

无线网络中 MAC 层违规行为的检测和惩罚

吴春辉, 郑淑丽, 侯整风

(合肥工业大学 计算机与信息学院, 安徽 合肥 230009)

摘要: 802.11 无线媒体访问控制(MAC)协议利用分布式争夺解决机制处理无线信道的共享。在这种环境下, 那些没有执行 MAC 协议的节点可以获得不公平的信道带宽。IEEE 802.11 要求节点等待一个随机时间间隔之后再竞争访问信道。如果某些节点等待一个较小的时间间隔, 那么对其他普通节点来说, 这是不公平的。针对这种情况对 IEEE 802.11 MAC 协议做简单的改进来检测这样的违规行为并对其进行惩罚。

关键词: 802.11 协议; 媒体访问控制协议; 分布式协调功能模式

中图分类号: TN925⁺.93

文献标识码: A

文章编号: 1673-629X(2006)11-0222-03

Detection and Punishment of MAC Layer Misbehavior in Wireless Networks

WU Chun-hui, ZHENG Shu-li, HOU Zheng-feng

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

Abstract: Wireless medium access control (MAC) protocols such as IEEE 802.11 use distributed contention resolution mechanism for sharing the wireless channel. In this environment, some hosts that fail to adhere to the MAC protocol may obtain an unfair share of the channel bandwidth. IEEE 802.11 requires nodes competing for access to the channel to wait for a "backoff" interval, randomly selected from a specified range, before initiating a transmission. Selfish nodes may wait for smaller backoff intervals than well-behaved nodes, thereby obtaining an unfair advantage. The article presents modifications to the IEEE 802.11 protocol to simplify detection and punishment of such behaviors.

Key words: IEEE 802.11; MAC; DCF mode

0 引言

IEEE 802.11^[1,2]无线媒体访问控制协议(MAC)利用分布式争夺解决机制处理共享无线信道。争夺的解决方案是基于协作机制以确保所有的节点能公平、合理地使用信道。在这种情况下, 网络中一些节点没有按照协议执行, 企图获取不公平的信道占用, 使普通节点的吞吐量降低。因此, 检测和惩罚违规行为是非常必需的。

IEEE 802.11 MAC 协议^[1]有两种争夺解决模式: 有中心的机制 PCF(接入点协调功能)和全分布的机制 DCF(分布式协调功能)。PCF 需要一个中心控制器, 并且只能应用在有框架网络中(PCF 模式是 802.11 的可选模式)。DCF 可以被应用在有框架的无线网络和自组织临时网络(Ad Hoc)中。文中只讨论在 DCF 模式下的违规行为。

1 IEEE 802.11 DCF 模式简介

DCF 模式利用 CSMA/CA(载波监听多路访问/冲

突避免)来解决信道的争夺。发送方在发送数据前先从 $[0, CW]$ (CW 为争夺窗口)中随机选择一个值作为自己退避计数器的初始值, 每个节点都有自己的 CW , 并且是可变的。当信道空闲时, 退避计数器每过一个时间间隔(时间间隔的大小是 IEEE 802.11 定义的)减 1, 直到为零时, 节点便可以访问信道; 而当信道忙时, 计数器的值不变。退避计数器的值减小到零, 发送方可以通过交换控制包预约信道。发送方先发送一个 RTS(请求发送)包给接收方, 接收方响应 CTS(清除发送)包, 这是为了传输数据而预约信道。两个包都包括一个时间字段, 其他听到 RTS 的节点延迟一段时间(由包里的时间段决定), 等待接收方返回 CTS; 接收方接收到 RTS 之后响应 CTS, 而其他听到 CTS 的节点必须延迟一段时间(CTS 中的时间段)等待本次传输完成。当 RTS/CTS 成功交换后, 发送方发送数据, 接收方响应 ACK(确认)包确认已经成功接收到数据包。如果发送节点的数据传输成功后, 重设它的 CW 到最小值 CW_{min} , 否则 CW 设为两倍, 直到最大值 CW_{max} 。

由于无线局域网采用的是开放式接入, 如果节点采用的是 802.11 协议, 就可以接入, 而对协议内部的参数并没有严格的限制, 这样一些违规节点可以通过以下行为, 获得不公平的带宽:

收稿日期: 2006-02-10

作者简介: 吴春辉(1979-), 男, 山西大同人, 硕士研究生, 研究方向为无线网络; 侯整风, 教授, 硕士生导师, 研究方向为计算机网络与信息安全。

1)从平均退避值比较小的分布中选择退避值,而不是按照 DCF 定义的(例如,从范围 $[0, CW/4]$ 中选,而不是 $[0, CW]$)。

2)用不同的重传策略,当冲突时 CW 不变。

这样的违规行为严重地降低了普通节点的吞吐量。

近来对网络中违规行为的研究已经在网络层展开^[3~5],一种方法是确定违规节点并避免这样的节点路由^[6];另一种方法是设计协议通过惩罚违规行为来鼓励协作^[7]。网络层机制只能识别网络层的违规行为,例如违规节点对路由发现,对丢失、延迟或错误路由数据包等的干涉。文中仅说明媒体访问控制层(MAC)的违规行为,也是对网络层机制的补充。

2 设计思路

在 TCP 协议中,文献[8]发现了 TCP 冲突控制算法的弱点,通过发送方改变冲突控制算法的执行,可以使接收方获得比其他接收方更大的带宽,文献[8]对此算法做了一些简单的修改来阻止接收方的优势,文中对 802.11 协议的修改是基于文献[8]的思想。

这里定义下列名词:

违规节点:完全不按照 MAC 协议执行的节点。

违规行为:某些节点由于某些情况而没有执行 MAC 协议的行为(只有几次没有执行 MAC 协议行为的节点并不一定是违规节点)。

发送方:发送数据的节点。

接收方:从发送方接收数据的节点,接收方监视发送方并检测它是否有违规行为。

一个节点既可以是发送方,也可以是接收方,在 IEEE 802.11 DCF 模式下,发送数据传输之前先进行 RTS-CTS 控制包交换(RTS-CTS 控制包在 DCF 模式下是可选的)。

这里对 IEEE 802.11 DCF 模式做最小的改动,并且允许接收方提前检测发送方的违规行为。众所周知,有基础设施的无线网络在实际中应用已经很广泛,由基站组成,通过基站与其他无线节点连接。基站由网络服务提供商来维护,所以接收方是可信的。因为基站的行为都是正常的,所以在发送信息的时候不会出现违规行为。另一方面,在 DCF 模式下的发送节点却是不可信的,可能通过违规行为获得更高的吞吐量。因此,作为接收方的基站应该检测发送方的违规行为。假设:1)接收方都是可信的;2)发送方和接收方之间没有约定(收发双方恶意阻塞网络)。当然接收方也可能有违规行为(Ad Hoc 网络),文中不考虑这种情况。

2.1 设计方案

在 IEEE 802.11 DCF 模式下对 MAC 层违规行为进行检测和惩罚。在 IEEE 802.11 协议中,发送方先从 $[0, CW]$ 的范围中随机选择一个退避数作为退避计数器的初始值,当退避数为零时,才能发送数据。因此,接收方不能

凭借从发送方观测到的退避数来判断发送方是否有违规行为,因为不能确定退避数是否是随机选取的还是选择一个比较小的。有一种方法,可以把接收方接收到的包放在一个长队列里,然后根据这些包的信息来判断发送方是否违规,这在理论上行得通,实际中却引入了很大的延迟(需要根据对数据包的统计数据来判断)。而且,当节点的移动性很高的时候,这种方法更不可行。因此,通过对 802.11 协议的修改,使接收方能在较短的观测时间内判断发送方是否有违规行为。具体方法如下:当发送方第一次给接收方发送之前,先随机选择一个退避数来初始化它的退避计数器,接收方收到发送方的 RTS 之后,随机选择一个退避数通过 CTS 或 ACK 包发送给发送方,发送方下一次利用这个退避数作为退避计数器的初始值,来给接收方发送数据。这样,接收方就可以通过观测发给发送方最后一个 ACK 包和发送方最近一次的 RTS 包之间的间隔来判断发送方是否违规,如果观测的空闲时间间隔小于分配的退避数,那么发送方有违规行为。当接收方察觉发送方两次传输之间等待的时间比分配给它的退避数小(不能通过一两次的行为来判断是否是违规节点),那么接收方将扩大分配给发送方的退避数作为惩罚。如果发送方没有遵守接收方对它的惩罚(延长退避时间),它成为违规节点的可能性更大。如果发送方遵守了惩罚,在吞吐量上不会获得明显优势。

2.2 具体实施

方案由两部分组成:一,接收方在接收完毕以后判断发送方是否有违规行为;二,接收方确定了发送方的违规行为,它就根据违规次数来惩罚发送方。

1)确定违规行为。

除了对退避方案的适当修改之外,其他都遵循 IEEE 802.11 DCF 模式规则。修改之后使发送方能根据接收方指定的退避数发送数据给接收方。例如,有两个节点:S(发送方),R(接收方),S 发送数据给 R,第一次 S 自己随机选择一个退避数发送数据,之后给 R 传输,S 必须使用 R 指定的退避数进行传输,如图 1 所示。

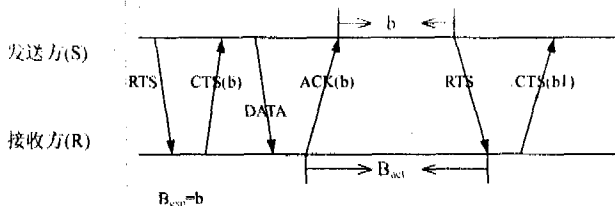


图 1 修改后的协议执行过程

B_{exp} 是从 R 从 $[0, CW]$ 中随机选择的一个退避数, B_{rcv} 是从 R 上一次 ACK 到 S 本次的 RTS 之间的空闲间隔,R 计算 B_{rcv} :

$$B_{rcv} < \alpha * B_{exp}, \quad 0 < \alpha \leq 1 \quad (1)$$

式(1)中 α 是一个系数,它是根据信道的环境不同而设置的,以减小错误判断的概率。

CW 的值反映的是与接收方通信的信道使用情况,对

于接收方来说,最初的 CW 值为 CW_{min} ,当没有冲突发生的时候,CW 一直保持 CW_{min} ,对于所有与 R 通信的 S 都是公平的;当有冲突发生的时候,CW 的值就变大,说明信道比较忙,所有与 R 通信的 S 将会获取一个比较大的退避数,从而等待一个比较长的时间,以避免发生冲突,提高算法的公平性。

本方案可以对发送方的包重传进行处理。在 RTS 包头加一个 number 字段,每当传输成功以后,就把该字段设为 1,传输不成功则加 1。当该次传输成功后,CW 设为 CW_{min} ,否则,对于第 i 次重传,CW_i 设为 $\min((CW_{min} + 1) * 2^{i-1}, CW_{max})$,这是 IEEE 802.11 定义的。图 2 描述了重传机制的执行。

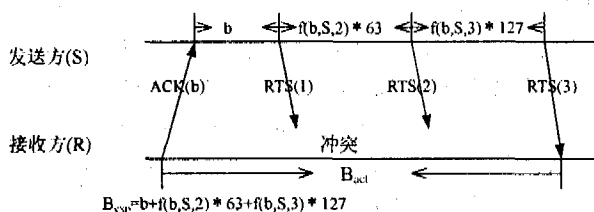


图 2 协议修改后的包重传执行过程

图 2 中 RTS 后面圆括号中的数字就是 number 字段的值。当 RTS 冲突时,发送方用接收方传给退避数(backoff)和决定函数 f ,发送节点号(nodeId),number 的值和 CW 共同产生一个退避数:

$$\text{new Backoff} = f(\text{backoff}, \text{nodeId}, \text{number}) * CW$$

函数 f 产生一个 0 到 CW 之间的数,确保冲突后产生一个不同的退避数^[9]。

RTS 冲突后,发送方用 f 产生一个新的退避数,避免再次冲突。接收方根据接收到的 RTS 包也能计算出一个退避数,并且两个退避数是相同的。当接收方成功接收到 RTS 后,可通过下式计算出 B_{exp} ,利用 B_{exp} 来判断是否违规行为:

$$B_{exp} = \text{backoff} + \sum_{i=2}^{\text{number}} f(\text{backoff}, \text{nodeId}, i) * CW_i$$

但另一个问题又产生,那就是 number 字段的值可能是不正确的,为了确保值的正确性,接收方可以监听信道的使用情况^[10]。如果信道冲突率很高,那么传输成功率将很低,这时有 number 值很低或传输成功率很高的 S 存在,接收方就会随机地丢掉一些来自 S 的 RTS 包,而 S 并不会知道是因为发生冲突而丢失的,还是被接收方丢掉的,那么 number 的值就会增大。而且偶尔丢弃 RTS 包不会影响 S 的吞吐量。

2) 惩罚方案。

有违规行为的节点可以比普通节点获得更大的吞吐量。惩罚方案是通过给 S(发送方)分配一个更大的退避数来惩罚 S 的违规。因此,当 R(接收方)检测到一个违规行为,它就计算公式: $D = \max(\alpha * B_{exp} - B_{act}, 0)$ 。从文献[9]的分析来看,对违规节点需要附加的惩罚,所以最后分配给 S 的退避数 $\text{Backoff counter} = \text{backoff} + D + \text{附加的惩}$

罚。同时,对所有有违规行为的节点都进行惩罚,其中也包括普通节点(由于环境的原因,前面介绍过),但对于普通节点,它的违规行为很少,所以惩罚对普通节点不会造成很大的影响。而当某个 S 的违规行为达到一定数量,就可以判断该节点为违规节点,可以通过网络层的机制拒绝接收该节点的任何数据包。

3 结束语

处理 MAC 层的违规行为对信道的公平共享非常重要。文中提出的思想,对 IEEE 802.11 协议作了简单的修改。以上方案是在假设接收方是可信的基础上建立的,在 Ad Hoc 网络中,接收方是不可信的,当接收方违反协议分配小或大的退避数给发送方的时候,可以采用与检测发送方类似的方法解决(用共同的初始化退避数函数)。如果发送方和接收方之间有企图,可以通过第三方软件来监视。对于一个节点用多个 MAC 地址来传输数据的违规行为的问题,有待于进一步探讨。

参考文献:

- [1] IEEE Standard for Wireless LAN - Medium Access Control and Physical Layer Specification[S]. P802.11,1999.
- [2] Clark D D. The Design Philosophy of the DARPA Internet Protocols[J]. ACM SIGCOMM Computer Communication Review,1995,25(1):106-114.
- [3] Yang Hao, Meng Xiaoqiao, Lu Songwu. Self-organized Network Layer Security in Mobile Ad Hoc Networks[C]//In First ACM Workshop on Wireless Security (WiSe). [s.l.]: [s.n.],2002.
- [4] Zhou L, Hass Z J. Securing Ad Hoc Networks[J]. IEEE Network,1999,13(6):24-30.
- [5] Bharghavan V, Demers A, Shenker S, et al. MACAW: a media access protocol for wireless LAN's[C]//ACM SIGCOMM Computer Communication Review. Proceedings of the conference on Communications architectures, protocols and applications SIGCOMM '94. [s.l.]: [s.n.], 1994:212-225.
- [6] Marti S, Giuli T J, Lai K, et al. Mitigating Routing Misbehavior in Mobile Ad hoc Networks[C]//Proceedings of the 6th annual international conference on Mobile computing and networking. [s.l.]: ACM Press, 2000:255-265.
- [7] Buttyán L, Hubaux J P. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks[J]. Mobile Networks and Applications,2003,8(5):156-163.
- [8] Savage S, Cardwell N, Wetherall D, et al. TCP Congestion Control with a Misbehaving Receiver[J]. In ACM Computer Communications Review,1999,29(5):71-78.
- [9] Kyasanur P, Vaidya N H. Detection and Handling of MAC Layer Misbehavior in Wireless Networks[J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2003.

(下转第 227 页)

须预测错误产生的条件和处理所产生的异常,当本地方法诊断出一个它无法解决的问题时,它应该将问题报告给 Java 虚拟机,产生一个异常,但 C 语言没有异常,此时须调用 Throw 或者 ThrowNew 函数来建立新的异常对象。在本例中,笔者从打开注册表项、调用 API 函数、注册表值项的类型等方面入手,对程序中可能出错的情况进行估计,并根据错误情况抛出异常。

例如:如果打开注册表项错误,则抛出以下异常:

```
if (RegOpenKeyEx (root, cpath, 0,
KEY_READ, &hkey)! = ERROR_SUCCESS)
{
    (* env) -> ThrowNew(env, (* env) -> FindClass(env,
"Win32RegKeyException"), "Open key failed");
    (* env) -> ReleaseStringUTFChars(env, path, cpath);
    return NULL;
}
```

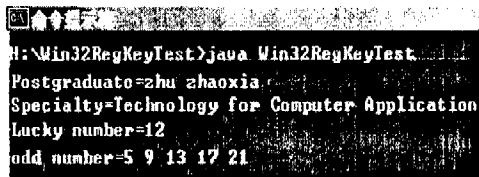
在本例中对出现的异常问题以统一的方式进行处理,不仅增加了程序的稳定性和可读性,而且规范了程序的设计风格,有利于保证程序的质量。

2.5 程序的运行

针对以上程序在 DOS 提示符依次运行如下命令:

```
javac Win32RegKey.java
javah Win32RegKey
javah WinRegKeyNameEnumeration
cl -l d:\jdk1.3.1-01\include -I d:\jdk1.3.1-01\include
\win32 -LD Win32RegKey.c advapi32.lib -FeWin32RegKey.
dll
javac Win32RegKeyTest.java
java Win32RegKeyTest
```

如运行成功在 Windows 的 DOS 命令提示符下运行结果如图 3 所示。



```
H:\Win32RegKeyTest>java Win32RegKeyTest
Postgraduate=zhu zhaoxia
Specialty=Technology for Computer Application
Lucky number=12
odd number=5 9 13 17 21
```

图 3 DOS 下显示的注册表中的部分键值
此时通过 Windows 注册表编辑器可观察到图 4。

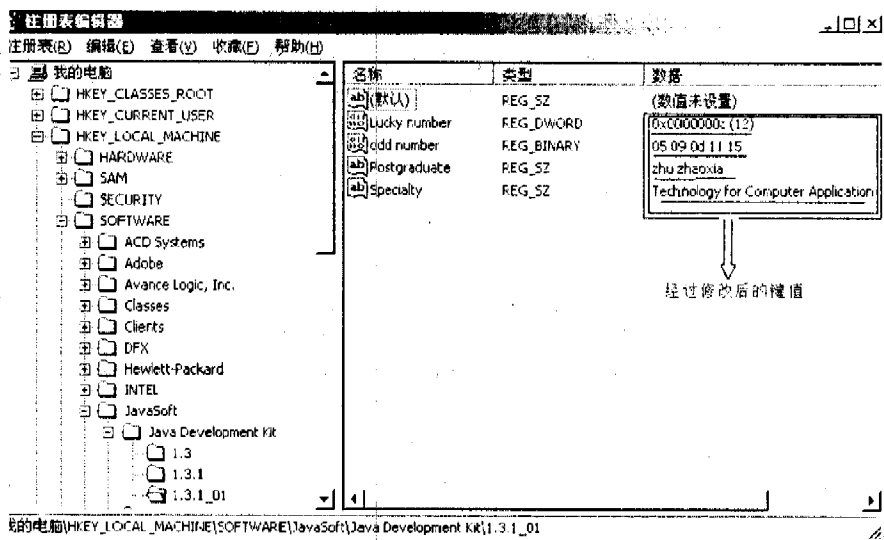


图 4 修改后的注册表内容

3 结束语

通过 SUN 公司提供的 Java 本地方法不仅解决了 Java 不能访问 Windows 注册表的难题,同时通过此实例展示了如何用 Java 平台包装程序来包装普通 C 语言的 API 子集。通过 Java 本地方法,可方便地查询及修改注册表的任何位置的有关信息。由于注册表是 Windows 的核心数据库,一旦操作失败会对系统和应用程序造成不可预见的影响,所以在操作之前一定要注意对注册表的保护工作,以免一次误操作导致系统的崩溃。目前在从事网络平台设计、分布式 DDOS 攻击的实时 IDS 研究和基于移动代理的 P2P 计算等项目过程中都需要利用 JNI 本地方法来进一步完成相关任务。

文中程序运行平台为 Windows NT 及在 JDK1.3.1-0 环境下。

参考文献:

- [1] 刘晓华.精通 Java 核心技术[M]:北京:电子工业出版社,2003.
- [2] Horstmann C S.最新 Java 2 核心技术[M]:北京:机械工业出版社,2003.
- [3] 管贻生. Java 高级实用编程[M]:北京:清华大学出版社,2004.
- [4] SUN Corp. Java Application WITH Java Native Interface[EB/OL]. 1994. <http://java.sun.com/docs/books/tutorial/native.1/2002/2004>.
- [5] LIRON T. Enhance Your Java Application with Java Native Interface[EB/OL]. 1998. <http://www.public.asu.edu/~wjanjua/java/jni/2003/2004>.

(上接第 224 页)

7(1):118-127.

[10] Deng Jing, Varshney P K, Haas Z J. A New Backoff Algorithm for the IEEE 802.11 Distributed Coordination Function

[EB/OL]. 2003-10. http://www.ces.syr.edu/research/SensorFusionLab/Downloads/Jing%20Deng/Amild_cnds04.pdf.