一种WLAN中MAC层自私行为的防御方法

叶 进[1] 李陶深[1] 王正飞[2] 张向利[2]
(广西大学计算机与电子信息学院 南宁 530004)[1] (桂林电子科技大学信息与通信学院 桂林 541004)[2]

摘 要 WLAN中修改无线网卡参数的自私行为,严重影响了其他用户公平使用无线网络的权利,甚至会造成拒绝服务攻击而使整个WLAN瘫痪。为了寻找真实环境中简单可行的解决方案,提出了一个应用于无线AP的防御算法,该算法按照一定的规则对自私节点发来的包进行丢弃,以达到限制节点发包速度的目的,通过保证用户公平地使用无线网络达到防御上述自私行为的目的。在无线网卡驱动中应用了该算法,并且搭建了真实的无线局域网进行测试。结果表明,提出的算法无论是对自私行为还是拒绝服务攻击都能起到明显的缓解作用。

关键词 WLAN,MAC,自私行为,拒绝服务攻击
中图法分类号 TP393 **文献标识码** A

Defense Mechanism against Misbehavior of MAC Layer in WLAN

YE Jin[1] LI Tao-shen[1] WANG Zheng-fei[2] ZHANG Xiang-li[2]
(School of Computer, Electronic and Information, Guangxi University, Nanning 530004, China)[1]
(School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China)[2]

Abstract The misbehavior that deliberately modifies the parameters of wireless NIC (Network Interface Card) driver impairs the legitimate users' rights, even may cause DoS attack and cripple the entire WLAN. To defense against such misbehavior, this paper proposed a defense mechanism that is applied at AP which controls the nodes' sending speed by dropping the incoming packets in a certain rule and makes WLAN users to use wireless channel fairly. We constructed a WLAN to test our proposed algorithm by applying it in wireless NIC driver. The experiments show that our algorithm performs well in defensing against MAC layer misbehavior even against DoS attack.

Keywords WLAN, MAC, Misbehavior, DoS attack

1 引言

随着无线网络的快速发展,无线网络用户数量激增。为了让WLAN中的所有节点都能公平、合理地使用信道,IEEE802.11 MAC协议使用了分布式协调功能(Distributed Coordination Function,DCF)来解决多节点接入问题。传统的IEEE 802.11MAC协议是一个开放的并且假设各接入节点都合法的非约束型多点接入控制协议。也就是说,只要节点采用该标准的接入方式,无论节点是合法的还是恶意的都允许接入,即它对接入节点的行为不进行检测,只负责按照二进制退避算法控制各节点的接入秩序。这样一来,就使得某些节点为了获得更大的吞吐量而采取诸如减小退避窗口(Contention Window,CW)值、减小分布式帧间隔(DCF Inter-Frame Space,DIFS)或者增大网络分配矢量(Network Allocation Vector,NAV)的行为,这些行为统称为自私行为,具有该行为的节点称为自私节点。显然,自私行为违反了IEEE802.11公平性原则,使其他合法节点的吞吐量降低,甚至有可能造成拒绝服务攻击[1],严重损害了合法节点的利益。

文献[2]中作者对MAC层的自私行为根据其退避窗口的变化方式不同分为了4类,并对这4类自私行为进行了仿真,分别在低负载、中负载和高负载的情况下将自私节点与合法节点的吞吐量进行对比。仿真结果表明:自私节点在网络负载较高的情况下,退避窗口值越小对其他节点越不公平。

对于无线网络中MAC层的自私行为,目前的应对方案大多是先通过相应的检测算法识别自私节点[3,4],再启动相应的防御机制[5,6]。文献[3]提出的DOMINO是一个基于有效观察判断自私行为的方法,通过将最大退避值、实际退避值、连续退避值等指标与某一限值做比较,来判定节点是否更改网络协议参数。然而自私节点有可能采取更加灵巧的方式来规避检测,例如,采用随机化修改CW值的方式使得退避时间(backoff time)仍然很小。文献[4]提出了一种基本的检测模型以适应这种灵巧的规避行为和复杂环境的干扰。文献[5]在AP处以一定的概率拒绝回复源节点MAC层的ACK,这个概率由作者提出的在线反馈机制来决定,从而可以惩罚那些自私节点。这些方案都是在AP处作出修改,不需要额外的通信或者其他节点的合作。文献[6]提出了一种解决节

到稿日期:2013-05-21 返修日期:2013-07-17 本文受国家自然基金项目(61163060),国家科技支撑计划项目(2012BAH18F03),广西科技开发项目(桂科攻12118017-2C),桂林市科技开发资助项目(20120104-13),广西区研究生创新基金(YCSZ2012069)资助。
叶 进(1970—),女,博士,教授,主要研究方向为网络协议优化,E-mail:yejin@gxu.edu.cn;李陶深(1957—),男,教授,博士生导师,主要研究方向为优化理论;王正飞(1989—),男,硕士生,主要研究方向为网络安全;张向利(1968—),女,博士,教授,主要研究方向为无线传感器网络。

点选择较小退避窗口问题的方法。该方法由接收者界定退避时间并将它通过 CTS 帧或 ACK 帧发送给发送者。

目前，应用到真实网络环境中的防御方法很少，上述文献提出的检测方法以及应对措施的效果和性能还有待真实网络环境的检验。本文提出了一种基于 802.11 协议的自私行为防御算法，并将其应用到了真实网络环境中。实际测试的结果表明，本文提出的算法能有效地遏制自私行为，对于拒绝服务攻击也表现出良好的抗性。

2 IEEE 802.11 中自私行为的分析

自私节点修改网卡参数的目的是为了在与其他正常节点竞争共享信道的时候获得更大的接入概率，降低了其他合法节点接入信道的概率，这就使得自私节点的发包速度高于合法节点的发包速度。WLAN 中的自私行为主要有 3 种：减小 CW 值、减小 DIFS 和增大 NAV 值。

IEEE 802.11 标准 DCF 接入机制规定：一个节点要发送数据时，首先监听信道，当信道空闲时间超过 DIFS 时才能启动发送程序。自私节点减小 DIFS 之后就会比其他合法节点等待更少的时间。

为了避免所有节点同时发送数据而造成冲突，IEEE 802.11 标准需要错开每个节点的发送时间，当节点监测到信道空闲时间超过 DIFS 时，节点并不是立即发送数据，而是再等待一段退避时间 T。T 值由该节点当前的 CW 值决定，即

$$T=Random(0,CW)*TimeSlot \tag{1}$$

由上式可知，自私节点通过减小 CW 值缩短了等待时间，所以，它在与合法节点竞争信道时更占优势。

DCF 接入机制规定：发送长数据帧时启动 RTS/CTS 机制，即发送端首先发送 RTS(Request to Send)帧预约信道。RTS 帧中有一个持续时间域(Duration Field)，该域标明了传输整个数据帧和其 ACK 所需的时间，收到这个 RTS 帧的每个节点都会根据这个持续时间来设置自身的 NAV 值，当 NAV 值减小到 0 时，节点才可以发送数据，在此之前都保持抑制状态。自私节点通过增大 RTS 帧中的持续时间，间接地增大了其他节点的 NAV 值，造成其他节点需要等待更长的时间才能参与信道竞争，从而达到长期占用信道的目的。

WLAN 中 MAC 层的自私行为的特点为：

(1)隐匿性更强，而且还独立于高层(MAC 层以上)，难以被高层的检测机制检测到，因此可以与高层配合以获得更好的效果。

(2)危害性更大。这种发生在 MAC 层的自私行为对所有类型的数据包都起作用，因为所有的数据包都必须通过 MAC 层，这将使得由它造成的拒绝服务攻击更难防御。

自私节点修改网卡的相应参数后，在与其他合法节点竞争信道时便可获得更大的接入概率，而其他合法节点接入信道的概率却降低了，这就造成了自私节点的发包速度(单位时间内成功发送的数据包的数量)比合法节点的发包速度高。以减小退避窗口为例，即假设 WLAN 中有 n 个节点和 1 个 AP，其中有 l 个节点是自私节点。每个自私节点都带有一个自私因子，记为 $g_1\cdots g_l(0<g_1\cdots g_l<1)$，表示其 CW 值减小的程度(或者说自私的程度)。其余 $n-l$ 个节点都是合法节点。设一个自私因子为 $g_a(1<a<l)$ 的自私节点 S_a 选择的退避值为 $Random(0,g_aW-1)$，其中 W 表示当前的退避窗口值，而合法节点选择的退避值为 $Random(0,W-1)$。一个节点在不同的时刻会有不同的退避窗口值和不同的退避时间。若 $CW=2^iCW_{\min}(0\leqslant i\leqslant m)$，则表明节点当前处于第 i 阶段，此时的退避窗口值为 W_i。令 p_0 表示合法节点的发包冲突概率(发送数据包产生冲突的概率)，p_a 表示自私节点 S_a 的发包冲突概率。根据上述环境，文献[7]建立一个马尔可夫模型：在网络饱和的情况下，合法节点在任意一个时隙内接入 AP 的概率 τ_0 和自私节点接入 AP 的概率 τ_a，并得出接入概率、冲突概率和退避窗口之间的关系如下：

$$\begin{cases}\tau_0=\dfrac{2(1-2p_0)}{(1-2p_0)(W+1)+p_0W(1-(2p_0)^m)}\\ \tau_1=\dfrac{2(1-2p_1)}{(1-2p_1)(g_1W+1)+p_1g_1W(1-(2p_1)^m)}\\ \cdots\\ \tau_l=\dfrac{2(1-2p_l)}{(1-2p_l)(g_lW+1)+p_lg_lW(1-(2p_l)^m)}\\ p_0=1-(1-\tau_0)^{n-l-1}\prod\limits_{1\leqslant i\leqslant l}(1-\tau_i)\\ p_1=1-(1-\tau_0)^{n-l}\prod\limits_{2\leqslant i\leqslant l}(1-\tau_i)\\ \cdots\\ p_l=1-(1-\tau_0)^{n-l}\prod\limits_{1\leqslant i\leqslant l-1}(1-\tau_i)\end{cases} \tag{1}$$

已知总节点数为 n，自私节点数为 l，只要给出自私因子 g_i 的值即可求出节点的接入概率 τ_i 和冲突概率 p_i。从数学模型(1)可以得出，自私节点的自私因子直接影响节点的接入概率，即 g_i 越小(退避窗口)的节点，其接入概率 τ_i 越大。

对于接入概率大的节点，在单位时间内，AP 将收到比其他节点更多的数据包。因此，本文自定义了节点 N_i 的源地址重复度，记为 d_i，它表示 AP 连续收到来自节点 N_i 的数据包的次数，即 AP 当前收到的数据包的源地址与上一次收到的数据包的源地址相同的次数。当 $d_i=0$ 时，表示当前 AP 收到的数据包与上一次收到的数据包的源地址不同。当 $d_i=n$ 时，表示当前 AP 连续收到 n 个来自 N_i 的数据包，其概率分别为：

$$\begin{cases}P(d_i=0)=1-\tau_i\\ P(d_i=n)=\tau_i^n\end{cases} \tag{2}$$

定义 D 为单位时间内收到的每个数据包对应的源地址重复度的和。AP 每收到一个数据包都要更新其源节点 N_i 对应的 d_i。假设单位时间内收到来自节点 N_i 的数据包的数量为 n，令 $d_{i,j}$ 为 AP 收到来自节点 N_i 的第 j 个数据包对应的源地址重复度($1\leqslant j\leqslant n$)。则 N_i 在单位时间内源地址重复度总数 D_i 为：

$$D_i=\sum_{j=1}^{n}d_{i,j} \tag{3}$$

如果 AP 收到来自 N_i 的第 j 个数据包和上一个数据包也来自节点 N_i，则第 j 个数据包被认为是连续收到的，此时，$d_{i,j}=1$；否则第 j 个包被认为是非连续收到的，$d_{i,j}=0$。

τ_i 为 N_i 的接入概率，AP 每次收到 N_i 的包是连续的概率为 τ_i^2。意味着在来自 N_i 的 n 个包中有 $\tau_i^2(n-1)$ 个数据包(第一个数据包的源地址重复度被认为是 0)的 $d_{i,j}=1(0\leqslant j\leqslant\tau_i^2(n-1))$。而在这 $\tau_i^2(n-1)$ 个包中会存在二次连续的包，其存在二次连续包的概率为 τ_i^2。意味着在来自 N_i 的 n 个数据包中有 $\tau_i^2(\tau_i^2(n-1)-1)$ 个数据包的 $d_{i,j}=2(0\leqslant j\leqslant\tau_i^2(\tau_i^2(n-1)-1))$。依此类推，在来自 N_i 的 n 个数据包中有 $\tau_i^{2a}(n-1)-\tau_i^{2(a-1)}-\tau_i^{2(a-2)}-\cdots-\tau_i^2$ 个数据包的 $d_{i,j}=a$ $(a=1,2$

…)，若 $\tau_i^{2a}(n-1)-\tau_i^{2(a-1)}-\tau_i^{2(a-2)}-\cdots-\tau_i^2<1$，则不再计算下去。因此，

$$D_i=(\tau_i^2 n-\tau_i^2)+(\tau_i^4 n-\tau_i^4-\tau_i^2)+\cdots+(\tau_i^{2a}(n-1)-\tau_i^{2(a-1)}-\tau_i^{2(a-2)}-\cdots-\tau_i^2) \quad (4)$$

而单位时间内 N_i 发送数据包的数量为：

$$n=\frac{1}{TimeSlot*T}\tau_i \quad (5)$$

式中，T 为传输每个数据包所需的时间。

通过文献[7]建立的马尔可夫模型可以推出减小退避窗口与节点接入概率的函数关系，并且得知退避窗口越小的节点接入概率越大。从式(4)中可知节点单位时间内源地址重复度的总和 D_i 与节点的接入概率 τ_i 的函数关系，τ_i 越小，D_i 越接近 0。当自私节点发动拒绝服务攻击独占信道的时候，其接入概率 τ_i 为 1，则该自私节点单位时间内源地址重复度的总和为：

$$D_i=(n-1)+(n-2)+\cdots+1=n(n-1)/2 \quad (6)$$

式中，n 为自私节点单位时间内发包的数量。式(6)说明，当节点发动拒绝服务攻击独占信道的时候，其单位时间内源地址重复度总和随着发包速度的增加呈几何级增长。

本文通过搭建 WLAN，在实际网络环境中测试了自私行为对节点源地址重复度的影响。本实验采用 WLAN 的基础工作模式，即 infrastructure 模式，使用 DCF 接入控制方式。其拓扑结构如图 1 所示。AP 和节点 A 都安装 Fedora 13.0 操作系统，使用 Madwifi[8] 作为驱动的 TL-WN550G 网卡。节点 B 安装 Windows XP 操作系统，使用 WL-167g USB 无线网卡及其配套驱动。

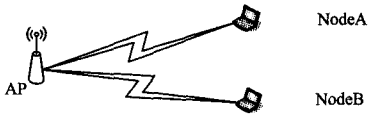

图 1　搭建 WLAN 的拓扑结构

实验中，我们在 AP 上配置了一个基于 Socket 的多线程 TCP 服务器，在节点 A 和 B 上配置客户端。A 和 B 都向 AP 发送 TCP 数据包，数据包的大小都设置为 1400 字节。节点 A 是自私节点，节点 A 在不同自私行为和强度的情况下，对节点 A 和 B 的源地址重复度进行统计。

定义 η 为节点 A 与 B 的平均源地址重复度的比值的自然对数，即

$$\eta=\ln\frac{\text{节点 A 的平均源地址重复度}}{\text{节点 B 的平均源地址重复度}} \quad (7)$$

图 2 中表示节点 A 分别在减小 DIFS 值、减小 CW 值和增大 NAV 值的情况下，η 的变化情况。其中，$g_0\cdots g_5$ 代表节点 A 在 3 种自私情况下的不同强度，$g_0\cdots g_5$ 分别对应节点 A 减小 DIFS 和 CW 至 1、0.8、0.6、0.4、0.2、0.1 倍或者 A 增大 NAV 至 1、10、50、100、200、500 倍。自私行为强度为 g_0 时，表现为正常行为，强度为 g_5 时，表现为极端的自私行为。

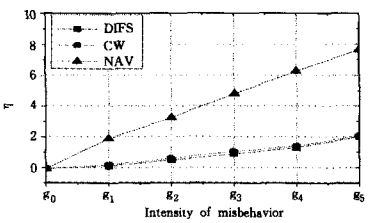

图 2　不同强度的 3 种自私行为下 η 的变化

从图 2 中可以看出：1)节点的自私行为影响着其源地址重复度的变化，即越自私其源地址重复度越大；2)增大 NAV 的自私，行为对节点的源地址重复度增加尤其明显。因此，本文把节点的源地址重复度作为判断节点是否是自私节点的依据。

3　基于 IEEE 802.11 MAC 协议的防御算法

为了限制 WLAN 中节点的自私行为，必须对自私节点做出惩罚。为此，我们在 AP 处建立一个虚拟列表，用于缓存各节点的地址(即源地址)，并对每个节点都设置一个丢包门限 Threshold。当收到来自该节点的数据包数量达到丢包门限时丢弃当前收到的数据包。为了能将本算法部署在 AP 上的 MAC 层中，我们在无线网卡驱动代码中为每一个节点建立了一个缓存列表来记录各节点的信息：

```
struct packet_list{
int staMAC[6];//记录节点源地址
int Threshold;//丢包门限
int count;//收包计数器
int d;//源地址重复度
};
```

其中，Threshold 表示丢包的门限值，其计算公式为：

$$Threshold=Threshold-d*p \quad (8)$$

d 表示节点的源地址重复度，p 为惩罚系数(可手动设置)。count 用于记录 AP 收到来自某个节点的数据包的数量，当从一个源地址收到包的数量大于门限值 Threshold 的时候(即 count>Threshold 时)，当前收到的这个数据包将被丢弃。如前所述，节点的自私行为越严重，其源地址重复度 d 越大，该节点的丢包门限 Threshold 也就减小得越快，因为惩罚性丢包次数增多，限制了该节点连续接入 AP 的能力。图 3 说明了一个完整的算法过程。

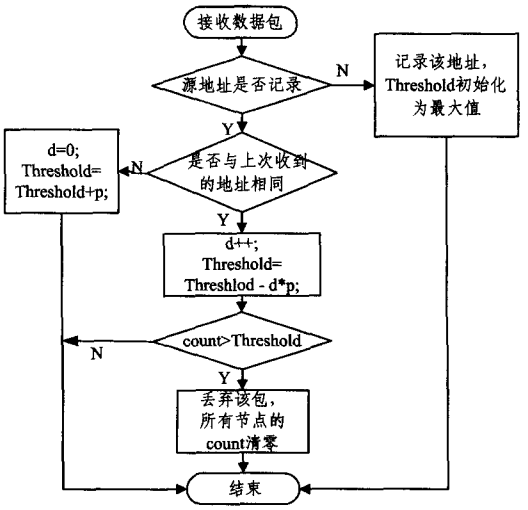

图 3　自私行为防御算法流程图

关于本算法有以下几个关键点：

(1)源地址重复度 d 的计算。当 AP 当前收到的数据包的源地址与上一次收到的相同时，该源地址对应节点的 d 值增加 1，否则，d 清零。当 WLAN 中每个节点的接入概率都相差不大时，每个节点的 d 值大部分时候应该都为 0。当 WLAN 受到拒绝服务攻击时，来自攻击节点的数据包会充斥整个信道，AP 会收到大量来自自私节点的接入请求，因此自私节点的 d 值必然较大，而合法节点的 d 值大部分时候为 0。

(2)丢包门限 Threshold 的计算。当 AP 收到数据包的源

地址与上一次收到的相同时，则根据式(8)减小该源地址对应节点的丢包门限。惩罚系数 p 用来控制发包速度的收敛速度，p 太小则发包速度收敛慢，p 太大则发包速度波动大。当AP收到数据包的源地址与上一次收到的不相同时，当前源地址对应节点的丢包门限值增加 p，即

$$Threshold = Threshold + p \tag{9}$$

实际上，本算法是通过动态变化的丢包门限值来调整虚拟列表中各节点的丢包概率，间接地控制了各节点的接入概率，而且对丢包门限设置了上下限，保证了每个节点正常使用信道。

(3)节点的收包数量 count 达到丢包门限时，丢弃当前收到的数据包，并且清空所有节点的 count。这样做的目的是：只对当前发包速度最大的节点做出相应的惩罚性丢包。

4 实验与分析

在本节的实验中，我们将在节点 A 的 Madwifi 驱动中分别修改其 DIFS 值、CW 值和 NAV 值，使其成为自私节点，并在 AP 的网卡驱动中添加防御算法。通过对比 AP 启动防御前后 A、B 的发包速度和吞吐量来验证本文提出的防御算法的有效性。图 4—图 6 给出了针对 3 种不同自私行为下 A 和 B 发包速度的对比。

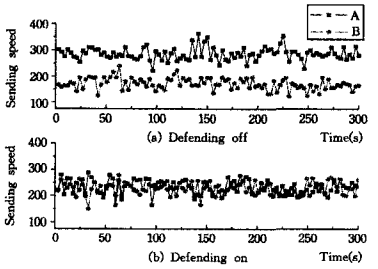

图 4 节点 A 减小 DIFS 到 1 时 A 和 B 的发包速度

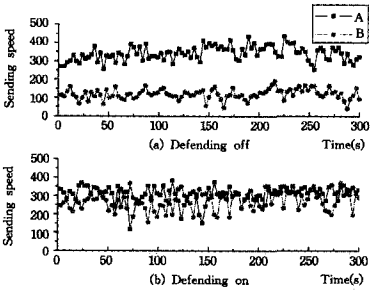

图 5 节点 A 减小 CW 到 1 时 A 和 B 的发包速度

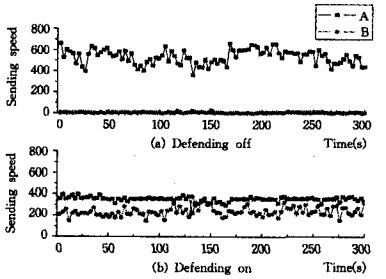

图 6 节点 A 增大 500 倍 NAV 时 A 和 B 的发包速度

从我们对 3 种自私行为的实测结果可以看出：自私节点会严重影响其他合法节点的发包速度，甚至有可能瘫痪整个网络。在 AP 上部署防御算法后，情况得到了明显的改善，提高了节点使用信道的公平性，即使在 WLAN 遭受拒绝服务攻击时也能保证合法节点正常地使用信道。

表 1 记录了不同强度的自私行为下自私节点 A 与合法节点 B 在启动防御算法前后吞吐量的数据。吞吐量＝发包速度 * 数据包大小(每个包 1400Bytes)，单位：kbps。

从表 1 可知：启动防御算法之前，随着节点自私行为的加剧合法节点的吞吐量明显下降；启动防御算法后，尽管存在自私行为，但是合法节点与自私节点的吞吐量基本持平，说明本防御算法能够有效地限制 WLAN 中 MAC 层的自私行为，提高了公平性。

表 1 不同强度的 3 种自私行为下节点 A 和 B 的吞吐量变化

不同强度的3种自私行为		防御前		防御后	
		A	B	A	B
减小 DIFS	0.8DIFS	336.2	324.4	336.0	325.0
	0.4DIFS	354.0	311.5	328.1	330.4
	0.1DIFS	399.5	239.9	335.8	321.5
减小 CW	0.8CW	341.2	330.5	330.8	334.0
	0.4CW	361.6	318.3	342.0	320.9
	0.1CW	476.0	168.1	355.4	343.2
增大 NAV	10NAV	430.5	301.0	383.0	366.4
	100NAV	558.2	196.3	405.0	355.0
	500NAV	840.6	8.4	483.2	330.4

结束语 本文提出的自私行为防御算法只需部署在 AP 上(仅修改 AP 的无线网卡驱动即可)，不需要额外的开销或者其他节点的配合就能限制攻击节点的自私行为，从而保证了合法节点接入信道的概率，维护了 WLAN 的公平性。另外，从实测的数据来看，本防御算法对拒绝服务攻击也能起到良好的防御效果。

参 考 文 献

[1] Gupta V, Krishnamurthy S, Faloutsos M. Denial of service attacks at MAC layer in wireless ad hoc networks[C]// Proceedings of 2002 MILCOM Conference. Anaheim, CA, 2002: 1118-1123

[2] Giri V R, Jaggi N. MAC layer misbehavior effectiveness and collective aggressive reaction approach[C]// Proceedings of the 33th IEEE Sarnoff Symposium. Princeton, NJ, 2010: 1-5

[3] Raya M, Aad I, Hubaux J P. DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 Hotspots[J]. IEEE Transactions on Mobile Computing, 2006, 5(12): 1-11

[4] Radosavac S, Baras J S, Koutsopoulos I. A framework for MAC protocol misbehavior detection in wireless networks[C]// Proceedings of the 4th ACM Workshop on Wireless Security. Cologne, Germany, 2005: 33-42

[5] Kyasanur P, Vaidya N H. Selfish MAC layer Misbehavior in wireless networks[J]. IEEE Transactions on Mobile Computing, 2005, 4(5): 502-516

[6] Konorski J. Multiple access in ad hoc wireless LANs with noncooperative stations[C]// Proceedings of NETWORKING. London, UK: Springer-Verlag, 2002: 1141-1146

[7] Lee S K, Rong Yan-xia, Choi H A. Detecting Stations Cheating on Backoff Rules in 802.11 Networks Using Sequential Analysis [C]// Proceedings of 25th IEEE International Conference on Computer Communications. Barcelona, Spain, 2006: 1-13

[8] Madwifi: a linux kernel device driver for wireless LAN chipsets [EB/OL]. http://madwifi.org/, 2011