

IEEE 802.11 中站点 MAC 协议的分析与初步实现^{*}

卢佳 杨士中

(重庆大学通信工程学院 重庆 400044)

摘要 简要分析了 IEEE802.11 中站点(STA)的 MAC 协议,给出了一种基于 ARM9 和 uCOS-II 的 IEEE 802.11 中站点 MAC 协议实现方法,重点阐述了站点 MAC 协议中数据流的 C 语言实现,包括将 LLC 层传来的数据组装成帧、分段、加入队列缓冲区、帧效验序列(FCS)的计算,以及接收端的相反操作。

关键词 IEEE802.11, 无线局域网, MAC, ARM, uCOS-II

Analysis and Primary Implementation for IEEE 802.11 MAC in STA

LU Jia YANG Shi-Zhong

(College of Communication Engineering, Chongqing University, Chongqing 400044)

Abstract This paper analyzes the IEEE802.11 STA's MAC standard briefly and introduces a method of primary implementation for IEEE 802.11 MAC in STA (Station) based on the platform of ARM9 and uCOS-II, details the implementation of data flow including making a frame, fragmentation, queue and FCS calculation in transmission and the reverse operations in reception.

Keywords IEEE 802.11, WLAN, MAC, ARM, uCOS-II

IEEE 802.11^[1]无线局域网因其灵活性、移动性、高吞吐量和低成本得到了广泛的应用,但目前无线局域网的核心技术仍然为国外所掌握,所以,对此项技术进行深入的研究和开发是非常必要的。IEEE 802.11 MAC 层制定了多用户共享媒体的规则,是无线局域网的关键技术之一,对无线局域网的网络的性能(如数据吞吐量、网络延迟等)有着重要的影响。

1 IEEE 802.11 中站点 MAC 协议概述

802.11 的 MAC 层主要有三个功能:无线介质访问、网络连接、数据的验证和加密。在介质访问方面采用的是 CSMA/CA 机制来避免碰撞。网络连接方面,当站点(STA)进入一个或多个基本服务集(BSS)覆盖范围时,它会选择加入某一 BSS 进行通信,这一过程包括扫描、同步、鉴权、关联。在数据的验证和加密方面采用的是 CRC32 算法和 WEP 机制,保证了数据传输的准确性和网络安全。此外,MAC 层制定了 RTS/CTS 机制,以解决隐藏节点的问题;提供了分段重组功能以减少数据重传概率,提高系统性能;采用活动、节能这两种功率管理模式来增强移动 STA 电池的使用时间。

2 站点 MAC 层数据收发流程

STA 的 MAC 层从 LLC 层接收数据,组装成 MAC 帧,经过分段后加入发送队列缓冲区,确定系统中无数据传输后提取队首的数据准备发送,在 STA 争抢到信道后计算 FCS 并发送。接收操作与发送相反,接收到一个完整分段后将此分段加入接收缓冲区,在数据的所有分段接收完毕后重组成一个 MAC 帧,传至 LLC 层。

2.1 发送流程

数据的发送可由 5 个模块来实现,如图 1 所示。最高层

模块 MSDU_from_LLC 从 LLC 层接收数据,一层一层地向下传送,每个模块都会对上个模块传来的数据进行处理。数据成功发送出后,将产生传输确认信号,确认信号也会经过各模块的处理后向上层层传送,直至 MAC_SAP。

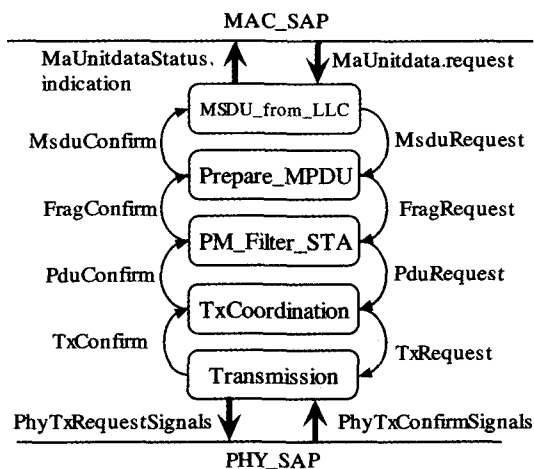


图 1 发送数据流程图

MSDU_from_LLC:该模块从 LLC 层接收消息,消息中不仅包含了需要物理层传输的有用数据 LLCdata,还包括了该数据的源地址、目的地址、路由信息、优先级及服务等级。该模块将对这些附加信息进行判断,将符合要求的消息保留下来,组装成 MAC 帧,传至下一模块任务的消息队列中。

Prepare_MPDU:该模块收到上层消息后,将重新确定消息中 MAC 帧的目的地址和接收地址,并根据该帧的长度决定是否进行分段。

^{*}基金项目:国家发改委 CNGI2005 示范工程项目(CNGI-04-4-2D);重庆市教委科学技术研究项目(040507);重庆市科委自然科学基金项目(CSTC2006BB2164)。卢佳 硕士研究生,主要从事无线局域网的研究;杨士中 中国工程院院士,教授,博导,长期从事通信技术的研究。

PM Filter STA:该模块需设计一个队列缓冲区,若模块任务的消息队列中有消息,则将消息中的数据加入缓冲队列,等待传输。在收到上一数据传输完毕的确认信号后,取出队首的数据,发送至 TxCoordination 模块,准备传输。

TxCoordination:该模块是传输协调模块,主要完成以下功能:判断是否需要启用 RTS/CTS 机制,启动退避,接收应答帧(CTS、ACK),对发送成功的、失败的、重传的帧进行计数,并据此判断是需要重传还是直接丢弃某帧。若某帧被分段,则每次只向下一模块发送该帧的一个分段,在收到已发送的确认信息后,开始等待 ACK,收到 ACK 后,需等待一个 SIFS(短帧间间隔)再传送该帧的下一分段。

Transmission:该模块分为两个子模块:Backoff_Procedure 和 Data_Pump。Backoff_Procedure 子模块主要实现退避功能,根据 $CW(aCW_{max} \leq CW \leq CW_{min})$ 的值,计算需退避的时隙数 slotCnt,若在信道空闲时收到 slot 信号,则将 slotCnt 减一,当 slotCnt 为 0 时,退避完成。Data_Pump 子模块将计算需发送帧的 FCS,并将该帧连同 FCS 一个字节一个字节地发送,每发送一个字节都需向物理层发送请求,在收到确认信号之后才传输下一字节。整个帧传输完毕后向上层发送确认信号。

2.2 接收流程

接收时需要进行与发送相反的操作,如图 2 所示,主要由以下模块实现:

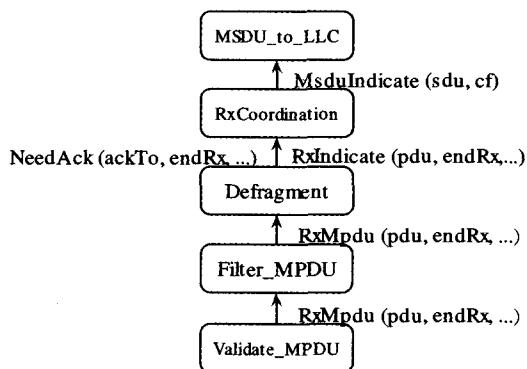


图 2 接收数据流程图

Validate_MPDU:该模块从物理层一个字节一个字节地接收数据。计算 FCS,判断该帧在传输过程中是否出错,出错的帧将被丢弃,正确的帧在去掉 FCS 后发送到 Filter_MPDU 模块。

Filter_MPDU:该模块主要完成以下功能:判断接收到的帧是单播帧还是多播帧;判断单播帧的目的地址是否与 STA 的 MAC 地址相同;判断多播帧是否来自于同一 BSS;判断该帧是否被重复接收;判断该帧是否需要 ACK 应答等。

Defragment:该模块有一个缓冲区,缓冲区中可以存放从不同地址发来的 MAC 帧,并完成被分段的 MAC 帧的重组。重组好后的 MAC 帧被发到 RxCoordination 模块。

RxCoordination:该模块是接收控制模块,在接收到数据帧时,向上层传送至 MSDU_to_LLC;接收到 NeedAck 消息时,构造 ACK 帧并发送至 Data_Pump 模块;接收到 RTS 时,构造 CTS 帧并发送至 Data_Pump 模块,收到管理帧则发送到相应的管理模块。

MSDU_to_LLC:该模块处于接收流程的最高层,将从 RxCoordination 接收到的 MAC 帧中提取出有用数据及其它

信息如目的地址、源地址等,传送到 LLC。

3 站点系统功能的实现

3.1 实现框架

站点的 MAC 层实现以 ARM9 和 uCOS-II^[2] 为平台,整个系统可划分为多个模块来实现其功能。每个模块的功能由一个或多个 uCOS-II 的任务来实现,可将任务看作是协议描述语言(SDL)中的状态,每个任务有一个或多个消息队列,任务之间通过发送和接受消息来进行通信。当某任务获取 CPU 的使用权后,将从该任务的消息队列中提取消息,将该消息进行处理之后发送到下一任务的消息队列中。一个任务向另一任务发送消息,将有可能引起任务的切换(SDL 中状态的转移^[3])。另外,部分模块中也用到了信号量来控制任务的执行。

系统中的每个任务都是无限循环,任务的执行与否由 uCOS-II 统一调度,决定任务是否执行取决于该任务的优先级和它是否处于就绪状态。发送队列缓冲区是一个单向循环链表,通过三个全局指针来完成从队列存数据、从队列取数据和释放已发数据所对应的内存块的操作。接收的缓冲区是结构体数组,由三个函数分别完成查找未使用的缓冲块、查找接收某分段的缓冲块以及清理使用完毕或接收超时的缓冲块的操作。发送和接收数据的缓冲区中都用到了动态内存分配,内存块的获取和释放都由 uCOS-II 的系统函数 OSMemGet 和 OSMemPut 完成。

3.2 主要数据类型的定义

在各模块中传递和处理的数据有以下几种基本的自定义的数据类型:字节、字符串、自定义的结构体和数组。它们的定义如下(仅列出几种最基本的):

```

typedef unsigned char INT8U;
typedef INT8U Octet; /* 字节的定义 */
typedef struct
{
    int length;
    INT8U * ptr;
} Octetstring; /* 字符串的定义[4] */
typedef Octetstring Frame; /* 帧的定义 */
typedef INT8U MacAddr[5]; /* MAC 地址的定义 */
    
```

3.3 分段

MSDU 经过 Prepare_MPDU 模块处理后将被存放在 FragSdu 的结构体中,该结构体的定义如下:

```

typedef Frame FragArray[sMaxFragNum];
typedef struct frs
{
    struct frs * pnext; /* 指向下一个 FragSdu */
    INT8U * pblk; /* 动态获取内存块的首地址 */
    bool fUse; /* 标记该 FragSdu 是否已用 */
    .....
    MacAddr dst; /* 该 MSDU 的接收地址 */
    .....
    FragArray pdus; /* 每个数组元素为 MAC 帧 */
}FragSdu;
    
```

前三个成员是为处理分段和队列而增加的,Prepare_MPDU 模块在处理某帧时,将动态申请一内存块^[5](该内存块将在该帧的所有分段传输完成后释放),用于存放该帧。若某帧需要分段,则被分段后的帧将成为数个完整的 MAC 帧被依次存放于动态申请的内存块中,并与 pdus 域中的每个数组元素对应,如图 3 所示。

3.4 队列缓冲区

发送和接收流程中的缓冲区实现的基本方法是一致的,

(下转第 136 页)

- shops (ICDEW'06)
- 3 Chuang Shui-Lung, Hsu J Yung-jen. Tree-structured Template Generation for Web Pages. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)
 - 4 Li Zhao, Ng Wee Keong. WICCAP: From Semi-structured Data to Structured Data. In: Proceedings of the 11th IEEE International Conference and Workshop on the Engineering of Computer-based Systems (ECBS'04)
 - 5 陈彦,徐宏炳,王能斌. 基于标记图的 Web 数据模型. 计算机学报, 1999, 22(3)
 - 6 李效东,顾毓清. 基于 DOM 的 Web 信息提取. 计算机学报, 2002, 25(5)
 - 7 Mundluru D, Katukuri J R, et al. Automatically Mining Result Records from Search Engine Response Pages. In: Proceedings of the Fifth International Conference on Data Mining (ICDM'05)
 - 8 Joshi A, Todwal S. Evolutionary Machine Learning for Web Mining. IEEE, 2003
 - 9 鲁明羽,孙建涛,陆玉昌. 一种基于联想的网页推荐方. In: Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, P R China, June 2004
 - 10 高军,王腾蛟,杨冬青,等. 基于 Ontology 的 Web 内容二阶段半自动提取方法. 计算机学报, 2004, 27(3)
 - 11 Baumgartner R, Flesca S, Gottlob G. Visual Web Information Extraction with Lixto. In: Proceedings of the 27th VLDB Conference, Roma, Italy, 2001
 - 12 陈兰,等. 一种新的基于 Ontology 的信息抽取方法. 计算机应用研究, 2004
 - 13 Muslea I, Minton S, Knoblock C. A Hierarchical Approach to Wrapper Induction. In: Proc. Third Ann Conf Autonomous Agents, 1999
 - 14 Kushmerick N. Wrapper Induction: Efficiency and Expressiveness. Artificial Intelligence, 2000
 - 15 Meng Xiaofeng, Wang Haiyan, Hu Dongdong, et al. SG-WRAM Schema Guided Wrapper Maintenance: A Demonstration. In: Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
 - 16 Meng Xiaofeng, Hu Dongdong, Li Chen. Schema-guided Wrapper Maintenance for Web-Data Extraction. WIDM'03, November 2003
 - 17 Meng Xiaofeng, Wang Haiyan, Hu Dongdong, et al. A Supervised Visual Wrapper Generator for Web-Data Extraction. In: Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC'03)

(上接第 91 页)

在此只详述发送中的缓冲队列。

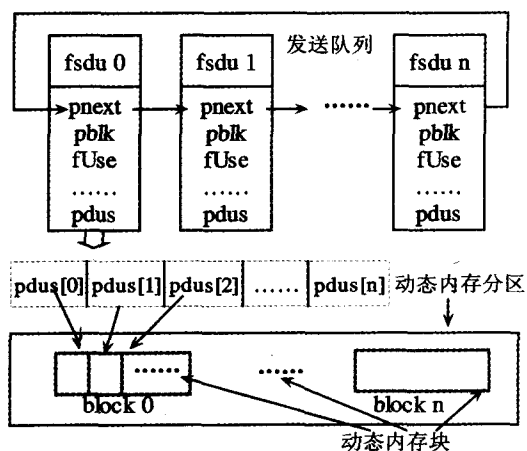


图 3 发送队列缓冲区

发送模块的队列缓冲区用于存放待发送的数据,当系统处于退避状态或各种帧间间隔时,可以继续处理来自 LLC 层的数据,将其放入缓冲队列中,等待发送。当收到上一帧传输完成的确认消息后,再将缓冲区队首的数据取出,准备发送。这样既不会对正在发送的数据产生干扰,又提高了执行效率。本方案采用单项循环串表加 3 个全局指针来实现这一功能。

基本数据类型定义如下:

```
typedef FragSdu SduQueue[Max_Frag_In_Q];
SduQueue txQ;
```

整个队列由一个 SduQueue 类型的数组构成,数组中的每个元素都是 FragSdu 结构体,每个 FragSdu 中的 pnext 域都指向下一个 FragSdu,最后一个 FragSdu 的 pnext 指向第一个 FragSdu,这样形成的队列实际上是一个单向闭合串表。

三个全局指针:

```
FragSdu * txQFreeList; /* 指向未使用的 FragSdu */
FragSdu * txQFirstList; /* 指向队首的 FragSdu */
FragSdu * txQPutList; /* 指向将被释放的 FragSdu */
```

关于队列的整个操作如下:

1) 在收到上层模块 (Prepare-MPDU) 发来的消息后,将消息中的 FragSdu 提取出来,复制到 txQFreeList 所指向的 FragSdu 中,将队列中的该 FragSdu 标记为“已用”(令结构体的 fUse 域为 1),让 txQFreeList 指向队列中的下一个 FragSdu。

2) 在一个 MSDU 传输完毕后,判断 txQFirstList 所指向

的 FragSdu 的 fUse 域是否为 1,若为 1,则说明该 FragSdu 中是有效数据,提取出该 FragSdu,发送至下一模块。令 txQPutList 指向该 FragSdu,便于在该 FragSdu 传输完成后释放其所指向的内存块,令 txQFirstList 指向下一个 FragSdu。

3) 收到传输完毕的确认消息后,释放 txQPutList 指向的 FragSdu 对应的内存块,将该 FragSdu 标为“未用”(令 fUse 域为 0)。

4 MAC 层闭合测试

目前,本方案只涉及到数据收发的软件实现,尚未和物理层及 LLC 层联合调试。我们采用 MAC 层自身闭合测试的方法来验证所做的工作,模拟了 MAC 层与物理层的接口 PHY-SAP,以及 MAC 层与 LLC 层的接口 MAC-SAP。在 MAC-SAP 的模块任务中,循环产生随机数发送到 MAC 层,并且接收来自 MAC 层的数据。在 PHY-SAP 的模块任务中,收发物理层数据传输请求及确认信息,将数据的收发联系起来。测试时,启用了 RTS/CTS 机制,产生的随机数经过 MAC 层发送端的成帧、分段、排队、计算 CRC、发送,再经过接收端的相反操作后,能成功复原,并且观察其中的每一步结果都是正确的。

结束语 系统中基本数据类型的定义和函数的实现都经过反复的测试,以适合于整个系统。在队列缓冲区的实现中,经验证最好采用 uCOS-II 系统自带的动态内存分配函数,虽然这样整个代码的可移植性有所降低,但是保证了系统在动态分配内存时,执行时间的可确定性,提高了系统的稳定性。在对连续信号的处理及整个代码结构的设计上尽量避免延时,减小系统开销,提高执行效率。

在今后的研发中,我们将系统中对速度要求比较严格的模块采用硬件来实现,即整个系统采用嵌入式芯片加逻辑电路^[6]来实现,这样会使系统运行的效率更高。

参考文献

- 1 IEEE Standards Board. 802 part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE Standard 802.11, 1999
- 2 Labrosse J. J. 邵贝贝,等译. 嵌入式实时操作系统 uC/OS-II 北京:北京航空航天大学出版社,2003
- 3 宋贸强. 通信软件设计基础[M]. 北京:北京邮电大学出版社, 2001
- 4 徐会明. 基于 IEEE802.11 协议的无线局域网 MAC 层研究与应用.[学位论文]. 武汉:华中师大,2001
- 5 任哲. 嵌入式实时操作系统 uC/OS-II 原理及应用. 北京:北京航空航天大学出版社,2005
- 6 沈力为. IEEE802.11 无线局域网 MAC 层与物理层基带处理芯片的研究.[学位论文]. 上海:复旦大学,2004