

基于 NS-2 的自适应差额 IEEE 802. 11 PCF 轮询机制

廖 勇¹ 杨士中¹ 徐昌彪²

(重庆大学通信与测控研究所 重庆 400044)¹ (重庆邮电大学通信与信息工程学院 重庆 400065)²

摘 要 在研究无线局域网 IEEE 802. 11 标准点协调功能 PCF(Point Coordination Function)机制不足的情况下,针对其中 Round Robin 轮询机制存在的 QoS(Quality of Service)不能得到有效保障的问题,提出了一套自适应差额轮询调度算法解决方案,阐述了算法的核心思想,介绍了调度机制的执行过程,并且通过修改 PCF 源代码,在网络模拟器 NS-2 平台上对算法进行了脚本模拟。实验结果表明,自适应差额 IEEE 802. 11 PCF 轮询算法比传统的 Round Robin 算法在端到端延迟、系统吞吐量和包延迟等 QoS 服务性能方面都有显著的提高。

关键词 无线局域网,点协调功能,自适应差额轮询,网络模拟器,服务质量

中图分类号 TN925 **文献标识码** A

Adaptive Deficit IEEE 802. 11 PCF Polling Scheme Based on NS-2

LIAO Yong¹ YANG Shi-zhong¹ XU Chang-biao²

(Institute of Communication & Tracking Telemetry Command, Chongqing University, Chongqing 400044, China)¹

(College of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)²

Abstract A adaptive deficit polling scheme was put forward which aimed to guarantee the QoS(Quality of Service) of Round Robin polling scheme under the condition of inefficient WLAN IEEE 802. 11 PCF (Point Coordination Function) mechanism. The core idea of this scheme was described as well as the processing of adjustment mechanism, and it was simulated on the platform of network simulator NS-2 by revising PCF source code. The results indicated that adaptive deficit IEEE 802. 11 PCF polling scheme is prior to Round Robin scheme with its better performance of QoS(Quality of Service) which includes end-to-end delay, system throughput, packet delay, etc.

Keywords WALN, PCF, Adaptive deficit polling scheme, NS-2, QoS

IEEE 802. 11 MAC(Medium Access Control)有两种接入方式^[1]:DCF(Distributed Coordination Function)和 PCF(Point Coordination Function)。在这两种接入方式中,DCF 是必选的,PCF 是可选的。DCF 主要应用于无实时要求的业务,PCF 可以满足实时性业务,在提高系统 QoS(Quality of Service)方面,PCF 能起到关键的作用。

但是 IEEE 802. 11 标准中 PCF 接入机制本身有 4 大缺陷^[2,3]:(1)集中控制问题。(2)Round Robin 调度机制问题。现有的 PCF 采用 Round Robin 调度机制,PC 根据 Polling 列表轮询无线节点,因空的 Polling 造成信道的浪费。第二个问题是轮询开销大。在 CFP 阶段,几乎每个数据帧的传输都需要 AP 的干预,尤其是当 AP 轮询到的节点没有数据要传输时,更是造成信道的浪费。要提高信道利用率,就要根据实时业务状态动态调整 Polling 列表,减少空的 Polling。(3)Beacon Delay 问题。(4)与 DCF 机制的结合问题。本文就是针对第二个 Round Robin 调度机制问题,提出了一套自适应差额轮询机制的解决方案,并且在 NS-2 平台上进行了算法的模拟,得出其 QoS 性能提高的结果,从而验证了自适应差额

PCF 算法的可行性。

1 自适应差额轮询调度机制解决方案

在 CFP 期间,一个基本的 BSS 站点内有以下几种类型的站点:

- (1)PC;
- (2)CF-Pollable STA;
- (3)非 CF-Pollable STA,但是是 Active 的 STA;
- (4)非 CF-Pollable STA,但是是 Power Save 的 STA。

这几种 STA 是同时存在的,并且由于在 CFP 阶段采用的是 PCF 控制,因此在这些站点中起核心作用的是 PC,它会控制媒介的使用,调度查询可轮询 STA,以及向非可轮询站点传输数据。PC 可以发送广播帧或多目标传输帧,定向的数据帧可管理帧到任何处于活动状态的 STA,也可以发送这些帧到无竞争可轮询以及处于节能模式的 STA。

自适应差额动态算法是先将所有的站点进行一个类型的划分,由于 PC 在 CFP 期间起核心控制作用,将 PC 站划为第一优先级的站点类型,即 Class1。而将剩余的 CF-Pollable、非

到稿日期:2008-12-22 返修日期:2009-03-02 本文受国家发改委 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(CSTC,2006BB2164)资助。

廖 勇 博士研究生,助教,研究方向为无线网络通信,E-mail:bravemanly@163.com;杨士中 教授,博士生导师,中国工程院院士,研究方向为无线网络通信、测控及遥感信息传感;徐昌彪 博士后,教授,研究方向为宽带无线接入和网络性能分析。

CF-Pollable(包括 Active 和 Power Save)的 STA 划分为第二优先级的站点,即 Class2。然后在 Class1 类型的站点 PC 内,将根据 PC 要发送给非 PC 站点的帧的类型,即根据发送到非 PC 的帧中的帧控制域里面的类型子域来差额地分配权重和量子。具体地说就是,PC 里面的所有数据分成 3 个服务队列,这 3 个服务队列是按照帧控制域里面的类型子域来划分的。具体的类型划如见表 1 所列。

表 1 帧类型及子类型

帧类型值 B2B3	类型描述
00	管理帧
01	数据帧
10	控制帧
11	保留值

队列划分完之后,在算法调度器中,每个服务队列都有一个与之相联的量子(quantum)值和差额(deficit)计数器^[9]。前者为每一轮获得服务的平均字节数,后者则被初始化为前者的值,依此为每个非空队列服务。每轮调度的字节数原则上等于量子值;只要差额计数器大于零,服务队列便获得服务;分组获得服务后,差额计数器将减去该分组的字节数;当差额计数器不大于零时,队列不获得服务。每当新一轮开始时,所有非空队列的差额计数器都将增加其量子值。这样,通过给不同的队列设置不同的量子值,可以支持不同的预约带宽。这个调度算法可以避免由不同的队列使用不同长度的分组引起的不公平。更为重要的是,由于在 CFP 阶段实时数据帧发送的重要性,对数据帧队列采用实时跟踪此队列长度并以此来调整量子值。同时,在非 PC 站点对应的 Class2 类型中,包括可轮询站点和非可轮询站点,它们对应的在 CFP 阶段的发送帧相对来说就比较单一。在可轮询站点中,发送帧包括 Data+CF-Ack,CF+Ack,NULL 等,而非可轮询站点的发送帧只是简单的一个 ACK。所以对 Class2 站点的服务队列的处理也比较简单,也是每个队列对应一个与之相联的量子(Quantum)值和差额(Deficit)计数器。

1.1 帧类型分配以及队列的划分

为了便于分析,假定在 CFP 期间在 BSS 中有 4 个站点,分别是 PC 一个、可轮询站点一个、非可轮询站点且活跃站点一个、非可轮询且节能站点一个。PC 属于 Class1,其它 3 个站点属于 Class2。然后对 PC 下属的服务队列按照业务类型(ToS; Type of Service)分组,可以分为实时数据服务队列、控制服务队列和管理服务队列。基本的服务队列参数如表 2 所列。

表 2 自适应差额调度队列基本参数

站点分类									
Class1(PC)					Class2(非 PC)				
ToS									
数据队列				控制 队列	管理 队列	可轮 询站 点队 列	非可 轮询 且活 跃站 点队 列	非可 轮询 且节 能站 点队 列	
队列 1(Variable)				队列	队列 3	队列 4	队列 5	队列 6	
长度 (byte)	<20000 [20000,30000) ≥30000								
权重	6.0	7.0	8.0	5.0	4.0	3	2	1	
量子 (byte)	13284	16128	18432	11520	9216	6912	4608	2304	

1.2 相关概念以及动作

队列长度 L_i 是指队列 i 的实时长度(单位为 byte)。在表 2 中, class1 设定了 2 个阈值:20000byte 和 30000byte。将队列 1 的长度分为 $L_1 < 20000\text{byte}$, $20000\text{byte} \leq L_1 < 30000\text{byte}$, $L_1 \geq 30000\text{byte}$ 3 段。

权重 W_i 是指队列 i 的权重。队列 i 被分配的带宽比例为 $W_i / \sum W_i$ 。自适应 DRR 的动态就在于在有特定 QoS 要求的队列中,权重能够根据用户的要求进行改变,从而达到调整带宽的目的。在表 1 中 W_1 根据队列 1 的实时长度而变化。

量子 Q_i 反映了队列 i 每一次循环希望发送的平均字节数:

$$Q_i = W_i * Lmtu$$

其中, $Lmtu$ 是最大传送单元的字节数。在表 2 中, $Lmtu = 2304\text{byte}$ 。注:如果帧实体没有加密,则其最大长度是 2304 字节;如果帧实体内容被加密,则其最大长度是 2312 字节。在 IEEE 802.11 协议中由于没有考虑加密算法,因此这里取最大传输帧实体单元为 2304 字节。

差额计数器(D_i)对应于每一个队列,用来控制每一个队列分组的发送^[9],其动作如下:

- (1) 空队列时为零。
- (2) 队列开始获得服务时,该队列的差额计数器初始化为量子值。
- (3) 队列中的分组获得服务后,差额计数器减去已服务分组的长度。只要差额计数大于 0,该队列就能获得服务,否则就轮到下一个队列获得服务。
- (4) 每当新一轮开始时,所有非空队列的差额计数器都将加上其量子值。
- (5) 当数据队列 1 的队列长度的变化越过设定的阈值时,其权值和量子值都要改变。

1.3 无竞争轮询列表

自适应差额 PCF 机制按照如下的规则对轮询列表进行轮询:

(1) 轮询列表的处理

对与含有 PC、可轮询站点、非可轮询站点(包括活跃站点和非可活跃站点)在 CFP 期间的处理服务顺序如下:

- ① PC 先轮询处理完所有的可轮询站点;
- ② PC 处理完所有可轮询站点之后,如果 CFP 时间还未到,那么接着处理的是非可轮询且处于活跃状态的站点;
- ③ 在处理完非可轮询且活跃的站点后,再处理非可轮询

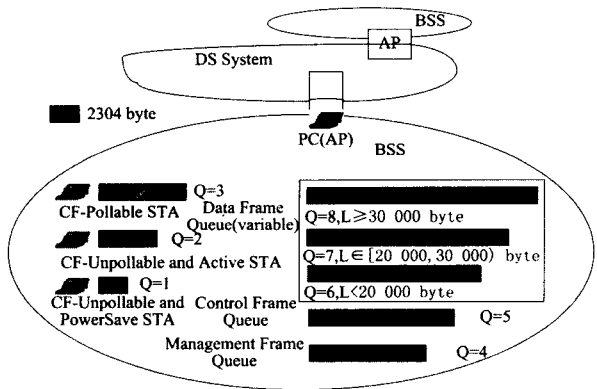


图 1 自适应差额调度算法实例图

且节能的站点;

④ 处理完所有 3 类站点之后,服务顺序从①开始。

(2) 轮询列表更新规程

STA 在关联和重新关联请求真的容量信息域中的 CF-Pollable,其子域中表示是否支持被轮询。如果一个 STA 希望改变 PC 对其有关轮询能力的记录,则该 STA 必须执行重新关联过程。在关联期间,无竞争可轮询 STA 可以通过容量信息域的 CP-Poll Request(无竞争轮询请求)位置为“(true)真”,请求 PC 将其放入轮询列表中。

如果某个无竞争可轮询 STA 没有在列表中,而且在它最近关联中也没有请求被轮询,它可能被 PC 动态置于列表中,以处理由它引发的大量的帧传输。

2 PCF 机制在 NS-2 中的实现

实验过程中借鉴了 Anders Lindgren^[4,5] 等人在 NS-2^[6] 下实现 IEEE 802.11 PCF 机制的思想,在其实现 PCF 机制的已有 mac-802_11.h(cc),mac-timers.h(cc),mac.h 和 ll.c 文件上进行了修改,然后把修改.cc 和.h 文件编译到 NS2.28 内核,添加脚本语言实现模拟。

在 ll.cc 里面主要包括了 LLHeaderClass,LLClass 和 LL 的实现。LLHeaderClass 继承 PacketHeaderClass,实现自己的构造函数。LLClass 类继承 TclClass,实现自己的类初始化。在 LL 类中,通过 command 实现了与 Tcl 脚本的交互。

```
static class LLHeaderClass:public PacketHeaderClass
{
public:
    LLHeaderClass(); PacketHeaderClass("PacketHeader/
    LL",sizeof(hdr_ll))
    {
        bind_offset(&hdr_ll,offset_);
    }
} class_hdr_ll;
static class LLClass:public TclClass
{
public:
    LLClass();TclClass("LL") {}
    TclObject* create(int,const char* const*)
    {
        return (new LL);
    }
} class_ll;
```

在 mac-802_11.cc 里面首先定义了 PHY_MIB 和 MAC_MIB 的全局变量,实现了 Mac802_11Class 与 TclClass 类的挂钩链接,实现了 Mac 功能函数:接口函数、网络分配矢量函数、接收函数、发送函数、延时函数、退避函数以及信标帧函数。在 mac-802_11.cc 里面最重要的部分就是 ADPCFList 的实现。在 ADPCFList 里面实现了以下函数:

(1) int ADPCFList::getNextPollable(),PC 获取下一个可轮询站点的 id,如果列表里面没有可用节点,那么返回-1。

(2) void ADPCFList::insertNode(int macId,int cfPollable,int cfPollReq);macId 是插入节点的 id 值。cfPollable 显示这个节点是否可以被轮询,1 表示可以,0 表示不可以;cfPollReq 表示这个节点是否将被轮询,1 表示可以,0 表示不可

以。

(3) int ADPCFList::deleteNode(int macId):从轮询列表里面删除节点,成功返回 0,否则返回-1。

(4) void ADPCFList::initPoll():初始化 PCF 列表。

(5) void ADPCFList::enterClass():这个里面包括将 BSS 里面的站点进行分类,然后根据站点的情况分为两个 Class;Class1 是 PC,Class2 是可轮询站点和非可轮询站点(包括 active 和 power save 状态)。在 Class1 里面也就只有单独的 PC 站点,然后根据 PC 所拥有的所有帧的 ToS 再进行分类,分成数据、控制和管理队列。数据队列将会进入核准权重和量子值处理函数 int ADPCFList::assignWeight(int length)进行处理,其它的队列直接进入队列长度统计函数 int ADPCFList::countQueueLength()进行处理。

(6) int ADPCFList::assignWeight(int length):对数据队列进行权重的分配,根据传进去队列的长度 length 分配系统带宽也就是对应的量子值,返回对应处理好的队列的量子值。

(7) void ADPCFList::exitQueue():这个暂时只针对数据队列。出队的时候,将量子值加到差额计数器,然后发送一个分组,差额计数器减去被发送分组的长度,如果这时差额计数器小于 0,那么再核准权重和量子值,统计队列长度,结束。如果差额计数器大于 0,表明队列里面还有数据帧要传输,那么继续发送。

(8) int ADPCFList::countQueueLength():统计队列里面帧的长度和。

(9) int ADPCFList::numPollable():返回可被轮询的站点数目。

在修改的 mac-802_11.cc 里面自适应轮询调度算法主要是通过以下 3 个函数实现的:

```
void ADPCFList::enterClass()
int ADPCFList::assignWeight(int length)
void ADPCFList::exitQueue()
```

3 实验及其结果分析

实验的平台是 Windows XP SP2,Cygwin 下安装的是 ns-allinone-2.28。改后的 mac-802_11.h(cc),mac-timers.h(cc),mac.h 和 ll.c 等文件放置在 ns-2.28/mac 文件夹里面,然后在 ns-allinone-2.28 根目录下面进行./install,编译安装成功,表明自适应差额轮询调度机制模块已成功嵌入到 ns-2.28 里面。

3.1 实验拓扑图

在实现的脚本语言文件上,设置了 8 个无线节点 0-7,其中 0,2,4,6 节点都是 UDP 代理,并 CBR(Constant bitrate)承接在 UDP 代理之上。单数无线节点则是 LossMonitor 代理,主要完成跟踪接收到的数据包的任务。

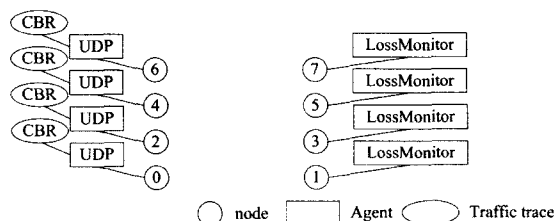


图 2 实验拓扑图

3.2 参数设置

修改之后的 mac-802_11.h 里面有 PHY 和 MAC 默认参数设置,如表 3 所列。

表 3 802.11 规格默认参数设置

PHY_MIB		
Parameters	Values	Notice
DSSS_CWMin	31	
DSSS_CWMax	1023	
DSSS_SlotTime	0.000020	20us
DSSS_CCATime	0.000015	15us
DSSS_RxTxTurnaroundTime	0.000005	5us
DSSS_SIFSTime	0.000010	10us
DSSS_PreambleLength	144	144bits
DSSS_PLCPHeaderLength	48	48bits
MAC_MIB		
MAC_RTSThreshold	3000	bytes
MAC_ShortRetryLimit	7	retransmissions
MAC_LongRetryLimit	4	retransmissions
MAC_Fragmentation Threshold	2346	bytes
MAC_MaxTransmitMSDULifetime	512	time units
MAC_MaxReceiveLifetime	512	time units

PCF 的主要参数设置如表 4 所列。

表 4 PCF 参数设置

Beacon length	40Byte
CF-Poll length	34Byte
CF-end length	20Byte
CF-Null length	34Byte

实验中的节点配置参数如表 5 所列。

表 5 节点参数设置

channel type	Channel/WirelessChannel
radio-ropagation model	Propagation/TwoRayGround
network interface type	Phy/WirelessPhy
MAC type	Mac/802_11
interface queue type	Queue/Drop Tail/PriQueue
link layer type	LL
untenna model	Antenna/OmniAntenna
routing protocol	DSDV

在实验中采用的是 DSSS 扩频技术,传输天线和接收天线的高度都是 1.5m,传输天线和接收天线的增益是 1,物理层数据传输速率是 11Mb。

采样记录的时间是 0.9s,节点 0 在 $t=1.4s$ 时开始传输,节点 2 在 $t=10s$ 时开始传输,节点 4 在 $t=20s$ 时开始传输,节点 6 在 $t=30s$ 时开始传输,整个模拟在 $t=80s$ 时结束。

3.3 统计结果的绘制以及结果分析

我们分别把采用普通 IEEE 802.11 默认 MAC 轮询机制和 ADPCF 算法机制得到的 trace 文件进行了统计,把对应节点 0-1,2-3,4-5,6-7 对应的 trace 文件用 awk^[7] 进行了处理,然后在 gnuplot^[8] 下对处理后的文件进行绘制,对应吞吐率、端到端延迟和丢包率的对比结果如图 3—图 5 所示。

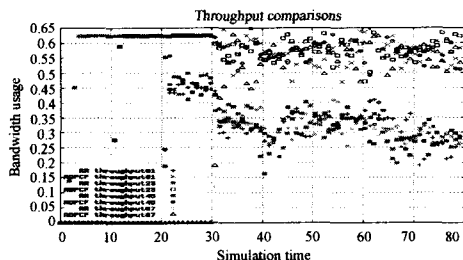


图 3 吞吐量对比

3.3.1 吞吐量对比分析

(1) 共性:对于 RR 机制和我们设计的 ADPCF 调度算法而言,无线节点 0 在时间 $t=1.4s$ 时开始传输,节点 2 在时间 $t=10s$ 时开始传输。在时间段 $[1.4s, 10s]$ 之间,节点 0 是唯一一个传输数据的节点,占用整个可用带宽。这证明节点 0 在特定的时间间隔内具有很高的效率。在 $t=10s$ 时,节点 2 开始传输,因此和节点 0 一起分享信道资源,位传输率造成很大的下降。除此之外,随着传输节点的增加,位传输率经历更大的振荡和下降。

(2) 差异:ADPCF 可以在相同的情况下比传统的 RR 轮询机制具有更高的吞吐量。可以看到,即使在增加节点分享系统带宽的时候,对每个发送节点其吞吐量都下降得不是很大。在 $t=20s$ 时,也就是在节点 0 和节点 2 已经传输的情况下,节点 4 开始传输数据包的时候,传统的 RR 轮询下的节点的吞吐量比 ADPCF 算法下的吞吐量有大幅度的下降。在 $t=30s$ 时,节点 6 开始向无线节点 7 传输 UDP 数据,传统的 RR 轮询机制下节点的吞吐量又有较大幅度的下降。而 ADPCF 机制下的节点保持一个比较稳定而且比较高的系统吞吐量的分享状态。

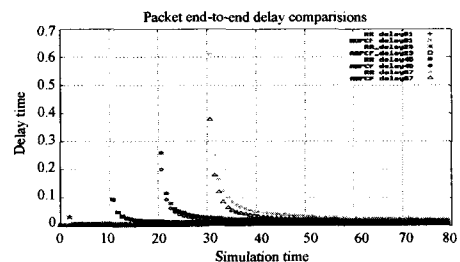


图 4 端到端延迟对比

3.3.2 端到端延迟对比分析

(1) 共性:随着分享网络资源的节点增加,端到端延迟开始增加并且每个节点调整 CW 的时间也变得更长。

(2) 差异:整体来说,ADPCF 机制比传统的 RR 轮询机制有更低的端到端延迟特性。不过,注意到在 $t=0.4s$ 时,节点 0 开始向节点 1 传输数据, $t=10s$ 时节点 2 开始向节点 3 传输数据的时候,传统的 MAC 轮询机制下的节点之间的端到端延迟几乎为 0,而在 $[0.4s, 20s]$,也就是在节点 0 开始传输到节点 4 开始传输之间,ADPCF 调度算法在端到端延迟的性能上不及传统 RR 轮询机制。但是从 $t=30s$ 之后起,节点 6 开始向节点 7 传输数据,对应的 ADPCF 机制就开始在端到端延迟上具有明显的优势。也就是说,随着传输节点数目的增多,ADPCF 机制在端到端延迟性能上比传统 RR 轮询机制更好。

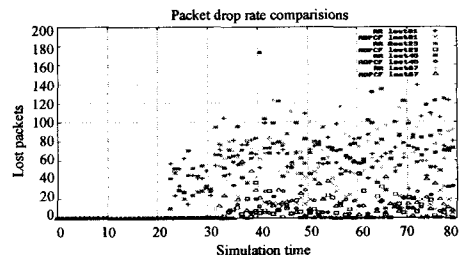


图 5 包丢失率比较

3.3.3 包丢失率比较分析

(1) 共性:图 5 表明随着分享网络资源的节点数目的增(下转第 96 页)

图6显示了采用 Matlab-Simulink 对不同算法进行自适应跳频同址干扰抵消误码率仿真的结果。

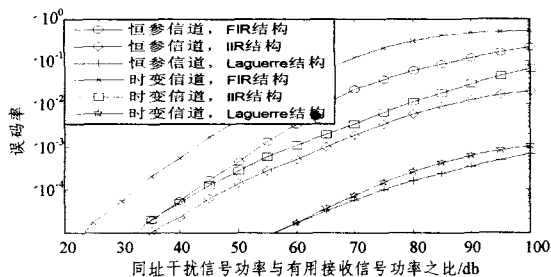


图6 自适应跳频干扰抵消的误码率性能比较

图6的误码率仿真结果表明在两种信道情况下,基于Laguerre滤波器的自适应跳频同址干扰抵消器均能获得更好的误码率性能,即便在同址干扰信号比有用信号强100dB时,仍能获得约 10^{-3} 的误码率性能,远优于其它两种方法的抵消效果,可以满足跳频数据通信的要求。FIR结构的自适应干扰抵消器在同址干扰强的时候,误码率大于 10^{-1} ,已经基本无法满足通信的正常进行。

总之,采用Laguerre滤波器实现的自适应频率选择性跳频同址干扰抵消器,无论在恒参信道还是时变信道,均能获得更好的干扰抵消性能和稳定性,满足跳频电台多台同址通信的需求,是一种非常有效的方案。

结束语 针对FIR结构和IIR结构的自适应干扰抵消器在时变长脉冲响应干扰应用中的不足,提出一种采用Laguerre滤波器实现的自适应跳频同址干扰抵消器,在讨论抵消器结构和最优极点估计方法的基础上给出了实现该干扰抵消器的算法。研究结果表明,该干扰抵消器能够有效抑制频率选择性跳频同址干扰;相同的滤波器长度时,在两种信道条件下均能取得比FIR和IIR结构干扰抵消器更好的抵消性能和稳定性。该方法实现简单,有着广泛的应用前景,一种基于该方法的自适应跳频同址干扰抵消器正在研制之中。

(上接第39页)

加,包丢失率也随着增加。

(2) 差异:在传统的RR轮询机制下,[1.4s,10s]间隔之内,丢包率是0。这个很容易理解,因为在这个时间段里,只有一个节点开始使用网络。但是随着更多的节点开始分享网络资源,丢包率也开始增加。ADPCF机制下,在[0s,30s]之间,对于已经开始传输数据的节点0,2,4都没有包的丢失率,这个性能比传统RR轮询机制延迟了10s的时间,并且在各节点出现丢包的情况下,ADPCF机制下的包丢失率也比传统MAC轮询机制性能好。

从吞吐量、包延迟和端到端延迟这几个QoS性能的分析上可以得出,我们设计的ADPCF机制比传统的RR轮询调度算法有比较大的优势,从而从实验上验证了ADPCF机制的可行性及其与传统RR轮询算法在性能比较上的优越性。

结束语 针对传统的Round Robin调度机制不能有效地提高无线传输系统QoS的情况,本文提出的自适应差额轮询IEEE 802.11 PCF调度算法机制在NS-2平台上进行了模拟,得出本算法在系统吞吐量、端到端延迟以及包丢失率等QoS性能方面比传统Round Robin算法有较大的提高。

参考文献

[1] IEEE Standards Board. 802 part 11: Wireless LAN Medium Ac-

参考文献

- [1] Allsebrook K Dr, Rhble C Mr. VHF cosite interference challenges and solutions for the United States Marine Corps' expeditionary fighting vehicle program[C]// 2004 IEEE Military Communications Conference, 2004:548-554
- [2] Jagasse L M, Lexington MA. Cosite Interference Rejection System Using an Optical Approach[P]. U. S. 7231151B2. 2007
- [3] Maxson B D. Optimal Cancellation of Frequency- Selective Cosite Interference[D]. University of Cincinnati, 2002
- [4] 丛卫华, 刘孟庵, 惠俊英. 自适应宽带多途干扰抵消的实时处理算法[J]. 声学及电子工程, 1999(3): 1-10
- [5] Simon, Haykin. Adaptive Filter Theory[M]. Beijing: Publishing House of Electronics Industry, 2003:728-734
- [6] Xu Sun, Guang Meng. Steiglitz-Mcbride type adaptive IIR algorithm for active noise control[J]. Journal of Sound and Vibration, 2004:441-450
- [7] 黄高勇, 张家树. 一种抑制直扩通信窄带干扰的新型非线性自适应预测滤波器[J]. 电子与信息学报, 2007, 29(6): 1328-1331
- [8] Jing Yuan. Adaptive Laguerre Filters For Active Noise Control [J]. Applied Acoustics, 2007, 68(1): 86-96
- [9] Kowalski A M. Wideband co-site interference reduction apparatus[P]. U. S. 6693971. 2004
- [10] Lin S, Harris J G. Combined LMS/F algorithm[J]. Electronics Letters, 1997, 33(6): 467-468
- [11] Masnadi-Shirazi M A, Zollanvari A. Complex Digital Laguerre Filter Design With Weighted Least Square Error Subject to Magnitude and Phase Constraints[J]. Signal Processing, 2008, 88(1): 796-810
- [12] den Brinker A C, Sarroukh B E. Pole Optimisation in Adaptive Laguerre Filtering[C]// IEEE International Conference on Acoustics, Speech, and Signal Processing. 2004: 649-652
- [13] 王东昱, 张欣, 杨大成. 多径信道的自适应估计[J]. 北京邮电大学学报, 2006, 29(4): 36-40

cess Control (MAC) and Physical Layer (PHY) specifications [S]. IEEE Standard 802. 11. 1999

- [2] Kopsel A, Ebert J, Wolisz A. A performance comparison of point and distributed coordination function of an IEEE 802. 11 WLAN in the presence of real-time requirements[C]// Proceedings of Seventh Int'l Workshop on Mobile Multimedia Communications, Oct. 2000
- [3] Visser M A, El Zarki M. Voice and Data Transmission over an 802. 11 Wireless Network[C]// Proceedings of the 6th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'95). Sept. 1995: 648-652
- [4] Lindgren A, Almqvist A, Schelén O. Quality of Service Schemes for IEEE 802. 11, A Simulation Study[M]// Wolf L, et al., eds. IWQoS2001. Berlin Heidelberg: Springer-Verlag, 2001: 281-287
- [5] Lindgren A. Contributions to the network simulator: Support for the PCF mode of IEEE802. 11 forms-2. 1b8[EB/OL]. <http://www.sm.luth.se/~dugdale/index/software.shtml>
- [6] NS-2NetworkSimulaotr[EB/OL]. <http://www.isi.edu/nsnam/ns>
- [7] <http://linux.chinaunix.net/doc/program/2005-02-03/1085.shtml>
- [8] http://gnuplot.sourceforge.net/docs_4.1/node1.html
- [9] 孙力娟, 吴斌, 张登银. D-DRR: 一种可动态改变带宽的分组调度算法[J]. 江苏通信技术, 2004, 20(1)