依次和链表中的各个规则进行比较，直到找到一条匹配的规则或者达到链尾。尽管该算法存储效率高，简单，但是查找时间长，并且查找时间随规则数的增加而线性增加。

3) Cross-Product 算法

基于软件 Cache 的方法，Cache 策略是对过滤规则中各域值进行交叉组合的，预先计算出不同的 cross-product 对应的 ID 放在缓存中。为避免内存爆炸，并不将所有交叉组合的最佳匹配过滤规则存起来，而是固定实际存储的最佳匹配过滤规则的数

量，且动态更新。

5.2.2 WiMAX 中的分类器设计

5.2.2.1 WiMAX 中分类器的性能要求：

1) 占用内存少：分类规则库查找过程中动态分配的空间；
2) 查找速度快：单位时间内处理数据包的数量；
3) 更新效率高：目前 IP 分类问题并不关注。

基站侧的 CS 分类器查找算法，与目前 IP 分类问题类似。目前 IP 分类问题，它主要存在于路由器上，因此对查找速度要求很高，要求以线速度进行，分类速度至少到 1Gbps；其次是对空间复杂度的要求，要求占用内存空间尽可能少；最后才考虑更新效率问题，这是由于在路由器上的分类规则一般都是人工配置的，更新频率很低。

但对于基站侧的 CS 分类器，其性能要求与通常的 IP 分类问题有所不同。一是由于它位于网络边缘，数据流量远小于路由器，查找速度的要求不需要那么高；二是空间复杂度要尽量小，因为一个 BS 上的内存空间目前还不大；三是对更新效率要求比较高，因为对于 CS 分类器来说，由于分类的目的是将数据包对应到相应 CID 上，可能随时都要建立或删除 CID，因此分类规则是经常变动，这一点与 IP 分类问题不同。

5.2.2.2 分类算法描述

首先定义两个名词：规则（rule）和分类器（classifier）。用来对 IP 包进行分类的由包头中若干域组成的集合称之为规则，而若干规则的集合就是分类器。

1) 分类规则表的逻辑组成

规则表中的每条规则包括分类条件，分类器优先级，CID，SFID，Index 五项。其中分类器优先级不一定是唯一的，但要避免二义性；在 0-255 之间取值，数值越大，优先级越高，正常情况下建立的服务流优先级默认值为 128，临时服务流优先级默认值为 0。

分类条件包括以下域：

a）TOS 或是 DSCP（差分服务代码点）

tos-low, tos-high, tos-mask

注：IP 报头的 TOS 字段由 8 位组成，在这 8 位里，前 3 位舍弃不用，最后一位固定是 0，中间的四个位分别来控制封包的：最小延时，最大处理量，最大可靠度和最小花费。这四个位只有一位是 1。如果用十六进制来描述这四种状态的话，分别是 0x10,0x08,0x04,0x02。如果四个位都是 0，则表示正常运行，不做封包的特殊处理。

DSCP 根据 RFC2474 利用 IPv4 包头中的 TOS 字段与 IPv6 中所定义的流标签八

位组。

b）协议

与 IP 报头的协议字段（IPv6 中是下一个头字段）对应，例如 TCP、UDP 等类型。符合 RFC1700。

c) IP 源地址：Src Addr 和 smask。

d) IP 目的地址：Dst Addr 和 dmask。

e) 源端口范围：sportlow 和 sporthigh（若无协议规则绑定，则该端口规则无效）。

f) 目的端口范围：dportlow 和 dporthigh（若无协议规则绑定，该端口规则无效）。

g) IEEE802.3 / 以太网的源 MAC 地址：src 和 mask。

h) IEEE802.3 / 以太网的目的 MAC 地址：dst 和 mask。

i) 网络层协议：IP（0x0800），ARP（0x0806），RARP（0x8035）。

j) IPv6 流标签：n＊3（n 个流标签）。

分类规则表的组成如图 5-4 所示。

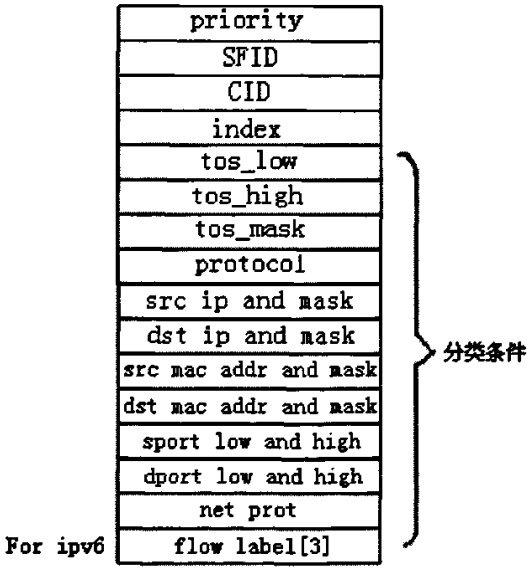

图5-4 分类规则表的组成

2) 分类器设计思想

对于规则的存储，本系统分为存储结构和查寻结构两部分。存储结构为以规则的 index 为索引的 HASH 表，主要用于规则的维护，HASH 函数为((index&oxffc0)>>6)；查寻结构用于快速定位匹配的规则。此外，还有一个专门存储缓存规则的顺序表，该

规则是还未等到服务流建立请求应答的规则。存储规则示意图如图 5-5 所示。

其中，index 的值是根据 dport，sport 和 prot 的不同组合计算得来的。

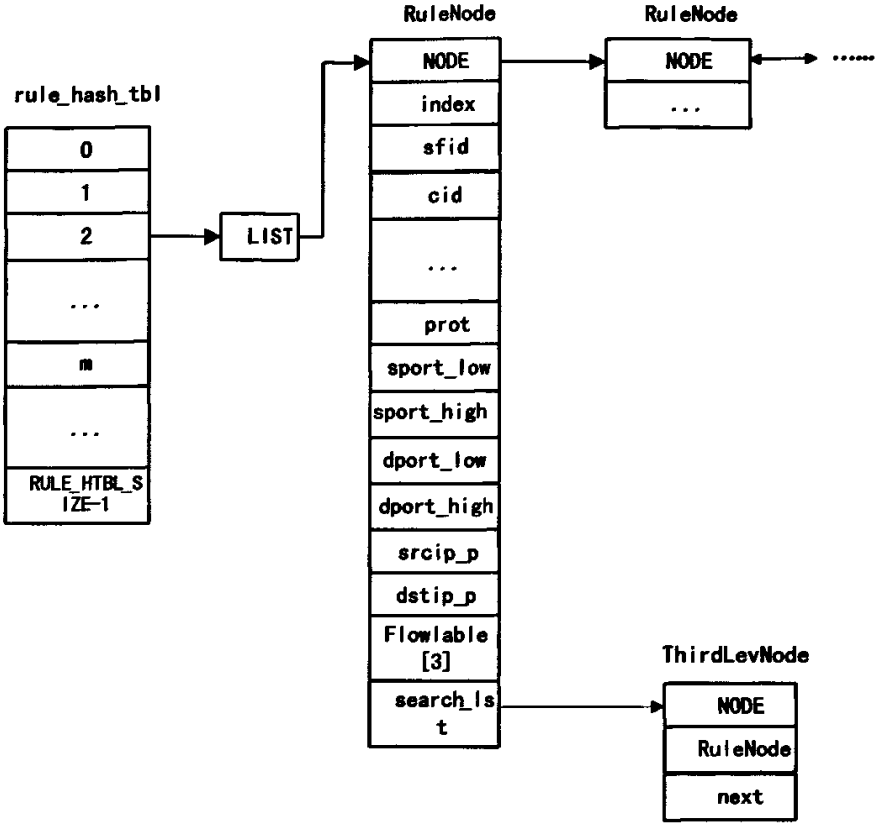

图5-5 分类器的存储结构图

根据本系统中分类器的性能要求，兼顾十个分类条件的常用性，本文设计了多维的分类查寻表。

a) 顶层：是网络层协议，目前分为 IP，ARP，RARP 三类。

b) 第二层：

对于 ARP 和 RARP 来说，直接指向 MAC 源地址和目的地址表（组织见 IP 对应的 MAC 地址表）

对于 IP，指向（协议号 prot，目的端口 dport，源端口 sport）的 hash 表。

首先设置三个集合：PROT_SET {TCP，UDP，ICMP，除 TCP / UDP / ICMP 外的协议，*}，DPORT_SET {80，21，23，小于 1024 的除 80 / 21 / 23 外的端口号，大于 1023 的端口号，*} 和 SPORT_SET {80，21，23，小于 1024 的除 80 / 21 / 23 外的端口号，大于 1023 的端口号，*}，分别对每个集合中的元素进行编号，例如

PROT_SET 中依次为 0，1，2，3，4。三个集合中的元素进行任意组合，共有 5(TRANSPROT_NUM)*6(SPORT_NUM)*6(DPORT_NUM) = 180 种组合，对这 180 种组合进行编号，规则如下：设 PROT_SET 中元素编号为 prot，DPORT_SET 中元素编号为 dport，SPORT_SET 中元素编号为 sport，则组合的编号为 sport* TRANSPROT_NUM* DPORT_NUM+dport* TRANSPROT_NUM+prot。以该规则为元素编号构造跳转表，跳转表的每个入口对应着 dport，prot，sport 的一个组合。并指向下一级查找结构。

c) 第三层：IP 地址，MAC 地址，TOS，流标签

对于由不确定值组成的跳转表项，除需查找 IP 地址，MAC 地址外，还应匹配不确定值所对应的匹配条件。例如跳转表的一项为（小于 1024 的除 80 / 21 / 23 外的端口号，大于 1023 的端口号，TCP），则该项对应的查找结构除 IP 地址，MAC 地址等，还包括源端口号和目的端口号。据统计这种情况仅占不到 10%，因此采用线性查找方式即可。

对于确定的情况，即只需确定 IP 地址或 MAC 地址匹配的情况，首先将查找分为四类，分别为 IPv4，IPv6，MAC（在这儿就假设不存在同时指定 IP 地址和 MAC 都要满足的情况）和通配类，通配类对应未指定任何地址的规则。在每种情况下，可以以 dst_ip 或 dst_mac 为标准划分几个范围（待定），这样可以将查找的范围进一步缩小。

本文采用三层匹配，因此第三层的节点即为查寻结构的叶子节点，该节点并不存储具体的规则，而是存储指向规则的指针。

该查询结构的示意图如图 5-6 所示。

5.2.3 分类器的实现

分类器的实现主要包括 2 部分：一是分类规则表的建立，一个是规则的匹配。在本系统中，分类规则表的建立是在服务流管理模块中完成的；规则的匹配是在分类器模块中完成的。下面，对这两个部分进行描述，其他模块的功能在下面的章节中介绍。

1) 分类规则表的建立

当 BS 接收到上层应用的服务流请求后，将该请求的规则 TLV 字段缓存，向通信对端发起服务流建立请求。当得到 MAC_CREATE_SERVICE FLOW.confirmation，判断建立服务流成功后，就根据缓存的 TLV 建立规则。

服务流管理模块的功能是维护分类规则表，提供基本的规则表维护函数；保存分类器规则表；对所有服务流管理消息进行响应；对涉及 IOCTL 的服务流管理消息进行处理。

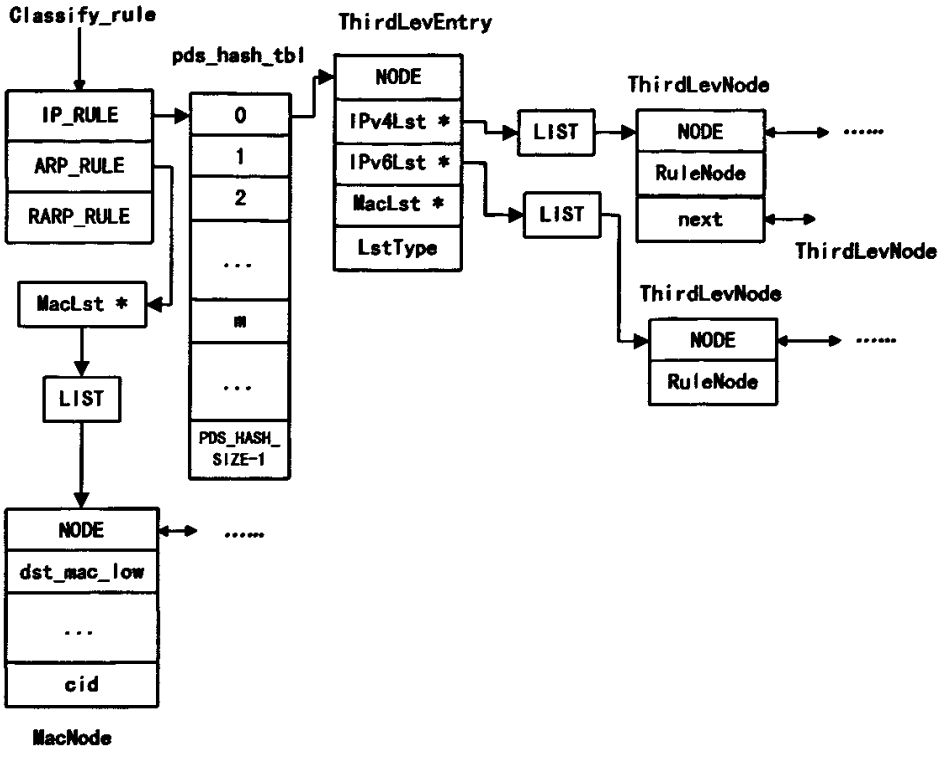

图5-6 分类器的查询结构示意图

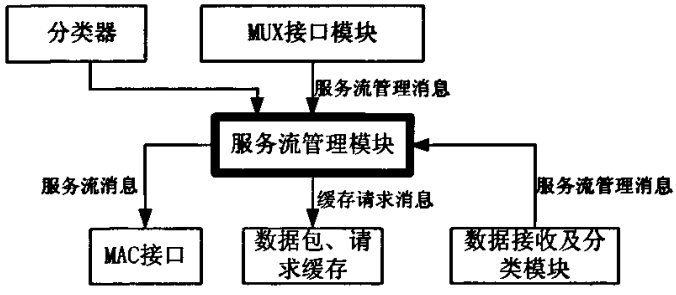

图 5-7 服务流管理模块与其他模块的交互关系

如果到达的一个包没有匹配到任何的服务流 CID，分类器模块就会调用服务流管理的函数发起一个建立默认参数的临时的服务流的过程，此时建立对应的规则是缓存规则，等收到 confirmation 消息后这些缓存规则改为临时规则，并设置此规则的 used 字段为 1，设置 timeout 字段，该字段表示该规则已超时的时间。

服务流管理模块启动一个定时器，定时扫描临时规则，如果一个规则的 used 字段是 0，则 timeout 字段减 1，如果 timeout 字段小于等于 0 则发起删除流程删除这个

服务流和规则；如果一个规则的 used 字段是 1，则设置 timeout 字段为初始值。

服务流管理模块与其他模块的交互如图 5-7 所示。

服务流管理模块的实现主要分为两个部分：

a)，规则表的建立/修改/删除

当 BS 接收到上层应用的服务流请求后，将该请求的规则 TLV 字段缓存，向通信对端发起服务流建立/修改/删除请求。当得到 MAC_CREATE/CHANGE/DELETE_SERVICE FLOW.confirmation，判断建立/修改/删除服务流成功后，就根据缓存的 TLV 建立/修改/删除规则。

b)， 对规则表的维护

服务流管理模块对规则表进行定期扫描，当发现有超期的规则表时，则对其进行相应的处理，或删除，或修改。

处理流程如图 5-8 所示。

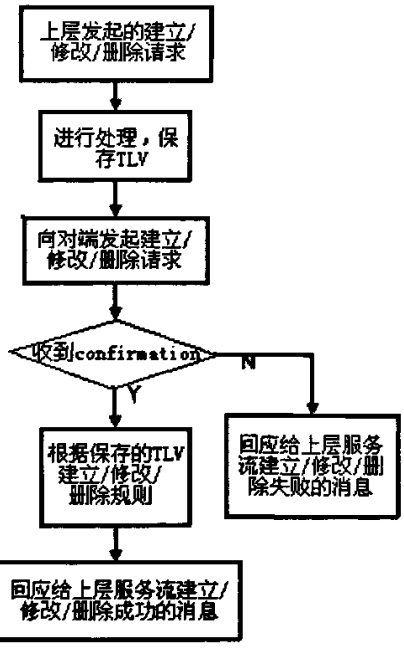

图 5-8 服务流管理模块对规则表的维护流程

2) 规则匹配

当上层有数据发送时，需要与分类规则进行匹配（对于 DHCP 等协议需要特殊的流程），将数据映射到相应的 CID 进行发送（通过 MAC 接口模块）。如果没有相匹配的服务流，则根据下一跳的 IP 地址从地址转换表中查询其对应 primarycid，在 primarycid 上向 MAC 层发送 MAC_CREATE_SERVICEFLOW.request 原语建立服务流

(参数取默认参数)，通知服务流管理模块建立临时规则,并将该数据包和请求缓存到数据、请求缓存表中。如果在地址转换表中没有查到相应的 primarycid 则丢弃该包。

本模块在系统中与其他模块的交互关系如图 5-9 所示：

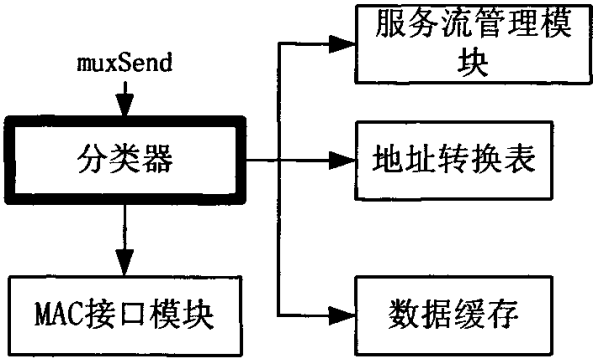

图 5-9 分类器模块与其他模块的交互关系

上层发送的数据报，一般从 transportcid 发送，因此要在规则表中进行匹配，查找到对应的 transportcid；但还有一些特殊情况。802.16 协议规定，基于协议的消息（例如 DHCP,SNMP 等）从 SecondaryCID 发送，本系统中规定 DHCP，time protocol 和 TFTP 从 SecondaryCID 发送。所以 CS 层在进行规则表匹配之前，首先解析数据包，看是否是这类消息，若是则直接去地址转换表根据目的 IP 或 MAC 地址查找对应的 SecondaryCID，若不存在则发送广播消息询问目的 MAC 或目的 IP 地址对应 SecondaryCID，得到应答后在该 SecondaryCID 上发送数据包，否则将包丢弃。

若不是系统规定的从 SecondaryCID 上发送的数据包，则进行规则表的查询。找到匹配规则对应的 CID 后，在该 CID 上发送数据；若找到未对应 CID 的临时规则，则直接将数据缓存；若未找到任何规则，对于 IP 数据包，则从地址转换表中查询目的 IP 对应的 primarycid，在该 primarycid 发送服务流建立请求以建立缓存规则，若未找到 primarycid，则进行与查找 secondarycid 相同的工作，发送广播消息获取 primarycid，并暂时将数据缓存。该处理的过程如图 5-10 所示。

5.2.4 分类器的关键数据结构

1) 存储结构

分类器的存储数据结构如下所列，存储结构分为规则表和缓存的规则表。

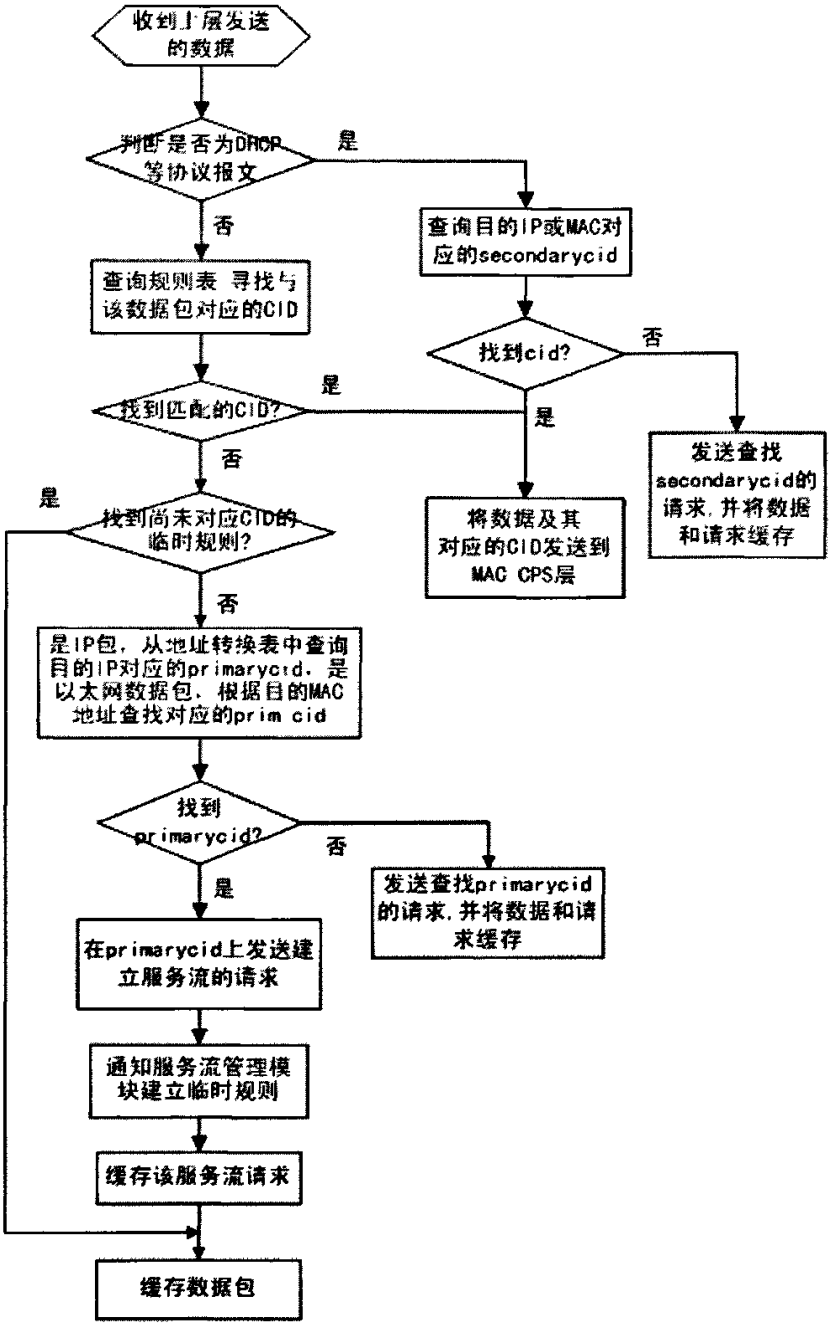

图5-10 分类器进行数据报处理的过程

a) 规则表的存储结构

#define RULE_HTBL_SIZE 1024

```
    LIST *rule_hash_tbl[RULE_HTBL_SIZE]; /*顺序存储的 HASH 表*/
typedef struct{
    NODE ln;
    UINT16 cid;
    UINT16 index;
    UINT32 sfid;
    UINT8 type:3; /*规则类型，在上面枚举类型中定义*/
    UINT8 used:1; /*在一个时间单位内是否已用*/
    UINT8 version:1; /*ipv4 or ipv6*/
    UINT8 resv:3;
    UINT8 timeout;/*该规则已经存在的时间*/
    UINT8 priority;
    UINT8 tos_low;
    UINT8 tos_high;
    UINT8 tos_mask;
    UINT16 netprot;
    UINT8 prot;  /*协议条件表*/
    UINT16 sport_low; /*源端口条件表*/
    UINT16 sport_high;
    UINT16 dport_low;  /*目的端口条件表*/
    UINT16 dport_high;
    void srcip_p; /*源 IP 地址条件表*/
    void   dstip_p /*目的 IP 地址条件表*/
    UINT8 flowlabel[3];  /*流标签条件表*/
    Struct ThirdLevNode *search_lst; /*指向在查找结构中与该规则对应的节点队列*/
}RuleNode; /*规则节点，其中 LIST 类型的项若不存在则 LIST 指向的节点为 NULL，
                而 tos，netprot 项不存在则全设为 0*/
```

b) 缓存规则表的存储结构

```
LIST cacherule_tbl; /*缓存规则表，是顺次存取的*/
typedef struct{
    NODE ln;
    UINT32 seqnum;
```

```
    UINT8 version; /*ipv4 or ipv6*/
    void dstip_p; /*目的 IP 地址条件表*/
}CacheRuleNode; /*缓存规则节点*/
```

2) 查询结构

分类器查询结构的数据结构如下。

a) 顶层

```
#define NETPROT_NUM 3; //网络层协议个数
enum{
    IP_RULE;    //IP
    ARP_RULE;   //ARP
    RARP_RULE;  //RARP
}  //网络层协议类型
void *classify_rule[NETPROT_NUM];  //分类表的入口，以网络层协议类型划分
```

b) 第二层

```
#define TRANSPROT_NUM 5; //协议种类
#define DPORT_NUM 6; //目的端口种类
#define SPROT_NUM 6; //源端口种类
#define   PDS_HASH_SIZE   TRANSPROT_NUM*DPORT_NUM*   SPROT_NUM;
                                    //协议 / 端口号的 HASH 表大小
UINT8 prot_set[ ] = {6,17,1}; //TCP:6   UDP:17   ICMP:1"除 TCP / UDP / ICMP 外的
                                协议"为最大 index+1，*为最大 index+2
UINT16 dport_set[ ] = {80,21,23} //小于 1024 的除 80 / 21 / 23 外的端口号为+1，//大
                                    于 1023 的端口号为+2，*为+3
UINT16 sport_set[ ] = {80,21,23} //同 dport_set
ThirdLevEntry *pds_hash_tbl[PDS_HASH_SIZE];  //对应第三层的入口
```

c) 第三层

```
typedef struct{
    LstType lst_type;  /*标志第二层的组合类别，例如含有不确定的端口号，这是由
                        于若某项不确定，第三层的节点需要将该项列出来*/
    IPv4Lst *ipv4_lst;
    IPv6Lst *ipv6_lst;
    MacLst *mac_lst;
```

```
    WildLst *wild_lst; /*未指定地址的规则表*/
}ThirdLevEntry;
typedef UINT8 LstType;  /*第二层组合类别，标识 sport，dport 和 prot 三项中哪项还
                          不是确定的值，具体值见下面的枚举定义*/
enum{
    EXACT,  /*sport，dport 和 prot 都是确定的值*/
    SPORT,  /*sport 还未确定，即值为小于 1024 的除 80/21/23 外的端口号等*/
    DPORT,  /*dport 还未确定*/
    PROT,   /*prot 还未确定*/
    SD,     /*sport 和 dport 都未确定*/
    SP,     /*sport 和 prot 都未确定*/
    DP,     /*dport 和 prot 都未确定*/
    SDP     /*dport，sport 和 prot 都未确定*/
}
struct ThirdLevNode {
    NODE ln;
    RuleNode *rule;
    Struct ThirdLevNode *next;/*在与存储结构相关联的队列中,指向下一个节点*/
}; /*第三层节点，IPv4Lst \IPv6Lst\WildLst 所包含的 NODE 都为该结构 */
```

5.3 会聚子层其他模块的设计和实现

5.3.1 MUX 接口模块

1) 模块功能

主要完成与 VxWorks 中 MUX 层的交互，作为驱动程序和网络协议栈的接口。主要功能是装载、启动设备，缓冲区管理，接收、发送数据接口，IP 地址和 MAC 地址转换。把应用程序的一系列原语转化为 CS 驱动程序的动作，这些原语包括两类：服务流管理原语、数据原语。

本模块与系统中其他模块的交互关系如图 5-11 所示。

2) 模块实现

模块为一组按 VxWorks MUX 接口规范实现的函数。完成驱动程序的接口。

a) WIMAXLoad 初始调用,初始化 END 结构，调用 WIMAXMemInit 初始花

MBLK 内存池；

b) WIMAXStart 启动 16 CS 和 MAC 层程序，调用各模块接口函数；

c) WIMAXRecv 被 CS 层接收分类模块调用，将接收到的 IP 分组，转化成 MBLK 结构，传递给 MUX 接口；

d) WIMAXSend 被上层 MUX 接口调用，将 IP 分组交给 CS 层；

e) WIMAXIoctl 对上层的 IOCTL 操作响应。应用层通过 IOCTL 操作，要求建立、修改、删除服务流。

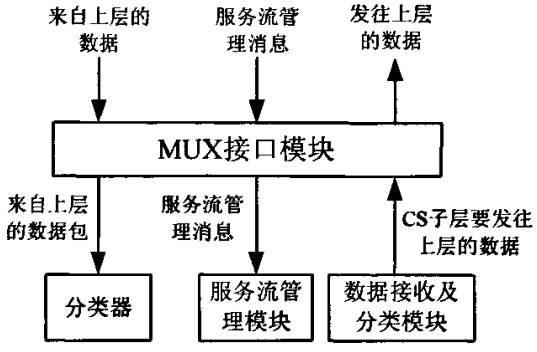

图5-11 MUX接口模块与其他模块的接口

5.3.2 地址转换模块

1) 模块功能

地址转换模块提供以下映射关系的生成和查寻：

a) 目的 IP 地址和 primary cid 之间的映射

b) 目的 IP 地址和 Second CID 之间的映射

c) 目的 MAC 地址和 Second CID 之间的映射

d) 目的 MAC 地址和 primary cid 之间的映射（对应 Ethernet）

分类器模块就是通过地址转换模块来得到以上的各种映射。

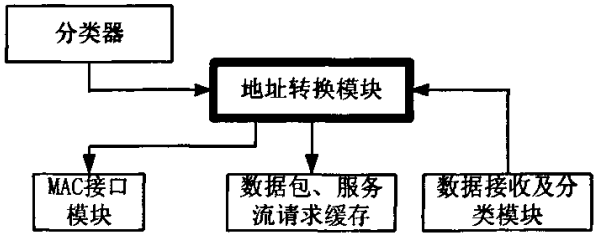

图5-12 地址转换模块与其他模块的交互关系

以上映射关系的生成是采用类 NDP 机制（本系统中称之为 WiMax-NDP），该机制对没有映射的 IP 或者 MAC 地址利用广播消息在 WiMax-NDP-CID 上进行广播查询

其 primary CID 或者 Secondary CID，SS 端 CS 层的地址转换模块进行响应，返回对应关系。基站 CS 层对地址应答消息进行处理。

此外，对于本模块维护的地址映射关系，进行定时的删除和更新。

本模块与系统中其他模块的交互关系如图 5-12 所示。

2) 模块实现

WiMax-NDP 机制的核心就是生成和维护终端 IP-CID 信息缓存表和终端 MAC-CID 信息缓存表，如表 5-1、表 5-2 所示。

表 5-1 终端 IP-CID 信息缓存表

表项	含义
Id	表项编号
Dst_ip	终端 IP 地址
Version	ipv4(0) or ipv6(1)
Type	标志 CID 的类型，0 为 primarycid，1 为 secondarycid
cid	IP 地址对应的 Primarycid 或 secondarycid
Lifetime	该表项的生存期
Used	是否正在用

表 5-2 终端 MAC-CID 信息缓存表

表项	含义
Id	表项编号
dst_mac	终端 MAC 地址
type	标志 CID 的类型，0 为 prim cid，1 为 second cid
sec_cid	MAC 地址对应的 secondarycid
Lifetime	该表项的生存期
Used	是否正在用

a) 生成机制：

本系统中生成机制是 on-demand 的，当上层需要获得 IP 地址或 MAC 地址对应的 cid 时，若在 CID 缓存表中该映射不存在，则启动生成过程：

首先检查已经发送但尚未收到回复的请求，是否与当前请求相同，若相同，则不重复发送 CID 请求，直接将该数据插入数据包缓存中。

若不存在相同的请求，则 BS 通过广播方式发送 CID 请求，然后将请求和需要该 cid 的数据包缓存，若得到 SS 方的 CID 通告，则保存该 cid 映射，并根据 cid 处理数据包。若在一定时间内未收到 cid 通告，则重发数据包。重发 3 次仍未得到通告，将数据包丢弃。

CID 请求和通告均为 MAC 管理消息，类型分别为 129 和 130。CID 请求为广播消息，在广播 CID 上传输；CID 应答为单播消息，在 PrimaryCID 上传输。消息的定义如下：

CID-REQ 消息：

CID-REQ_Message_Format{

Management Message Type = 129; /*8bit*/

CIDType; /*8bit，0 为 primary，1 为 secondary*/

SeqNum; /*32bit*/

TLV Encoded Information;

}

TLV 中包括的内容如表 5-3 所示。

表 5-3 TLV 中包括的内容

Type	Length	Value
0	4	IPv4 地址
1	16	IPv6 地址
2	6	MAC 地址

CID-RSP 消息：

CID-RSP_Message_Format{

Management Message Type = 130; /*8bit*/

CIDType; /*8bit*/

SeqNum; /*32bit*/

CID; /*16bit*/

TLV Encoded Information;

}

其中 SeqNum，TLV Encoded Information 与 CID-REQ 里的相同。

b) 维护机制：

定时扫描终端IP-CID信息缓存表和终端MAC-CID信息缓存表，lifetime的单位为扫描周期。扫描周期为5秒，则每扫描1次将lifetime减1，当lifetime为0时将used变为0，下次扫描若used仍为0则将该表项删除。每次查询得到某一匹配表项时，将used改为1，并将lifetime置初始值，即重新开始计时。

5.3.3 缓存模块

1) 模块功能

a) 数据、请求缓存表的建立

对于上层来的数据包，需要与分类规则进行匹配。如果没有相匹配的服务流，则

根据包的 IP 地址调用地址转换模块中的函数查询 Primary CID，向 MAC 层发送 MAC_CREATE_SERVICEFLOW.request 原语发起服务流建立过程。此时把该请求消息和相应的数据包缓存到数据、请求缓存表中，用请求消息的序列号 seq_num 做为索引。

b) 缓存消息的处理

为每一个缓存的请求消息设立一个定时器，如果超时或者有回应消息到达，则设置标志位标志超时或者是有回应消息到达，然后调用缓存结构中的回调函数进行处理。

对于没有映射到服务流的数据包和服务流管理模块中的缓存规则应该和缓存的请求消息之间有指针联系。以便在缓存规则到期后可以删除建立的缓存规则和相应的数据包，以及把后续的对应于此缓存规则的数据包链入请求消息缓存结构中。

本模块在系统中与其他模块的交互关系如下图 5-13 所示：

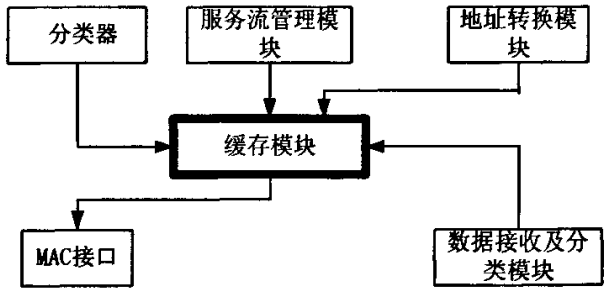

图 5-13 缓存模块与其他模块的交互关系

2) 模块设计

本模块的作用主要是对有缓存要求的包进行缓存，处理流程如图 5-14 所示。

在模块中，首先调用建立缓存结构函数，并将得到的缓存结构实例存放到哈希表中，接着启动定时器，然后调用插入数据包到包链表函数和建立回调结构实例函数，将数据包和回调结构实例分别插入到对应缓存结构实例的数据包链表和回调链表中。当定时时间到，就调用其关联函数进行处理，收到回应消息时，就调用回调函数，最后调用删除缓存结构实例函数把该缓存结构实例从哈希表中删除。

5.3.4 MAC 接口模块

1) 模块功能

将其他模块发来的包封装成合适的格式，然后发送到 MAC 程序。需要发送的消息主要包括两类：服务流管理消息和数据消息。主要有从分类器传来的服务流数据和服务流申请消息；从服务流管理模块传来的服务流删除、修改消息；从数据接收及分

类模块发来的缓存消息。

MAC 接口模块与其他模块的交互关系如图 5-15 所示。

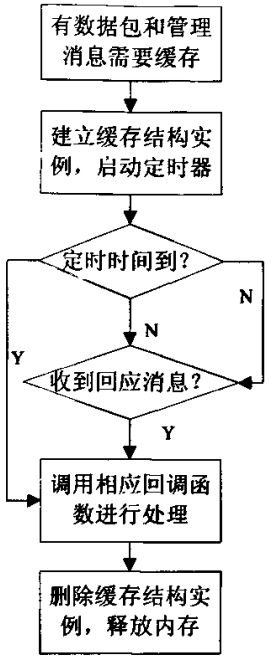

图5-14 缓存模块对缓存包的处理

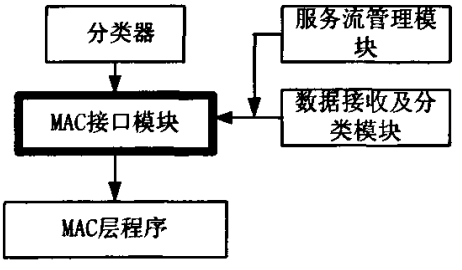

图 5-15 MAC 接口模块与其他模块的交互关系

2) 模块设计

本模块主要的作用是为其他模块提供接口函数，将要发送的数据和服务流消息发送到和 MAC CPS 交互的相应的队列中，在模块内部，不对接收到的数据和服务流消息进行处理，因此，模块的实现比较简单，是由几个接口函数来实现的。

5.3.5 数据接收及分类模块

1) 模块功能

该模块作为一个任务运行，接收来自 MAC 层 SDU 输入输出模块和服务流管理模

块发来的消息，并对消息进行判断。主要包括两类消息：服务流管理消息、数据消息。数据消息通过 mux 接口传到协议栈上层。如果是服务流管理消息则进行解析并更新服务流管理模块数据。如果是 confirmation 消息，则对缓存表中的请求消息进行处理。

数据接收及分类模块在系统中与其他模块的交互如图 5-16 所示。

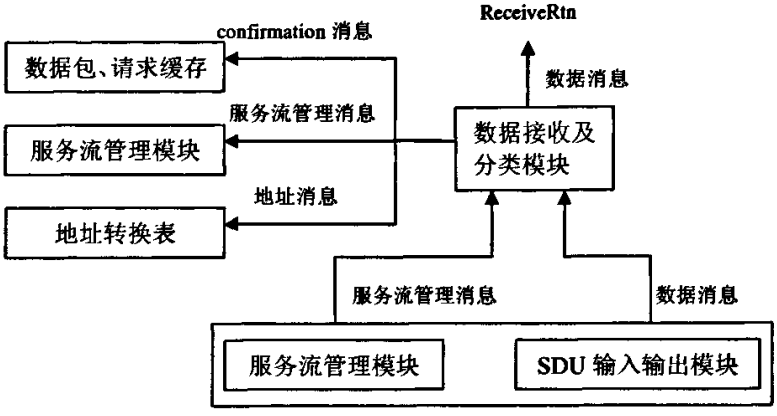

图 5-16 数据接收及分类模块与其他模块的交互

2) 模块实现

该模块是和 MAC CPS 子层交互的模块，主要是从交互的消息队列中读取 MAC CPS 子层传给 CS 子层的数据和消息，根据消息的类型，通知相应的模块进行处理。接收及分类模块的处理流程如图 5-17 所示。

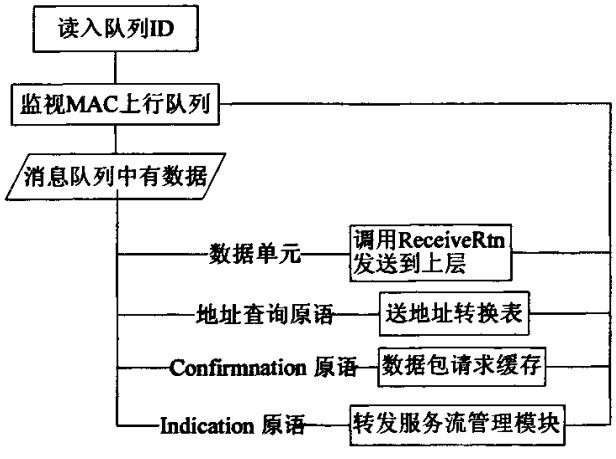

图5-17 数据接收及分类模块的处理流程

5.4 本章小结

本章介绍了一种 802.16 CS 子层分类器的设计和实现方法。802.16 CS 子层分为 7 个模块：MUX 接口模块，分类器模块，服务流管理模块，地址转换模块，数据接收及分类模块，缓存模块，MAC 接口模块。本文中给出了 CS 子层分类器的设计框架，模块划分，模块接口定义和数据结构定义。该方案被应用在 802.16 无线城域网话音通信系统的基站设备软件设计和实现中。

第六章 MAC 层 ARQ 模块的设计与实现

由于 WiMAX 的定位是基于无线模式进行“最后一公里”的连接，因此其 MAC 层的主要功能就是高效地对无线链路进行管理。IEEE 802.16 的 MAC 层支持两种网络拓扑方式：点对多点(PMP)和网状网(Mesh)。在本系统中，只支持 PMP 模式的。

IEEE 802.16 MAC 协议是面向连接的，当用户站被激活进入网络，会与基站建立一个或多个用于数据传输的连接。MAC 层对无线资源的使用进行调度并根据业务不同提供 QoS 保证，通过采用链路自适应技术和自适应重传(ARQ)技术以提供比较高的频谱效率。

ARQ 指自动重传请求，也指循环自动请求。它是指一种通信功能，指由于接收机检测到了差错而要求发送机将信息组或帧重发一次。WiMAX 技术在链路层加入了 ARQ 机制，减少到达网络层的信息差错，可大大提高系统的业务吞吐量。

6.1 ARQ 简介

ARQ 的作用原则是对出错的数据帧自动重发，它有三种形式：停等协议 ARQ、连续 ARQ 和选择重传 ARQ。

6.1.1 停等协议 ARQ

停止等待 ARQ 协议是指发送端发送一帧数据，等待接收端的确认帧，如果在定时时间内收到了确认帧，则发送下一帧数据，如果在定时时间内没有收到确认帧，则重发该数据帧。在发送端每发送完一帧数据后都要设置该帧的超时计时器[8]。

停止等待 ARQ 协议的工作原理如图 6-1 所示。

说明：

1) 结点 A 发送一个数据帧后，必须等到结点 B 的确认帧才可以发送下一个数据帧（图 a）；
2) 在结点 B 接收错误时，结点 B 发一否认帧(NAK)，要求结点 A 重发该帧（图 b）；
3) 为防止发送的数据丢失，结点 A 内部设置一个定时器。结点 A 发送完一个数据帧时，就启动一个超时计时器(timeout timer)（计时器又称为定时器）。若到了超时计时器所设置的重传时间 t_{out}（一般可将重传时间选为略大于“从发完数据帧到收到确认帧所需的平均时间”）而仍收不到结点 B 的任何确认帧，则结点 A

就重传前面所发送的这一数据帧（图 c）和图 d））。

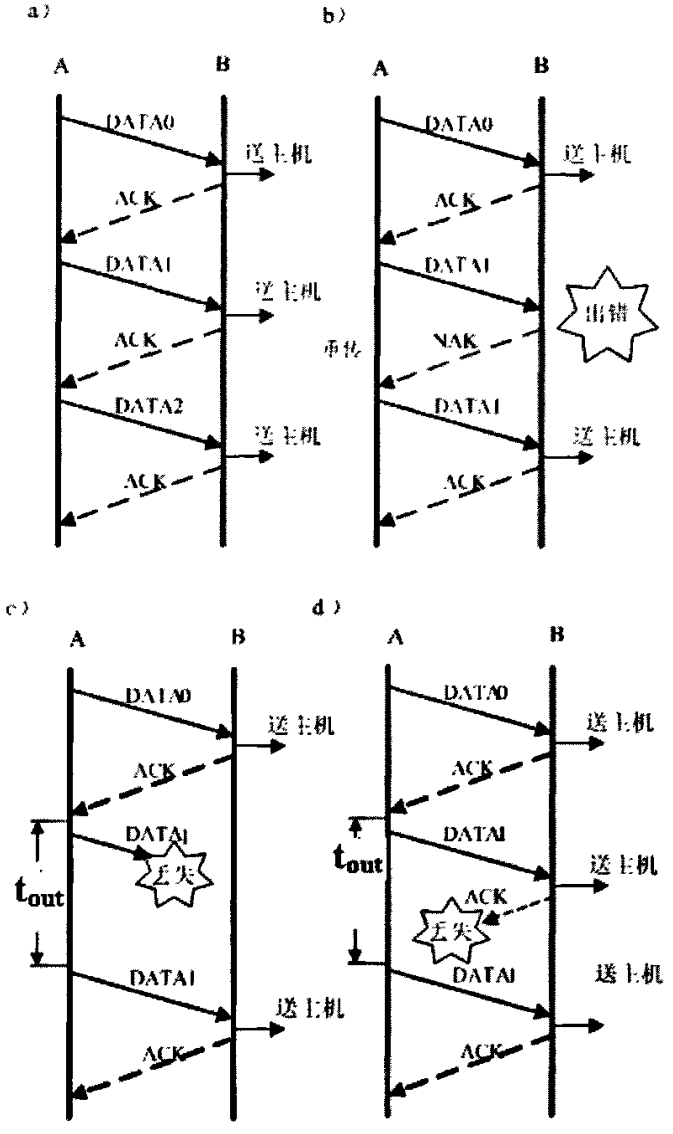

图 6-1 停止等待 ARQ

a) 正常情况 b) 数据帧出错 c) 数据帧丢失 d) 应答帧丢失

4) 为防止确认帧丢失而造成发方重发同一数据帧，发方给每一个数据帧带上一个序列号。使每一个数据帧带上不同的发送序号。每发送一个新的数据帧就把它的发送序号加 1。

a) 任何一个编号系统的序号所占用的比特数一定是有限的。因此，经过一段时

间后，发送序号就会重复。

b) 序号占用的比特数越少，数据传输的额外开销就越小。

c) 对于停止等待协议，由于每发送一个数据帧就停止等待，因此用一个比特来编号就够了。

d) 一个比特可表示 0 和 1 两种不同的序号。

5) 若结点 B 收到发送序号相同的数据帧，就表明出现了重复帧。这时应丢弃重复帧，因为已经收到过同样的数据帧并且也交给了主机 B。但此时结点 B 还必须向结点 A 发送确认帧（ACK），因为结点 B 已经知道结点 A 还没有收到上一次发过去的确认帧 ACK（图 d））。

6.1.2 连续 ARQ

工作原理

连续 ARQ 是允许发送方不等确认帧返回就连续发送多个数据帧。接收端只按序接收数据帧，不按序号到来的数据帧被丢弃。确认帧中包含着期望下次收到的帧的序号。在发送端发送完一帧后都要设置该帧的超时计时器。

连续 ARQ 协议的工作原理如图 6-2 所示。

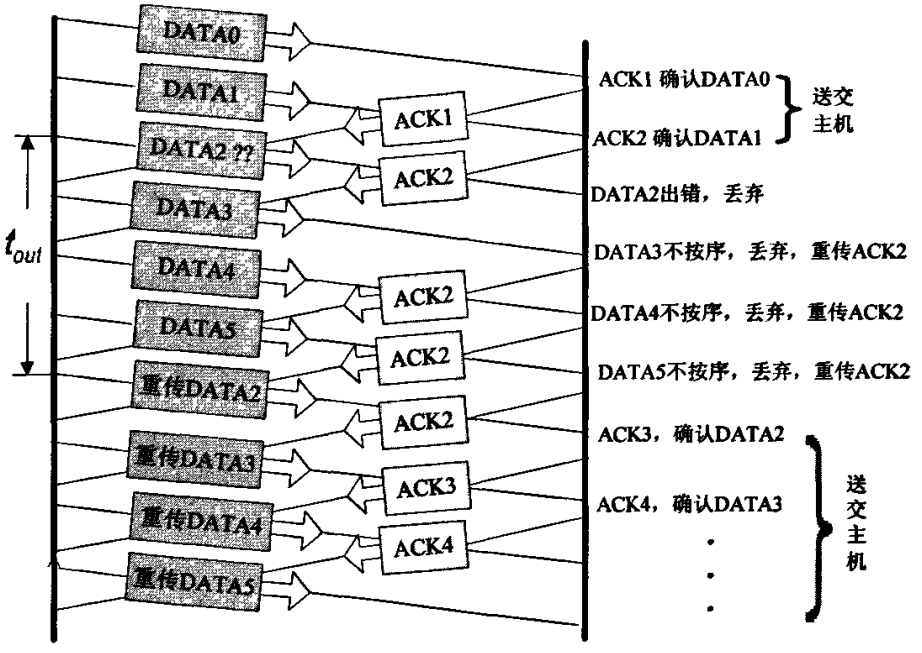

图 6-2 连续 ARQ 协议的工作原理

说明：

a) 接收端只按序接收数据帧。虽然在有差错的 2 号帧之后接着又收到了正确

的 3 个数据帧，但接收端都必须将这些帧丢弃，因为在这些帧前面有一个 2 号帧还没有收到。虽然丢弃了这些不按序的无差错帧，但应重复发送已发送过的最后一个确认帧（防止确认帧丢失）。

b) ACK1 表示确认 0 号帧 DATA0，并期望下次收到 1 号帧；ACK2 表示确认 1 号帧 DATA1，并期望下次收到 2 号帧；依此类推。

c) 发送端在每发送完一个数据帧时都要设置该帧的超时计时器。如果在所设置的超时时间内收到确认帧，就立即将超时计时器清零。但若在所设置的超时时间到而未收到确认帧，就要重传相应的数据帧（需重新设置超时计时器）。

d) 在等不到 2 号帧的确认而重传 2 号数据帧时，虽然发送端已经发完了 5 号帧，但仍必须向回走，将 2 号帧及其以后的各帧全部进行重传，因此连续 ARQ 又称为 Go-back-N ARQ，意思是当出现差错必须重传时，要向回走 N 个帧，然后再开始重传。

滑动窗口的概念

在连续 ARQ 协议中，发送方和接收方通过滑动窗口机制实现流量控制。

发送端设定发送窗口：发送窗口用来对发送端进行流量控制。发送窗口的大小 W_t 代表在还没有收到对方确认信息的情况下发送端最多可以发送多少个数据帧。

接收端设置接收窗口：在接收端只有当收到的数据帧的发送序号落入接收窗口内才允许将该数据帧收下。若收到的数据帧落在接收窗口外，则一律丢弃。在连续 ARQ 中，接收窗口大小 $W_r=1$。

i. 只有当收到的帧的序号与接收窗口一致时才能接收该帧。否则，就丢弃它。

ii. 每收到一个序号正确的帧，接收窗口就向前（即向右方）滑动一个帧的位置。同时发送对该帧的确认。

滑动窗口的示例如图 6-3 所示。

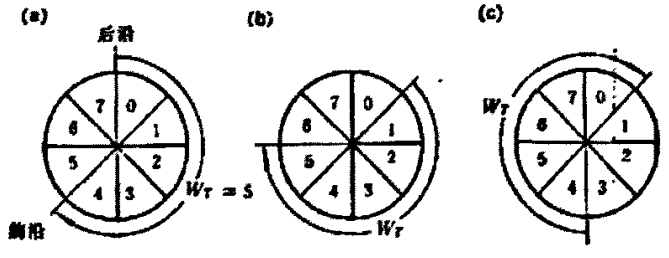

图 6-3 发送端的滑动窗口示例图

（a）允许发送 0 号至 4 号帧（b）允许发送 1 号至 5 号帧（c）允许发送 4 号至 0 号帧

设发送序号用 3 个比特来编码，即从 0 号到 7 号（共 8 个不同的序号）。又设发送窗口 $W_t = 5$，即在未收到对方确认信息的情况下，发送端最多可以发送出 5 个数据帧。在图 6-3(a)中,画出了刚开始发送时的情况。这时，在扇形的发送窗口内（即在窗口前沿和后沿之间）共有 5 个序号，从 0 号到 4 号，具有这些序号的数据帧就是发送端现在可以发送的帧。若发送端发完了这 5 个帧（从 0 号帧到 4 号帧）但仍未收到确认信息，则由于发送窗口已填满，就必须停止发送而进入等待状态。当 0 号帧的确认信息收到后，发送窗口就沿顺时针方向旋转 1 个号，使窗口后沿再次与一个未被确认的帧号相邻（图(b)）。这时 5 号帧的位置落入了发送窗口之内，因此，发送端现在就可以发送这个 5 号帧，以后设又有 3 个号的确认帧到达发送端，于是发送端再沿顺时针方向向前旋转 3 个号（图(c)），而发送端继续可以发送的数据帧的发送序号是 6 号，7 号，0 号。

当用 n 个比特进行编号时，若接收窗口的大小为 1，则只有在发送窗口的大小 $W_t \leq 2n-1$ 时，连续 ARQ 协议才能正确运行。

例如，当采用 3 bit 编码时，发送窗口的最大值是 7 而不是 8。

为了减少开销，连续 ARQ 协议还规定接收端不一定每收到一个正确的数据帧就比须发回一个确认帧，而是可以在连续收到好几个正确的数据帧以后，才对最后一个数据帧发确认信息。这就是说，对某一数据帧的确认就表明该数据帧和这以前所有的数据帧均已经正确无误的收到了。这样做可以使接收端少发一些确认帧，因而减少了开销。

6.1.3 选择重传 ARQ 协议

为了进一步提高信道的利用率，可以设法只重传出现差错的数据帧或者是定时器超时的数据帧。此时，必须加大接收窗口，以便先收下发送序号不连续但仍处在接收窗口中的那些数据帧。等到所缺序号的数据帧收到之后再一并送交主机。这就是选择重传 ARQ 协议。

选择重传 ARQ 的工作原理如图 6-4 所示[8]。

图 6-4 画的是选择重传 ARQ 协议的示意图。假设接收窗口 $W_R = 4$，再假定 6 号数据帧在传送时丢失，按照选择重传 ARQ 协议，接着传送的 7—8 号数据帧在接收端不是被丢弃，而是先暂存一下。等到 6 号数据帧由于超时定时器时间到而重传并到达接收端时，接收端再按数据帧的序号顺序交付给主机。这样做可避免重复传送那些本来已经正确到达接收端的数据帧；但是付出的代价是在接收端要设置具有一定容量的缓存空间。

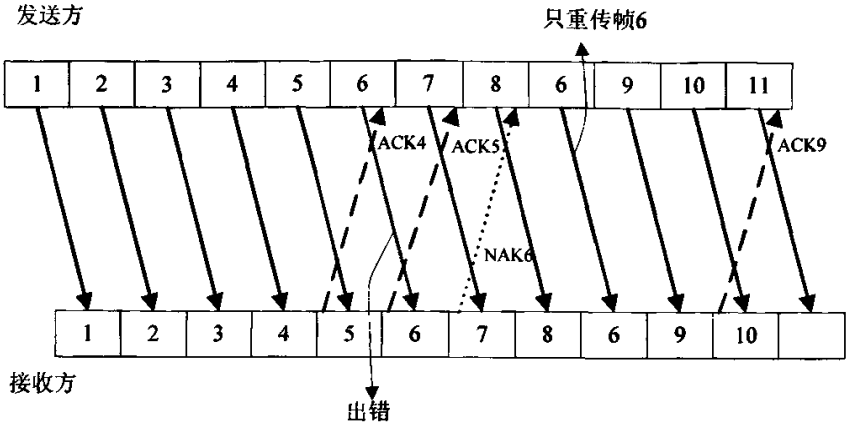

图 6-4 选择重传 ARQ 的工作原理

6.1.4 三种 ARQ 的比较

对三种 ARQ 协议的优缺点的比较如表 6-1 所示。

表 6-1 三种 ARQ 协议的比较

协议类型	信道利用率	实现难易程度
停等 ARQ 协议	信道利用率最低。	实现最简单。
连续 ARQ 协议	信道利用率在三种 ARQ 协议中居中。	实现上在三种 ARQ 协议中居中。
选择重传 ARQ	信道利用率最高。	实现上要求接收端的缓冲容量最大。

6.2 无线城域网中 ARQ 的设计

在 802.16 系统中，ARQ 机制是在每个连接上激活的，一个连接的 ARQ 应该在连接建立的时候指定和协商。一个连接不允许是 ARQ 和 non-ARQ 的混合，即一个连接在建立后，要么是支持 ARQ 的，要么是不支持 ARQ 的，不允许出现对一个 SDU 支持 ARQ，而对另一个 SDU 不支持的情况。对于 ARQ 允许的连接，是否支持分段是可选的。假如支持分段，则发送端将每个 SDU 分成几段，来进行分别传输，分段的参数是 ARQ_BLOCK_SIZE（ARQ_BLOCK_SIZE 是用于将 SDU 在发送之前分成的一系列 ARQ 块的长度，该值的大小是在连接建立的时候，BS 和 SS 协商确定的）的值。当不支持分段时，不管协商的 ARQ 块的值是多少，每个用于传输的段都应该是包括整个 SDU 的所有的数据块。此处的分段和 PDU 中的分段是不同的。

ARQ 的反馈消息可以是作为独立的 MAC 管理消息，在适当的 basic 管理连接上

发送，也可以被打包成一个 PDU，在一个已经存在的连接发送。ARQ 反馈消息不能被分段。

6.2.1 ARQ 设计中的几个关键名词

ARQ 块

一个 MAC SDU 逻辑上应该被分成几个 ARQ 块，块的长度是由连接的 TLV 参数 ARQ_BLOCK_SIZE 指定的。当 SDU 的长度不是块的整数倍时，该 SDU 的最后一块是由最后一个完整的块所剩余的 SDU 的字节形成的。一旦一个 SDU 被分成了一组 ARQ 块，在所有的 ARQ 块被正确的传到接收端之前或者 SDU 被发送状态机丢弃之前，这种划分机制仍旧是有效的。

用于发送的或者是重传的一系列 ARQ 块被封装成一个 PDU。一个 PDU 中可能包括第一次发送的块，也可能包括这些块的重传块。PDU 的分段应该仅仅在 ARQ 块的边界处进行。一组 ARQ 块作为一个 PDU 传输的时候是否需要在重传的时候仍然作为一个 PDU 是由发送端所决定的。图 6-5 表明了使用块进行 ARQ 传输和重传的情况；重传有两种选择：重新排列 ARQ 块或者是不重新排列。

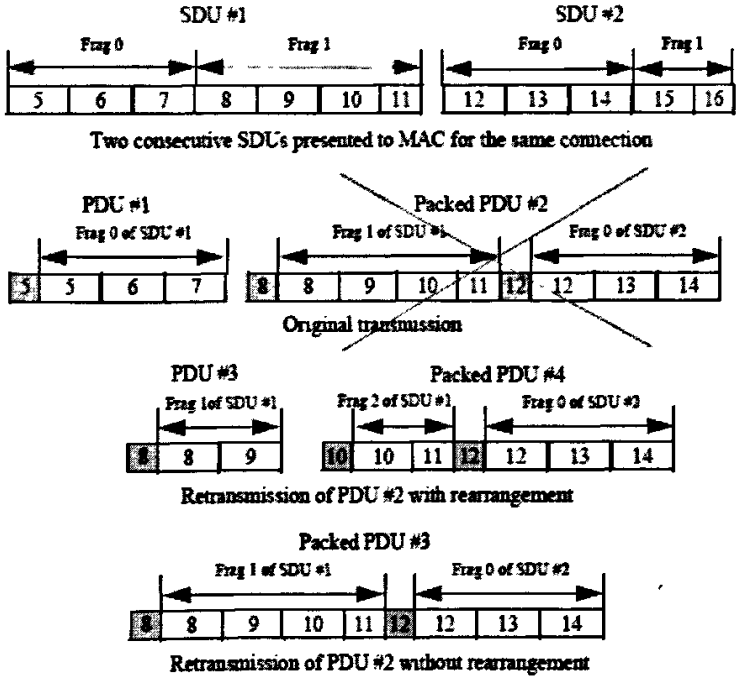

图 6-5 ARQ 传输和重传的情况

ARQ 的几个关键参数值

(1) BSN

ARQ 块的序列号。11bit，最大值为 2^{11}。

(2) ARQ_BSN_MODULUS

ARQ_BSN_MODULUS 等于 BSN 的特殊值。例如：2^{11} 。

(3) ARQ_WINDOW_SIZE

ARQ_WINDOW_SIZE 是由发送端和接收端处理的在滑动窗口内有连续 BSN 值的 ARQ 块的最大个数。ARQ_WINDOW_SIZE 的值应该小于或等于 ARQ_BSN_MODULUS 的值。

(4) ARQ_BLOCK_LIFETIME

ARQ_BLOCK_LIFETIME 是一个 ARQ 块的最大生存时间。如果此 ARQ 块被发送（或者重传）后，在该时间超时之前没有收到此发送块的 ACK，则此块被丢弃。

(5) ARQ_RETRY_TIMEOUT

ARQ_RETRY_TIMEOUT 是发送端等待重发的最小时间。在 ARQ 块发送之后，该时间开始计时，一旦到时但还没有收到该块的 ACK，发送端则重发该块。对于同时支持 HARQ 和 ARQ 的连接来说，此值的大小应该能保证在 ARQ 重传开始前，HARQ 对 ARQ 块的重传操作能够完成。

(6) ARQ_BLOCK_SIZE

ARQ_BLOCK_SIZE 是一个 ARQ 块的长度。

(7) ARQ_SYNC_LOSS_TIMEOUT

ARQ_SYNC_LOSS_TIMEOUT 是在发送数据的过程中，在声明发送和接收状态机不同步之前，ARQ_TX_WINDOW_START 或者 ARQ_RX_WINDOW_START 仍旧保持原来值的最大的时间间隔， ARQ 接收和发送状态机各有一个定时器。对于是否能够发送数据它们有各自的准则。

(8) ARQ_DELIVER_IN_ORDER

此参数是用于表明在接收端是否按照发送端发送数据的顺序传给应用层。

(9) ARQ_RX_PURGE_TIMEOUT

此参数是接收端在成功的收到一个不改变 ARQ_RX_WINDOW_START 的值的 block 时，在更新 ARQ_RX_WINDOW_START 的值之前，需要等待的时间。

6.2.2 无线城域网中 ARQ 的设计

在 WiMAX 系统中，为了保障其业务数据传输的可靠性，将其屏蔽于无线链路的不稳定性之外，为数据传输应用了 ARQ 机制，主要技术特点列出如下[2]：

a) 所有属于发送队列的报文段在发送之后不立即释放，而是添加到另一个已发送队列中等待确认；

b) 发送窗口大小根据空中链路传输时延设定；

c) 链路缓冲区大小根据发送窗口设定；

d) 对于收到来自对端确认的报文，从已发送队列删除；

e) 对于收到重传请求的报文，从已发送队列删除并加入等待重传队列，在下一次发送时进行重传；

f) 等待传输队列和等待重传队列都属于 TCP 队列，在 TCP 内部，重传队列级别较高，优先传输，以避免传输延迟过长。

完整的 ARQ 实现过程如图 6-6 所示。

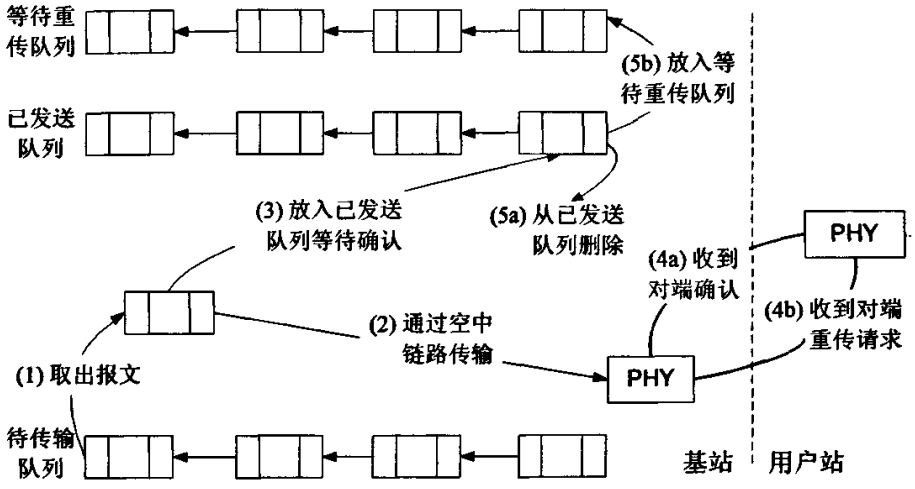

图 6-6 ARQ 处理流程图

以上描述的是数据平面下行方向发送的过程，上行方向接收的处理相对简单，控制平面收到来自物理层的数据帧时通过消息队列通知数据平面。数据平面首先处理链路相关的 MAC 层控制消息，如 ARQ 报文确认和重传请求，然后处理数据报文段，对报文序列号进行检查，发现有不连续的情况时向控制平面报告重传请求，若序列号都连续则根据报文头判断 IP 包的开始和结束，重组成 IP 包交给特定业务会聚子层。

6.3 无线城域网中 ARQ 的实现

在本系统中，ARQ 功能的实现是以一个模块进行的，称为 ARQ 模块。

6.3.1 ARQ 模块的功能

模块功能

本模块的主要功能是将来自分类器模块（CS 子层）的 SDU 根据连接建立过程中协商的 ARQ_BLOCK_SIZE 的值分割成 ARQ block，然后按照一定的传输规则（传输规则包括怎样发送和怎样重发等），将分割好的 ARQ 块发送出去。发送 ARQ 块后，如果在规定的时间内没有收到该 ARQ 块的 ACK，则将该 ARQ 块进行重发，直到收到该 ARQ 块的 ACK 或者是该 ARQ 块的生存期超时，然后将该 ARQ 块从发送队列中删除。

在本系统中，ARQ 的机制有发送状态机和接收状态机。发送状态机主要的功能是对 ARQ 块进行发送，并对发送过的 ARQ 块进行管理，将已经发送的 ARQ 块放入到已发送队列中；对于需要重发的 ARQ 块，放入到重发队列中去并从已发送队列中删除；对于超时或者是收到 ACK 的 ARQ 块，将其从已发送队列中删除；对接收到的 ACK 进行分析，判断其 BSN 值是否有效。

接收状态机的主要功能是对接收到的 ARQ 块的 BSN 值进行分析，判断是否是需要接收的 ARQ；如果是，则产生 ACK 发送给发送端，并将该 ARQ 块储存准备交给主机；否则，丢弃该块，返回发送端 NAK。

当发送和接收状态机不同步的时候，要进行 ARQ reset 的操作。该操作可由发送端或者是接收端发起。对端进行响应。

ARQ 模块在系统中的位置

ARQ 模块在系统中的位置如图 6-7 所示。

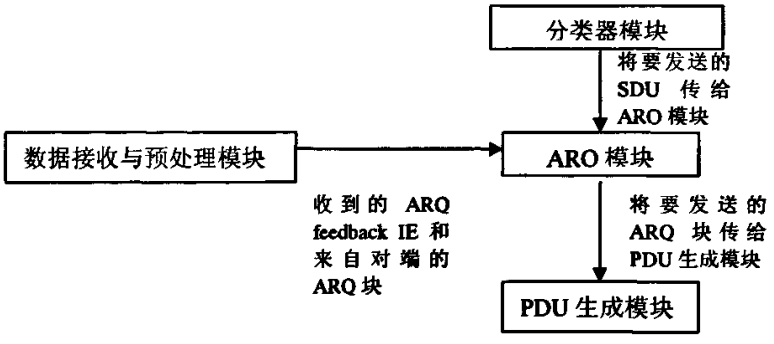

图 6-7 ARQ 模块与系统中其他模块的接口

ARQ 模块主要的接口是分类器模块，PDU 生成模块和数据接收与预处理模块。主要是接收分类器模块的 SDU，将其分割成 ARQ 块，发送到 PDU 生成模块中；从数据接收与预处理模块中接收来自对端的 ARQ 块，经过重新合并后交给 CS 的分类器部分；接收 ACK（NAK），对已发送的 ARQ 块进行删除或者重发操作。分类器模块是 CS 子层的分类器模块，PDU 生成模块和数据接收及预处理模块是 CPS 子层的模块。

6.3.2 ARQ 模块的实现

模块的内部结构

模块的内部结构如图 6-8 所示。

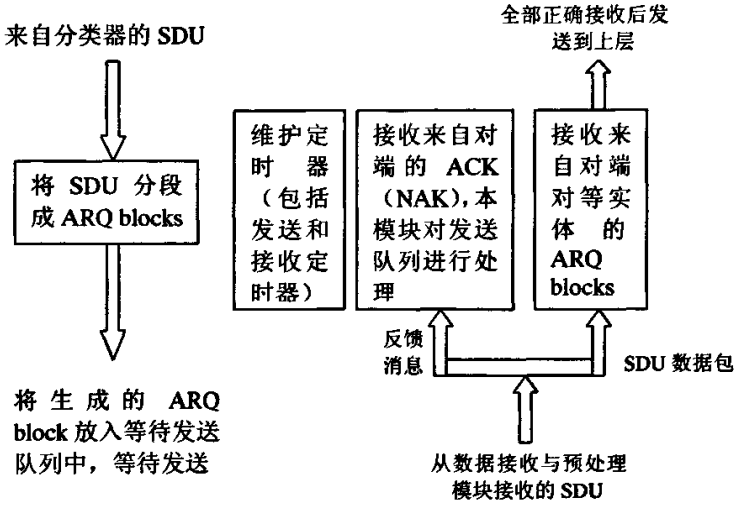

图 6-8 ARQ 模块的内部结构

在 ARQ 模块中，分为 3 个部分：

1，BS 作为发送端：

接收来自分类器的 SDU，根据 TLV 的值分割成 ARQ blocks，将生成的一系列 ARQ blocks 录入到连接管理模块维护的核心数据结构中。需要发送，则根据 TLV 协商的 ARQ_WINDOW_SIZE 的值，通知 PDU 生成模块进行发送。

2，维护定时器部分：

本模块包括的定时器有：

作为发送端的 BS 和接收端的 BS，共需维护如下几种定时器：

（a） 发送端的重发定时器（ARQ_RETRY_TIMEOUT）：此定时器用于控制 ARQ 块的重发操作。在定时器超时之前没有收到 ACK 或者是收到了 NAK，则重发；如果在超时之前收到了 reset 消息，则重置定时器的初值为 0。

（b） ARQ blocks 的生存期定时器（ARQ_BLOCK_LIFETIME）：如果在连接建立的时候没有设定此定时器的值，则此定时器不在起作用，默认为 ARQ 的生存期是无限；

（c） 发送端和接收端的同步定时器（ARQ_SYNC_LOSS_TIMEOUT）：发送端和接收端保持同步的定时器。在 ARQ_RX_WINDOW_START 和

ARQ_TX_WINDOW_START 的值改变的时候，重置此定时器的初值为 0。如果此定时器超时，则发起 reset 消息，重新建立同步。

（d） 接收端收到 ARQ 后等待定时器（ARQ_RX_PURGE_TIMEOUT）：此定时器的作用是当接收端收到的 ARQ 的值不影响 ARQ_RX_WINDOW_START 的值改变时，需要等待的时间。

3，BS 作为接收端：

a) 接收来自对端对等实体的 ARQ blocks，判断此 block 是否能正确接收，如果是已经接收的块，则将此块丢掉，然后返回给对端对等实体 ACK；如果是还未接收到的块，则将此块保存，返回对端对等实体 ACK。

b) 如果接收到的是 ACK 消息，则解析该消息，判断该 ACK 是否为有效的 ACK，如果该 ACK 有效，则将对应的 ARQ 块从发送队列中删除；否则，丢弃该 ACK，不做处理。

ARQ 模块的实现

ARQ 的作用原则是对出错的数据帧自动重发，它有三种形式：停等协议 ARQ、连续 ARQ 和选择重传 ARQ。在本系统中，采用的是连续 ARQ。

在 ARQ 模块中，需要使用 3 个队列：一个是等待重传队列，一个是已发送队列，一个是等待传输队列，由于 ARQ 模块生成的 ARQ 块还需要 PDU 生成模块生成 PDU，在本系统中，连接管理模块维护了一个全局队列 DU，PDU 生成模块直接从该模块中取出数据进行生成 PDU 的操作。因此，ARQ 模块中使用的 3 个队列由 ARQ 模块自己来维护。

a) 发送数据部分

i. BS 端作为发送端，对数据的处理过程：

ii. 先将接收到的 SDU 分成 ARQ 块，放入 ARQ 模块自己维护的等待传输队列中；

iii. 如果需要发送 ARQ 块，则根据发送窗口的大小将要发送的 ARQ 块放入到 DU 队列中（由连接管理维护，PDU 生成模块将从该队列中取出数据，生成 PDU 进行发送）；并为该 ARQ 块启动超时定时器；

iv. 并将该部分 ARQ 块放入到已发送队列中，从等待传输队列中删除；

v. 如果收到了重传请求或者是定时器超时，则将该部分 ARQ 块从已发送队列中取出，放入等待重传队列；在下一次发送时，将该部分 ARQ 块放入到 DU 队列中；

vi. 如果收到了已发送的 ARQ 块的 ACK，则将已发送的 ARQ 块从已发送

队列中删除；将该 ARQ 块的超时定时器停止；并将该 ARQ 块后续的 ARQ 块从等待传输队列中取出，放入到 DU 队列中。

b) 接收数据部分

BS 端作为接收端，如果 ARQ 模块收到了正确的 ARQ 数据，则产生该 ARQ 块的 ACK。

为了节省资源，在本系统中，采用在连续收到好几个正确的数据帧以后，才对最后一个数据帧发确认信息的方式。这就是说，对某一数据帧的确认就表明该数据帧和这以前所有的数据帧均已经正确无误的收到了。ACK 的参数包括连接标志符 CID，最后一个正确的 ARQ 块的 BSN 的值，ACK_type（本系统中 ack_type 的值为 0x1，CumulativeACK entry）。

作为发送端发送数据的过程如图 6-9 所示：

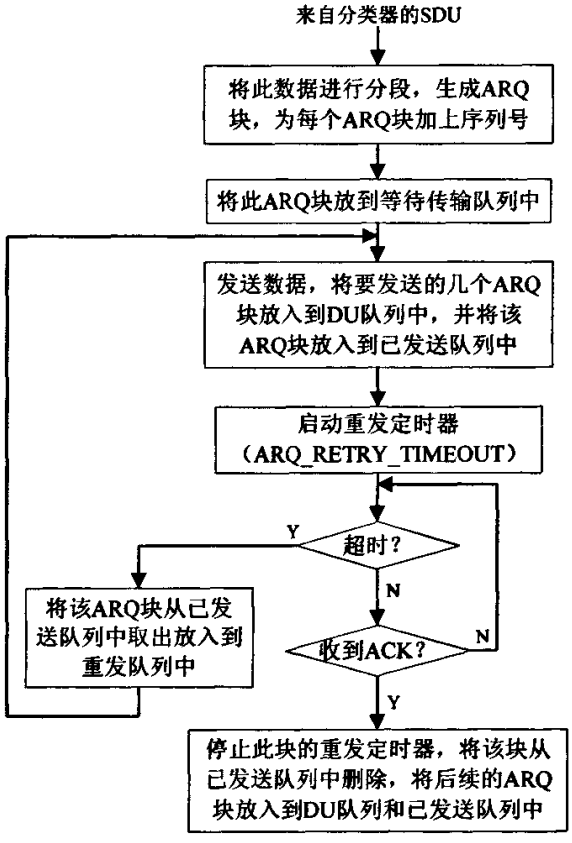

图 6-9 发送端对 ARQ 的操作过程

接收来自对端的数据的处理过程如图 6-10 所示。

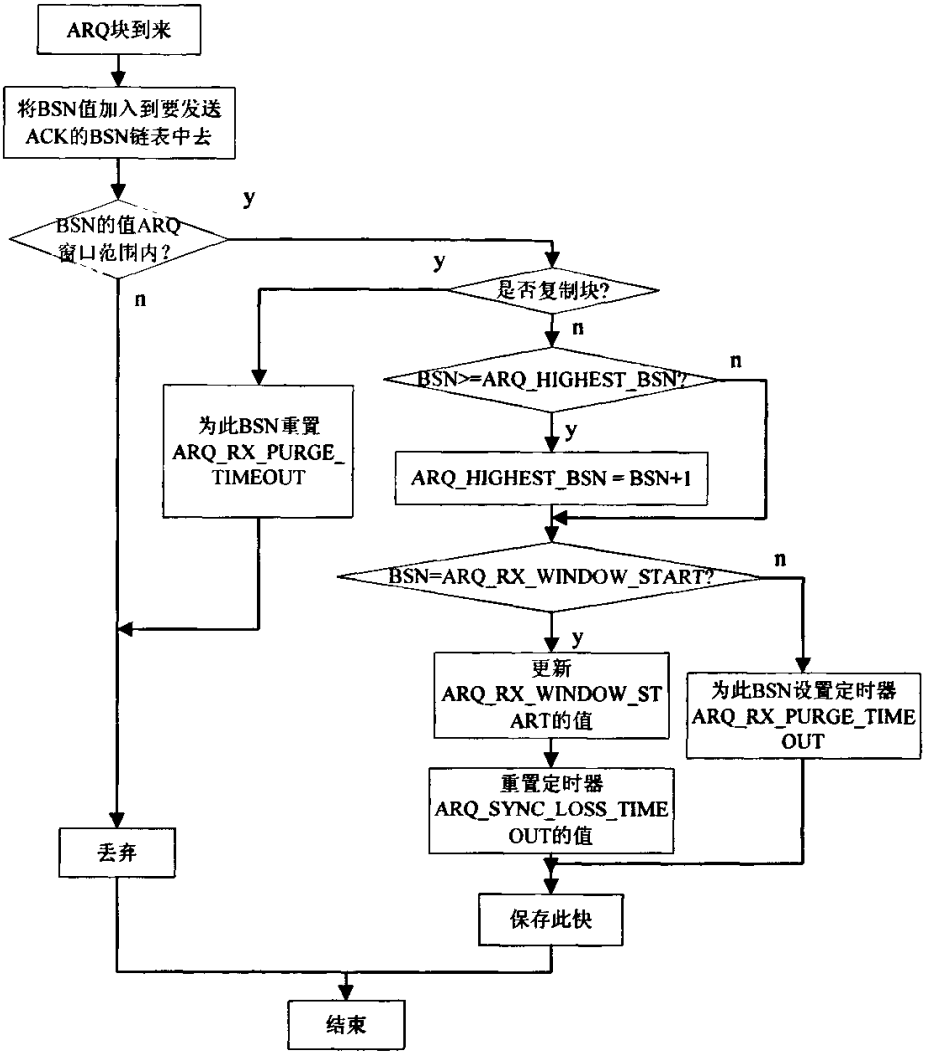

图 6-10 接收到对端数据的处理过程

6.3.3 ARQ 模块的核心数据结构

因为最后要将生成的 ARQ 块放入到核心数据结构中，因此，Arqblk_Node 的结构如下：

```
typedef struct{
    Node ln; //Node 是 VxWorks 上提供的变量类型
    void *du_p;
    UINT16 du_len;
```

```
    DuType du_type;
    UINT16 bsn:11;
    UINT16 last:1;/*该位表示该 Arqblk_Node 是否为该 SDU 上的最后一个，0 表
                    示否，1 表示是最后一个*/
    UINT16 resv:4;
}Arqblk_Node;
```

6.4 本章小结

IEEE 802.16 MAC 协议是面向连接的，当用户站(SS)被激活进入网络，会与基站(BS)建立一个或多个用于数据传输的连接。MAC 层对无线资源的使用进行调度并根据业务不同提供 QoS 保证，通过采用链路自适应技术和自适应重传(ARQ)技术以提供比较高的频谱效率。本章介绍了一种 802.16 系统中 ARQ 的实现方案。该方案被应用在无线城域网话音通信系统的基站设备软件设计和实现中。

第七章 测试与调试

宽带无线城域网多媒体实时通信系统项目在交付使用以前必须经过严格的软件测试，通过测试尽可能找出概要设计、详细设计以及软件代码编写中的错误，并加以纠正，从而获得高质量的产品。软件测试要求在对每一个模块的编码之后先做程序测试，再做单元测试，然后再进行集成（综合或组装）测试，系统测试，验收（确认）测试。本章简要介绍了 CS 子层的集成测试过程，在本章的最后，对测试进行了总结。

7.1 CS 子层的单元测试

7.1.1 测试目的

CS 子层各个模块的独立测试通过后，开始进行 CS 子层的集成测试。CS 子层各模块的联合调试，主要是对功能进行测试，测试项目包括：

1) 服务流和规则管理部分

测试能否正确的建立/修改/删除服务流，能否根据建立的不同 Qos 的服务流，保存相应的规则；

2) 分类器收发数据部分：

测试能否正确的对接收到的数据进行分类，找到匹配的规则。接收到的数据主要分为两种：

a) 从 transportcid 收发的数据部分

b) 从 secondarycid 收发的数据部分

3) 将定时器处理加入的综合测试部分

由于 CS 程序中缓存表，规则表和 cid 表都有定时器进行维护，在调试前面的 2 个部分时先将定时器禁掉，等这三部分调通后再根据 6.3.3 中的场景与定时器连调。

7.1.2 测试内容

1） 服务流和规则管理部分

a) 应用层进行 Qos 建立/修改/删除请求：CS 子层能否根据请求的 Qos 参数，发起建立/修改/删除服务流的请求给 MAC CPS 子层；

b) 在发送建立请求，得到对端的应答后，能否正确的保存规则；

c) 对于发送修改请求的情况，能否正确的删除原来的规则，并保存新的规则；

d) 对于发送删除请求的情况，能否正确的删除规则；

e) 处理来自对端的 Qos 建立/修改/删除请求：CS 子层能否正确的处理来自对端的服务流建立/修改/删除请求，能否根据请求产生正确的回应消息给对端；

2）分类器收发数据部分

a) 从 transportCID 收发的数据部分：

i. 接收到上层的数据查到对应 cid

ii. 接收到上层的数据未查到 cid，缓存规则和 primarycid

iii. 接收到上层的数据未查到 cid，缓存规则，但查到 primarycid

b) 从 SecondaryCID 收发的数据部分

i. 接收到上层的需要从 secondarycid 发送的数据并查寻 cid 成功

ii. 接收到上层的需要从 secondarycid 发送的数据但未找到对应的 cid

3）将定时器加入的综合调试部分

在以上测试过程中已经建立好的规则表，CID 表的基础上进行测试。主要测试一下内容：

a) 规则表和 CID 表的定时处理

b) 缓存队列的定时处理

7.1.3 测试方法

1） 服务流和规则管理部分

应用层模拟程序产生 DL_QOS_CREATE(/CHANGE/DELETE).request 消息→服务流管理模块分析消息，生成 MAC_CREATE(/CHANGE/DELETE)_SERVICE FLOW.request 消息发送到 MAC 接口模块 →MAC 模拟程序取出该消息，返回 MAC_CREATE(/CHANGE/DELTE)_SERVICE FLOW.confirmation 消息→服务流管理处理该消息，生成 DL_QOS_CREATE(/CHANGE/DELTE).response 消息给上层。在此过程中正确建立规则。如果是修改和删除的情况，需要对原来的规则进行删除的操作。

MAC 层模拟程序产生 MAC_CREATE（/CHANGE/DELETE）SERVICE FLOW.indication 消息→服务流管理模块分析该消息，生成 MAC_CREATE(/CHANGE/DELETE)_SERVICE FLOW.response 消息→ MAC 模拟程序获得该消息。

以上均可以通过打印消息内容看传递和解析是否正确。

2） 分类器收发数据部分

a) 从 transportcid 上收发的数据部分

i. 发送数据查到 cid

应用层模拟程序调用 sendpkt 函数发送数据→分类器模块进行规则匹配，查找 cid→找到则通过 MAC 接口函数传给 MAC 层→MAC 层模拟程序取的数据后进行解析打印，并将数据回传给 CS 层（MAC_DATA.indication）→CS 接收到数据后解析打印

ii. 发送数据未查到 cid，缓存规则和 primarycid

应用层模拟程序调用 sendpkt 函数发送数据→分类器模块进行规则匹配，查找 cid→未查到，查找缓存表未找到→primarycid 未找到，则发送 CID 请求→MAC 模拟程序收到地址请求后直接回复地址应答→地址转换模块处理地址应答，建立 CID 映射→在 primarycid 上发送服务流请求，建立临时规则→发送数据→MAC 模拟程序将数据回传→CS 接收后打印→发送相同的数据，检查查寻临时规则表的功能。

iii. 发送数据未查到 cid，缓存规则，但查到 primarycid

应用层模拟程序调用 sendpkt 函数发送数据→分类器模块进行规则匹配，查找 cid→未查到，查找缓存表未找到→找到 primarycid，在 primarycid 上发送服务流请求，建立缓存规则表→应用层模拟程序继续发送相同的数据，检查是否正确的缓存了这些数据→收到服务流应答成功建立临时规则→MAC 模拟程序将数据回传→CS 接收后打印

b) 从 secondarycid 上收发的数据部分

i. 发送需要从 secondarycid 发送的数据并查寻 cid 成功

应用层模拟程序调用 sendpkt 函数发送 DHCP 数据→分类器模块进行规则匹配，发现是需要从 secondarycid 发送的数据→从 cid 表中查到该 secondarycid，在该 cid 上数据发送→MAC 层模拟程序取的数据后进行解析打印，并将数据回传给 CS 层（MAC_DATA.indication）→CS 接收到数据后解析打印。在此测试以前，需要首先进行以下测试，建立 secondarycid 和 MAC 地址的对应表：

MAC 模拟程序发送 MAC_MANAGEMENT_CID.indication→地址转换模块建立 cid 表项

ii. 发送需要从 secondarycid 发送的数据但未找到对应的 cid

应用层模拟程序调用 sendpkt 函数发送 TFTP，DHCP 数据→分类器模块进行规则匹配，发现是需要从 secondarycid 发送的数据→从 cid 表中查找该 secondarycid，未找到→发送地址请求消息→ MAC模拟程序收到地址请求后直接回复地址应答→地址转换模块处理地址应答，建立 CID 映射→在 secondarycid 上发送数据→MAC 层模拟程序

取的数据后进行解析打印，并将数据回传给 CS 层（MAC_DATA.indication）→CS 接收到数据后解析打印→发送相同的 TFTP，DHCP 数据测试，验证是否可以正确查到 secondarycid。

3） 将定时器加入的综合调试部分

在以上测试过程中已经建立好的规则表，CID 表的基础上进行测试。

a） 规则表和 CID 表的定时处理

每隔一定时间发送一个可以从规则表中查到的数据报（根据规则表的定时器时间），开始时正常查到 cid 发送，当规则表超时以后就需要重新进行服务流请求。其中包括 primarycid 表未超时和已超时两种情况，若 primarycid 超时已被删除，则首先需要进行 cid 请求。

b） 缓存队列的定时处理

发送服务流请求时 MAC 模拟程序不进行应答，则缓存队列进行重发，重发到一定次数后将该缓存删除，并打印错误信息。

7.1.4 测试结果

1） 服务流和规则管理部分

建立服务流的测试输出结果如图 7-1 所示。

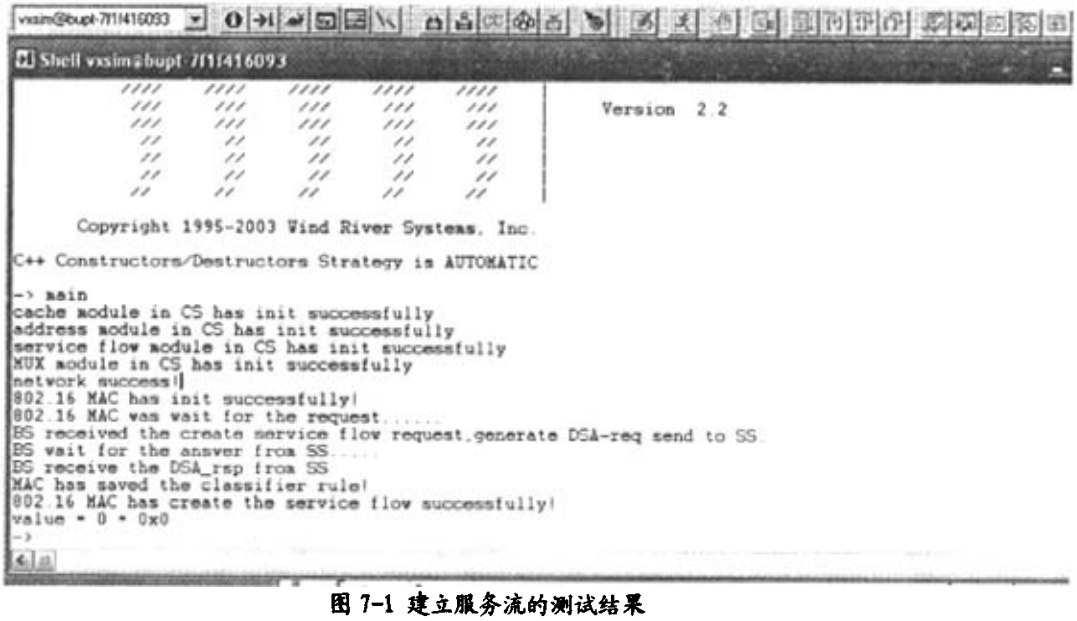

图 7-1 建立服务流的测试结果

测试结果表明，该服务流部分能够正确的根据接收到的 Qos 参数，建立/修改/删除服务流，并能根据服务流的情况，相应的保存，删除规则。

2) 分类器收发数据部分

找到 CID 的测试结果如图 7-2 所示。

图 7-2 分类器对数据包的处理结果

测试结果表明，该分类器能够正确的对接收到的数据进行分类，对于从 transportcid 上收发的数据，能够正确的查找到对应的 cid，发送数据；如果没有找到 cid，则查找到 primarycid，发起建立服务流的请求，并能够将数据包缓存。对于从 secondarycid 上发送的数据，能够正确的查找到 secondarycid，发送数据；对于没有找到 secondarycid 的情况，根据 MAC 地址获得 secondarycid，再发送数据。

3) 将定时器加入的综合调试部分

能够进行规则表和 CID 表的定时维护，对于超时的项，能够删除；对于缓存队列，能够实时的对各缓存的内容进行处理，超时的能够进行正确的处理（删除该项，并通知存入的模块）。

7.1.5 测试总结

对 CS 子层的集成测试是通过编写测试函数在模拟环境下进行的，下面总结一下在程序设计和编写过程中应该注意的问题：

1) 合理将整个 CS 子层划分成各个模块有助于工程的实现；模块的划分要注意同其它模块的接口设计；
2) 详细设计要包含功能实现的每一个细节，在程序实现中，涉及到根据数据报头查找匹配的规则，查找 CID 的过程，要能够正确匹配到规则，必须对数据报头的各个单元的值有正确的认识；

3) 在程序编写过程中，要注意内存的使用，在使用内存前，必须判断是否由申请空间，并且对动态申请的内存空间要能够保证释放；
4) 宏的定义要统一，在函数实现中尽量用宏代替常量；
5) 在各个模块开发中要保持全局数据结构的统一；
6) 因为涉及到各个模块间的消息传递，所以各个模块间的消息接口的定义要有一个统一的规范，而且修改后要保证版本的一致；
7) 需要添加相应的打印信息，以保证正确的定位问题。

7.2 宽带无线城域网 MAC 层的总体测试

在各模块进行完单元测试后，开始进行整个 MAC 层的联合测试，在本次测试的过程中，所用到的设备是：

PC 机：　　2 台；

开发板 DVK102：　　2 套（每套是一个子板和一个母板）；

串口线、网线：若干；

交换机：　　2 台

测试环境如图 7-3 所示。其中，pc 机和 ARM 板的 IP 地址，MAC 地址等都是配置过的。

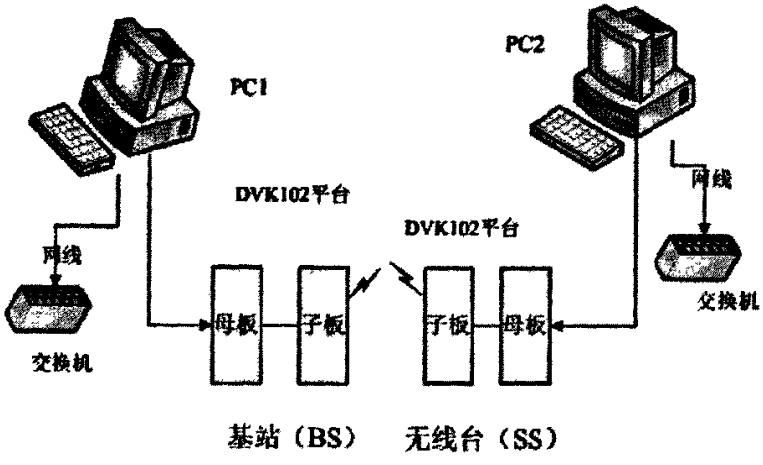

图 7－3 MAC 层的总体测试构建图

系统由 2 个 DVK102 开发平台组成，分别代表基站和无线台；2 台 PC 机分别连接到 2 块 DVK102 开发平台上，用于实现 WiMAX 系统中 MAC 层以上的数据处理。

在进行整体的测试过程中，以播放视频业务，ftp 下载业务和访问网页同时进行。

测试结果如表 7-1 所示。

表 7-1 MAC 层的测试结果

功能	测试方法	测试结果
网页浏览	将用户主机通过集线器连接到用户站，取得有效 IP 地址后，通过 IE 浏览器访问网站	用户主机可以进行网页浏览
文件下载	将用户主机通过集线器连接到用户站，取得有效 IP 地址后，通过 FTP 方式下载文件	用户主机可以通过 FTP 方式下载文件
视频播放	将用户主机通过集线器连接到用户站，在取得有效 IP 地址后，连接到服务器，点播一到多个视频文件	用户主机可以点播服务器上的媒体文件

结束语

很高兴在读硕士期间能够参与如此前沿和实用的项目，在项目组所有成员的共同努力下按时完成了无线城域网络多媒体实时通信系统的软件开发和联调工作，整个系统实现了设计方案中所要求的各项功能，也达到了预期的各项性能指标。

通过参与这个项目，使作者在理论知识和实践能力上都有很大收获。通过项目开发，一方面加深了作者对理论知识的掌握和理解，有机会全面深入地接触在通信领域处于领先地位的 WiMAX 技术，对无线通信系统有了更深刻的理解；另一方面，提高了作者的动手实践能力，培养了作者的学习能力以及思考、分析并解决问题的综合能力；更重要的是作者参与了整个系统的前期设计，模块开发和后期联调，在通信产品设计，嵌入式系统开发，协议软件开发和项目管理等方面积累了宝贵的经验和教训。下面谈谈作者的一些心得体会：

a) 嵌入式开发和桌面机开发有很大区别，很多在桌面机系统中很少出现的问题在嵌入式系统中却相当普遍；例如内存资源不足、调试手段有限等，而嵌入式系统自身对实时性和高效性的要求，也需要开发者对这些问题有更多关注，因此在设计编码前应认真阅读相关书籍，积累相关经验；
b) 软件产品开发主要分为：需求分析、概要设计、详细设计、编码、测试、安装及维护六个阶段。开发过程要合理按照这几个阶段步步推进；
c) 重视项目设计及文档工作。好的设计可以使后期工作事半功倍，差的设计则会导致事倍功半。同时，设计的文档化也至关重要，设计文档化的过程其实是设计程序流程，整理设计思路的过程；
d) 代码开发期间，要求每个人注意良好的编码习惯。格式整齐、命名规范、注释齐全的代码不仅有利于他人阅读，对提高自己的工作效率也有很大帮助；
e) 要善于利用编译选项以保留工程的不同阶段版本；
f) 系统联调是整个项目的最重要阶段，涉及所有系统模块，加强大家的合作精神和相互理解，是联调成功的关键；
g) 项目管理是项目开发中必不可少的部分，尤其是文档管理和项目进度管理；
h) 项目过程中要多与同组同学沟通；还要善于学习，不断提高，同时要锻炼积极思考、分析和解决问题的能力，并及时进行经验和方法的归纳总结。

参考文献

[1] 网络博客 WiMAX 宽带无线接入的特点及其应用模式

[2] 曹迎新 硕士论文 9 页

[3] 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, 2004.

[4] 曾春亮 WiMAX/802.16 原理与应用 机械工业出版社 2006 P46-49

[5] 宋海波，谈振辉 802.16 宽带无线接入系统的 QoS 保证和调度策略 电信科学 2005 年第 3 期 46-49 页

[6] 雷震洲 IEEE 802.16 和 WiMAX 现代电信科技 2004 年 7 月 2-7 页

[7] 王国辉 硕士论文 20－25 页

[8] 陈家琪 计算机网络 上海理工大学计算机工程学院讲课稿 第四章

[9] 毛妙，张卫，陆刚 嵌入式 IP 协议栈网络接口设计与分析 计算机应用 第 25 卷 2005 年 12 月

[10] 孔祥营，柏桂枝 嵌入式实时操作系统 VxWorks 及其开发环境 Tornado 中国电力出版社 2002

[11] 刘尉悦，张万生，邢涛等 VxWorks 操作系统及实时多任务程序设计 单片机与嵌入式系统应用 2001 年第 5 期

[12] 刘艳秋，安军社，张健等 基于 VxWorks 的以太网接口设计与实现 计算机工程 第 30 卷第 13 期 2004 年 7 月

[13] 谭浩强 C 程序设计 第二版 清华大学出版社 1999

[14] WiMAX FORUM White Paper, "IEEE 802.16a Standard and WiMAX Igniting Broadband Wireless Access," Sep, 2003

[15] WiMAX FORUM, "Business Case Models for Fixed Broadband Wireless Access based on WiMAX Technology and the 802.16 Standard," Oct, 2004

[16] Carl Eklund, Reger B. Marks, Kenneth L. Stanwood, et al., "IEEE Standard 802.16: Technical Overview of the Wireless MAN Air Interface for Broadband wireless Access, " IEEE Communications Magazine, pp. 98-107, Jun. 2002.

[17] Govindan Nair, Joey Chou, Tomasz Madejski, et al., "IEEE 802.16 Media Access

Control and Service Provisioning," Intel Techonogy Journal (Volume 8, Issue 3), pp. 213-228, Aug. 2004.

[18] Arunabha Ghosh, David R. Wolter, Jeffrey G. Andrews, et al., "Broadband Wireless Access with WiMax/802.16: Current Performance Benchmarks and Future Potential," IEEE Communications Magazine, pp. 129-136, Feb. 2005.

附 录

在本系统中，所用到的 CS 子层和 CPS 子层之间交互的原语：

a) MAC_CREATE_SERVICE FLOW.request
b) MAC_CREATE_SERVICE FLOW.indication
c) MAC_CREATE_SERVICE FLOW.response
d) MAC_CREATE_SERVICE FLOW.confirmation
e) MAC_CHANGE_SERVICE FLOW.request
f) MAC_CHANGE_SERVICE FLOW.indication
g) MAC_CHANGE_SERVICE FLOW.response
h) MAC_CHANGE_SERVICE FLOW.confirmation
i) MAC_TERMINATE_SERVICE FLOW.request
j) MAC_TERMINATE_SERVICE FLOW.indication
k) MAC_TERMINATE_SERVICE FLOW.response
l) MAC_TERMINATE_SERVICE FLOW.confirmation
m) MAC_DATA.request
n) MAC_DATA.indication

致 谢

在本论文完成之际，首先，作者要特别感谢导师崔晓燕副教授在作者研究生学习和工作中给予本人的悉心指导，以及对作者生活上的关怀和照顾。

崔老师治学严谨，她在学习和研究上的启发将使本人终生受益，并在本人的论文撰写过程中给予耐心的指点，并时时激励本人在学习和工作中不断进取。

在此，还要感谢本项目组的其他同学。他们对作者的研究工作提出了很好的建议，与他们的讨论使作者受益匪浅，他们孜孜不倦认真学习的态度和勤勤恳恳的工作作风共同维护了实验室中良好的研究环境与学术风气。

感谢作者的父母对作者多年的养育之恩、无私的爱和有力的支持，他们的付出作者终生都难以回报。

在即将完成论文、告别校园之际，我要深深地感谢母校对我多年的教育和培养。