

P2P 网络的节点状态研究

鲁 薇,周竹荣,邱玉辉

(西南大学 计算机与信息科学学院,重庆 400715)

(aloewei@swu.edu.cn)

摘 要:对等网络中,研究节点的行为差异有利于实现区分服务,激励节点共享资源,提高系统的性能。用节点状态表示节点行为表现,提出 P2P 节点状态模型,描述状态的动态变化规律,并将模型与节点行为评价机制结合,确定节点状态的约束条件,提高模型的可用性。实验证明模型有效地支持区分服务,能促进节点的资源共享,具有良好的扩展性和实用性。

关键词:对等网;节点状态;区分服务;激励机制

中图分类号: TP393 **文献标志码:** A

Research on node states in P2P network

LU Wei, ZHOU Zhu-rong, QIU Yu-hui

(College of Computer and Information Science, Southwest University, Chongqing 400715, China)

Abstract: Analyzing nodes' behaviors in a Peer-to-Peer (P2P) network can effectively provide different services and encourage nodes to share resources, thereby improving the system performance. Using states to represent nodes' behaviors, P2P node state model was proposed to describe the process of states transition. The model was combined with nodes' behaviors assessment mechanism to specify conditions of node state. Experimental results show that the model, which has good scalability and availability, can achieve service differentiation and promote resources sharing by nodes.

Key words: Peer to Peer (P2P); node state; differentiated service; incentive mechanism

0 引言

在对等网络中,节点的行为影响着整个网络的健壮性和稳定性。根据节点的行为表现定义节点状态,建立状态模型有利于实现区分服务,激励节点共享资源,提高系统的性能。目前基于节点行为的状态研究还没有引起足够重视。在文献[1]中,Murat 用状态表示节点的搭便车类型,形式化描述了抑制搭便车机制,但未对非搭便车节点进行讨论。文献[2]在提出一种支持区分服务的激励机制时,提到了所有节点在服务意愿的影响下会形成三种状态,但文献着眼于整个激励体系的建立,对节点状态的分析较为粗略,状态划分粒度较大,降低了对区分服务的支持度。

为解决以上问题,本文提出了节点状态模型,采用有限自动机(Definite Automata, DFA)的形式化方法,通过细化节点状态粒度,定义状态类型、迁移状态原语及转换规则描述了 P2P 整体节点的状态转换过程,并将模型与具体评价机制结合分析节点状态变化的约束条件,提高模型的可用性。通过对细粒度状态类型施行惩罚与奖励规则实现对区分服务的有力支持,实验证明了模型可实现性、扩展性与实用性。

1 相关理论

1.1 信任机制

P2P 网络提供真实世界人类交流的网络环境,是以人为中心的网络,与社会网络具有同构性。信任作为社会存在的一个整体部分,是社会网络中人与人之间的核心关系。文献

[3]中定义信任是个体在其行为被监视并影响接下来的行为之前,与另一个体发生特定行为的主观概率的特定水平。信任作为评价 P2P 网络中节点行为规范的参数,已存在多种算法及模型,如:EigenTrust^[4]、PowerTrust 等。

1.2 激励机制

P2P 网络中存在着极普遍的搭便车行为(节点只享用不贡献),大量搭便车者依附于极少数的共享节点^[5],这些现象会引发资源匮乏、网络阻塞等问题,严重影响到 P2P 网络的整体效用。因此,衡量节点的贡献度,分析搭便车行为^[1,5,6],设计有效的激励机制^[2,7,8],促进节点共享资源具有非常重要的意义。

2 P2P 节点状态模型

本文以 P2P 文件共享系统为基础,分析节点交易行为的差异,对相关概念定义如下。

定义 1 节点状态。节点状态表示节点在网络中的行为表现处于某个级别的模式。

定义 2 迁移原语。迁移原语是用来控制管理节点状态转换的一段不允许并发执行的程序。

结合上述定义,用 DFA 的形式化方法对 P2P 网络节点建立节点状态模型,简称 NS(Node State)模型, $G = (V_s, P, \delta, s_0, F)$ 。其中:

1) 节点状态集合 $V_s = \{ \text{Zombie, Block, Discreditable, Suspend, New, Normal, Creditable, Selfless} \}$;

2) 迁移原语 $P = \{ \text{create, activate, lock, sleep, wakeup,} \}$

收稿日期:2009-02-27;修回日期:2009-04-10。 基金项目:国家 973 计划项目(2003CB317008)。

作者简介:鲁薇(1986-),女,湖北黄冈人,硕士研究生,主要研究方向:P2P 技术、网络计算、多媒体技术及应用;周竹荣(1970-),男,四川大竹人,副教授,博士,主要研究方向:P2P 技术、语义网络、多媒体技术;邱玉辉(1938-),男,重庆江津人,教授,博士生导师,主要研究方向:语义网络、人工智能。

wait, upgrade, degrade};

3) 映射 $\delta: V_s \times P \rightarrow V_s$: 是节点从当前状态经过迁移原语转换到另一状态的转换函数;

4) 初始状态 $s_0 = \text{New} \in V_s$;

5) 终态集 $F = \{\text{Zombie}\} \subset V_s$ 。

2.1 节点状态类型

NS 模型根据交易行为中节点获得的诚信评价和对系统的贡献程度确定节点等级,划分状态类型,以便进行区分服务。NS 模型的状态集合 V_s 包含从低到高八种级别的状态类型。

1) Zombie: 终结状态。节点表现恶劣,失去可信度。

2) Block: 封锁状态。节点表现糟糕,尤其在可信度上。

3) Suspend: 挂起状态。节点贡献相对过低,搭便车性质严重。

4) Discreditable: 可疑状态。节点在贡献度及可信度上表现不佳。

5) New: 初始状态。初次加入网络的新节点,没有历史行为的记录。

6) Normal: 普通状态。节点在系统贡献度及行为可信度上表现一般。

7) Creditable: 可信状态。节点在贡献度及可信度上表现良好。

8) Selfless: 无私状态。节点表现非常出色,对系统提供了大量可信贡献。

2.2 状态迁移原语

规定一系列迁移原语来实现 NS 模型中节点从当前状态到下一状态的转换。迁移原语集合 P 中原语的名称及相应行为如下。

1) create: 新成员加入网络,进入初始状态。通知邻接节点有新增成员,初始化节点信息。

2) activate: 新节点参与网络交易,刷新空白历史行为记录,激活进入普通状态。

3) lock: 永久性封锁节点,进入终结状态,避免不可信节点的恶性影响。

4) sleep: 隔离节点进入封锁状态休眠,以便严厉惩罚节点的欺骗性行为。

5) wait: 隔离状态进入挂起状态,仅能提供服务,以便抑制节点的搭便车行为。

6) wakeup: 惩罚时间耗尽或条件解除,从当前隔离状态唤醒节点。

7) upgrade: 节点表现改善或提升,奖励节点进入高级状态。

8) degrade: 节点行为恶化或表现不佳,惩罚节点降低一级状态。

2.3 状态转换规则

由于受到主、客观因素的影响,节点在网络中行为无确定性,导致节点状态转换的方向存在多种可能,映射函数 δ 包含状态间所有的转换规则。

1) 初始。

$\delta(\emptyset, \text{create}) = \text{New}$: 新成员加入,节点状态类型初始化为初始状态。

$\delta(\text{New}, \text{activate}) = \text{Normal}$: 新节点经过交易获得历史行为记录进入普通状态。

2) 奖励政策。

$\delta(\text{Discreditable}, \text{upgrade}) = \text{Normal}$;

$\delta(\text{Normal}, \text{upgrade}) = \text{Creditable}$;

$\delta(\text{Creditable}, \text{upgrade}) = \text{Selfless}$;

由于多次提供可信服务使贡献度增加或诚信度提高,满足高一级状态的约束条件,节点从较低级升至高级状态。

3) 惩罚政策。

$\delta(\text{Selfless}, \text{degrade}) = \text{Creditable}$;

$\delta(\text{Creditable}, \text{degrade}) = \text{Normal}$;

$\delta(\text{Normal}, \text{degrade}) = \text{Discreditable}$;

节点的相对贡献度或诚信度降低到临界值,节点降低一级状态。

4) 隔离处罚。

$\delta(\text{Discreditable}, \text{wait}) = \text{Suspend}$;

节点在可疑状态相对贡献度过低,成为需要抑制的搭便车者,隔离节点至挂起状态。

$\delta(\text{Discreditable}, \text{sleep}) = \text{Block}$;

可疑状态节点诚信度低至边界值,隔离至封锁状态休眠。

$\delta(\text{Suspend}, \text{lock}) = \text{Zombie}$;

节点在被隔离的挂起状态仍提供虚假服务,信任度降至最低限值,为防止对系统产生深度恶性影响,调入终止状态。

5) 解除当前隔离。

$\delta(\text{Block}, \text{wakeup}) = \text{Suspend}$: 休眠时间结束,停止隔离,唤醒节点进入挂起状态。

$\delta(\text{Suspend}, \text{wakeup}) = \text{Discreditable}$: 经过挂起状态的提供服务,节点贡献度及诚信度达到高级可疑状态的要求,唤醒节点进入可疑状态。

状态转换规则奖励与惩罚政策相辅,体现区分服务思想,用状态转换图描绘状态非确定的动态变化过程,如图 1 所示。

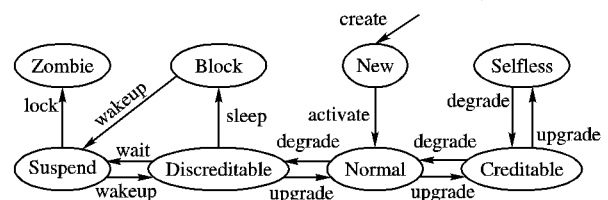


图 1 状态转换图

3 NS 模型与评价机制的结合

NS 模型的实际应用需要节点行为评价机制的数据支持,两者关系如图 2 所示。只有通过评价机制计算节点的贡献度及诚信度,然后根据结果确定 NS 模型状态的约束条件,才能形成具有实际应用价值的完整 NS 模型。因此,NS 模型与评价机制的结合过程必不可少。在结合过程中需要解决以下问题:

1) NS 模型的数据结构,主要是模型状态的数据结构;

2) 现存评价机制的选择(搭便车衡量方法、信任机制);

3) 如何利用评价结果规定各级状态的约束条件,即两者结合方法。

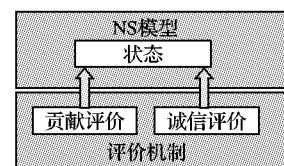


图 2 NS 模型与评价机制的关系

3.1 NS 模型的数据结构

在分布式 P2P 网络中,NS 模型整体思想是节点通过各自的 P2P 终端程序实时监控节点行为评价参数的变化,在参数达到临界值时调用迁移原语来控制节点状态的转换。

用一维数组 S 存储 NS 模型状态集合 V_s 中的 8 种状态类型,变量 $status$ 存储代表相应状态类型的整型数值 $0 \sim 7$,则满足关系 $\exists i \in \{0,1,2,3,4,5,6,7\}, \forall Si \in V_s \Rightarrow status = i$ 。迁移原语通过修改 $status$ 的值控制节点的状态变化。如 create 原语初始化节点 $status = 4$,代表状态 New;activate 原语修改节点 $status$ 为 5,控制节点状态类型由 New 转入 Normal。定义变量 $sleep_t$ 代表 Block 状态的休眠时间参数,每次进入 Block 状态时 $sleep_t$ 初值为秒级常数。当休眠时间耗尽, $sleep_t$ 为 0,休眠结束,wakeup 原语将节点 $status = 1$ 更新为 2,从 Block 状态唤醒节点。

3.2 评价机制的选择

文献[2]提出了基于特征向量的类 HITS (Hyperlink-Induced Topic Search) 算法衡量节点的贡献度,算法具有一定的稳定性,公式为:

$$s(x) = \sum_{y=1}^n S(x, y) * u(y) \quad (1)$$

$$u(x) = \sum_{y=1}^n U(x, y) * s(y) \quad (2)$$

式(1),(2)中 S 是贡献得分矩阵; U 是 S 的转置,代表使用得分矩阵;向量 s, u 分别代表节点的贡献、使用值。

文献[8]提出的 EigenTrust 信任机制根据节点的交易历史,计算本地信任度,再通过节点间信任值迭代为每个节点计算全局节点可信度:

$$c_{ij} = \frac{\text{Max}(s(i, j) - \text{uns}(i, j), 0)}{\sum_j \text{Max}(s(i, j) - \text{uns}(i, j), 0)} \quad (3)$$

$$t_i^{(k+1)} = (1 - \alpha) \left(\sum_j t_j^k * c_{ji} \right) + \alpha * p_i \quad (4)$$

式(3)计算本地信誉值,其中 $s(i, j)$ 为节点 i, j 之间诚信交易的次数, $\text{uns}(i, j)$ 代表节点 i, j 之间不诚信交易的次数;全局信任度计算式(4)中 p_i 是节点的信任基数, $\alpha \in [0, 1]$ 为节点对假定信任基数的信任程度。

3.3 NS 模型与评价机制结合过程

NS 模型与评价机制的结合思路是根据节点贡献度和诚信度的范围评定节点行为的等级,进而规定节点状态类型的约束条件。约束条件的确定标志结合的完成。具体过程如下:

1) 贡献度和诚信度的等级划分。

分别对节点贡献度和诚信度进行取值范围的区间划分,赋予等级特征。

s, u 更新表示贡献、使用值按百分制排队的结果^[2]。借助判断搭便车的辅助参数 $A, B (B < A)$, 划分百分制为三个区间: $P_1 = [0, B), P_2 = [B, A), P_3 = [A, 100]$ 。定义 3×3 的矩阵 M , 矩阵元素 M_{ij} 表示 $s \in P_i \wedge u \in P_j$, 描述节点贡献及使用值排队的情况,以此衡量节点的相对贡献度。将 M_{ij} 按贡献度等级分类,定义 $Ci (i = 1, 2, 3, 4, 5)$ 表示贡献等级,如表 1。

表 1 贡献度等级

贡献等级	相对贡献度
C1	M_{13}
C2	M_{23}, M_{12}
C3	M_{11}, M_{22}, M_{33}
C4	M_{32}, M_{21}
C5	M_{31}

EigenTrust 机制诚信度取值范围为 $[0, 1]$, 划分区间如表 2 所示,其中诚信度对应等级用 $Ti (i = 0, 1, 2, 3, 4, 5)$ 表示。

表 2 诚信等级

诚信等级	诚信度
T0	$t \in [0, 0.1)$
T1	$t \in [0.1, 0.3)$
T2	$t \in [0.3, 0.5)$
T3	$t \in [0.5, 0.7)$
T4	$t \in [0.7, 0.9)$
T5	$t \in [0.9, 1]$

2) 状态约束条件的形成。

根据上节状态类型定义,可规定状态到达的条件如表 3,表中状态等级 l 的值域为 $L = \{0, 1, 2, 3, 4, 5, 6\}$ 。

表 3 状态的到达条件

状态	等级	状态到达条件	贡献等级	诚信等级
Zombie	0	T0	—	T0
Block	1	$T1 \wedge sleep_t \neq 0$	—	T1
Suspend	2	$C1 \vee (T1 \wedge sleep_t = 0)$	C1	—
Discreditable	3	$C2 \wedge T2$	C2	T2
Normal	4	$C3 \wedge T3$	C3	T3
Creditable	5	$C4 \wedge T4$	C4	T4
Selfless	6	$C5 \wedge T5$	C5	T5

但表 3 忽略了对一些中间状态节点的归类。这些节点的两个参数不在同一等级,如满足 $C4 \wedge T4$ 的节点在 Creditable 状态活动一段时间后参数等级降低为 $C3 \wedge T4$,但由于不能满足等级 Normal 状态的到达条件 $C3 \wedge T3$,节点无法降级,被滞留在 Creditable 状态,出现惩罚漏洞。因此需要补充条件的约束。

针对节点的贡献等级和诚信等级补充规定:两个参数等级若相差 1 个级别,则节点留在较高等级状态;若相差 2 个级别,则平衡节点处于中间等级;若相差 3 个级别,则将节点状态相对较高级降两个级别。补充规定不针对惩罚状态 (Suspend, Block, Zombie),用规则表示如下:

节点状态等级判断规则: $Ci \wedge Tj (2 \leq i, j \leq 5)$

$$l = \begin{cases} \max(i, j) + 2, & |i - j| = 0 \\ \max(i, j) + 1, & |i - j| = 1 \\ \max(i, j) - 1, & |i - j| = 2 \\ \max(i, j) - 2, & |i - j| = 3 \end{cases}$$

根据上述规则,表 3 中部分状态约束条件更新如表 4。

表 4 部分状态的到达条件补充

状态	到达条件	补充条件
Normal	$C3 \wedge T3$	$(C2 \wedge Ti) \vee (Ci \wedge T2); i = 3, 4, 5$
Creditable	$C4 \wedge T4$	$(C3 \wedge Ti) \vee (Ci \wedge T3); i = 4, 5$
Selfless	$C5 \wedge T5$	$(C4 \wedge Ti) \vee (Ci \wedge T4); i = 5$

例如,NS 模型的启动从 New 状态开始,模拟真实社会的普遍情况,该状态节点进行的初次交易认为属诚信交易,因此,计算得到的诚信度^[4]在 T2、T3 等级,相对贡献度区间是 M_{11} ,对应等级为 C3,由表 3 可知,节点达到 Normal 状态要求,P2P 终端调用 activate 原语完成第一次状态变换。

3.4 NS 模型与评价机制结合后节点状态的变化分析

分析结合后其中较能体现处罚服务的三个状态相互转换的情况,如图 3。节点状态类型从高级 Normal 状态降到 Discreditable 状态时,满足条件: $C2 \wedge T2, status = 3$ 。此时节点表现不够理想,贡献度和可信度等级都较低。

若 Discreditable 状态节点继续大量索取系统资源,造成贡献度等级 C2 降为 C1,即相对贡献度由区间 M_{23} 或 M_{12} 进入区间 M_{13} 时,节点成为性质严重的搭便车者,P2P 终端调用迁移原语 wait 修改节点 status 为 2,进入轻微隔离的 Suspend 状态;

若 Discreditable 状态节点继续进行虚假交易等欺骗行为,使可信度 t 由 T2 降至 T1,则 sleep 原语设置 sleep_t, status 为 1,节点进入深度隔离的 Block 状态;

当 Block 状态节点的 sleep_t 由初始值逐减等于 0 时,节点被 wakeup 原语唤醒进入 Suspend 状态, status = 2;

当 Suspend 状态中严重搭便车者和休眠结束者持续提供良性服务,相对贡献度区间达到 M_{23} 或 M_{12} ,即贡献等级提升到 C2,并且可信度等级为 T2 时,wakeup 原语控制节点回到 Discreditable 状态, status = 3。

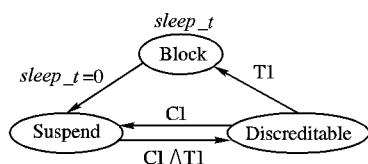


图3 三个状态间的转换

注意:从 Discreditable 状态出发到达的两种隔离惩罚状态中, Suspend 状态节点在满足要求后能直接回到 Discreditable 状态,而 Block 状态节点则不能。这是由于 Block 状态节点属于休眠期,不能通过交易活动提高自身诚信度,因此只能先恢复到能进行基本活动的 Suspend 状态,通过提供服务提高参数等级才能满足 Discreditable 状态的到达条件。这种间接唤醒的转换规则避免了节点长期封锁在 Block 状态的“休眠困境”。

同样由于 Block 状态休眠的特征, Suspend 状态中行为恶化被降级的节点跳过下一级深度隔离 Block 状态直接被打入终结状态,成为跳跃终结型的状态转换规则。

4 实验

为分析文中提出的 NS 模型的扩展性及实用性,本文采用 NS-2^[9] 作为仿真工具构建模拟网络环境进行仿真实验。模拟网络有 250 个节点,节点连接呈幂律 Power-law 分布,节点上均匀分布 2000 个文件。恶意节点比例为 0.05 ~ 0.15,以 90% 的概率提供虚假服务;初始无私节点比例为 0.05。所有节点积极参与交易,被系统断开连接后能重新加入网络。实验中平均每节点发起 500 次交易请求。

在本次仿真实验中,下载资源选择采用似然算法^[4]。相关参数设置:sleep_t 初始值为 60 s。判断搭便车辅助参数 $A = 80, B = 20$ 。EigenTrust 中 $\alpha = 0.3$,每 10 次交易计算一次全局信任值。

观察应用 NS 模型后节点的成功交易率随时间的变化,并与状态划分粒度较大的文献[2]提出的方案比较,如图 4。

图 4 描绘了 NS 模型和文献[2]中只划分成三种状态的成功交易率。由曲线可知,NS 模型提出的细粒度状态划分有相对优势,最高成交率达到 0.75。

在图 4 应用 NS 模型的成功率曲线选取上升期的 t_0, t_1 时刻以及稳定期的 t_2, t_3 时刻,分别测量各类型状态包含的节点数目,结果如图 5 所示,其中:Z 代表 Zombie, B 代表 Block, S 代表 Suspend, D 代表 Discreditable, N 代表 Normal, C 代表

Creditable, Ss 代表 Selfless。低级状态的节点数目减少,高级状态的节点数目增加,说明节点行为逐渐好转,抑制了搭便车及欺骗行为。

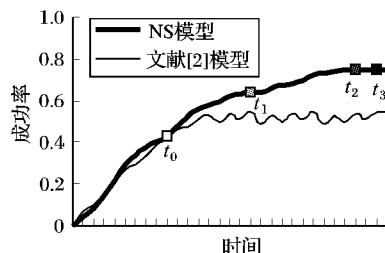


图4 成功交易率对比

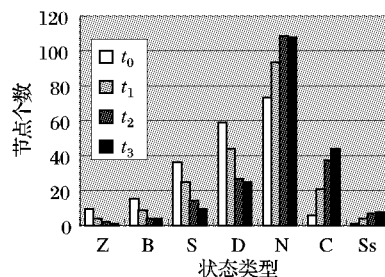


图5 状态节点数变化

实验结果表明,NS 模型与信任机制的成功结合,证明了模型具有良好的可扩展性。基于区分服务思想的转换规则有效地抑制了搭便车行为,激励节点共享资源,提升系统性能。

5 结语

本文提出基于 DFA 的 P2P 网络节点状态模型。模型通过分析节点行为表现的等级特征划分了细粒度的状态类型,定义了控制状态转换的迁移原语和状态转换规则,用状态转换图对节点状态的动态变化过程建模。节点行为评价机制与模型的结合证明了模型可实现并具有良好的扩展性。本文还通过状态隔离,限制服务等措施实现区分服务抑制不良行为。实验结果表明 NS 模型对节点共享资源有明显的激励效应,证明了 NS 模型的实用性。NS 模型尚在初级阶段,可进一步研究节点协作方案及如何减小系统开销,提出基于状态的激励机制,进一步提升 P2P 网络的性能。

参考文献:

- [1] KARAKAYA M, KORPEOGLU I, ULUSOY O. A distributed and measurement-based framework against free riding in peer to peer networks[C] // Proceedings of the 4th International Conference on Peer to Peer Computing. Washington, DC: IEEE Computer Society, 2004: 276 - 277.
- [2] KUNG H T, WU C-H. Differentiated admission for peer-to-peer systems: Incentivizing peers to contribute their resources[C] // Proceedings of the Workshop on Economics of Peer-to-Peer Systems. Berkeley, CA, USA: [s. n.], 2003.
- [3] ABDUL-RAHMAN A, HAILES S. Supporting trust in virtual communities[C] // Proceedings of the 33rd Hawaii International Conference on System Sciences. Washington, DC: IEEE Computer Society, 2000: 6007.
- [4] KAMVAR S D, SCHLOSSER M T, GAREIA - MOLINA H. The EigenTrust algorithm for reputation management in P2P networks [C] // Proceedings of the 12th International Conference on World Wide Web. New York: ACM Press, 2003: 640 - 651.
- [5] ADAR E, HUBERMAN B. Free riding on Gnutella[J]. First Monday, 2000, 5(10): 32 - 35.

- ence on Sensor, Mesh and Ad Hoc Communications and Networks. New York: ACM Press, 2006: 109–118.
- [8] NGAI E C H, LYU M R, LIU J C. A real-time communication framework for wireless sensor-actuator networks [EB/OL]. [2009-01-10]. <http://user.it.uu.se/~edding810/paper/aero06-Oct-25.pdf>.
- [9] 吴震东. 无线传感器网络多节点协同相关问题研究[D]. 杭州: 浙江大学, 2007.
- [10] YUAN H D, MA H D, LIAO H Y. Coordination mechanism in wireless sensor and actor networks [C]// Proceedings of the 1st International Multi-Symposiums on Computer and Computational Sciences. Washington, DC: IEEE Computer Society, 2006: 627–634.
- [11] GUNGOR V C, AKAN ÖZGUR B, AKYILDIZ I F. A real-time and reliable transport (RT)² protocol for wireless sensor and actor networks [J]. IEEE Transactions on Networking, 2008, 16(2): 359–370.
- [12] CHATZIGIANNAKIS I, KINALIS A, NIKOLETSEAS S. Priority based adaptive coordination of wireless sensors and actors [C]// Proceedings of the 2nd ACM International Workshop on Quality of Service & Security for Wireless and Mobile Networks. New York: ACM Press, 2006: 37–44.
- [13] AKAN B O, AKYILDIZ I F. Event-to-sink reliable transport for wireless sensor networks [J]. IEEE Transactions on Networking, 2005, 13(5): 1003–1016.
- [14] BOUKERCHE A, ARAUJO R B, VILLAS L. A novel QoS based routing protocol for wireless actor and sensor networks [C]// Proceedings of the 2007 IEEE Global Telecommunications Conference. Washington, DC: IEEE Computer Society, 2007: 4931–4935.
- [15] CARDEI M, YANG S H, WU J. Algorithms for fault-tolerant topology in heterogeneous wireless sensor networks [J]. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(3): 545–558.
- [16] OZAKI K, WATANABE K, ITAYA S, *et al.* A fault-tolerant model of wireless sensor-actor network [C]// Proceedings of the 9th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing. Los Alamitos: IEEE Computer Society, 2006: 185–193.
- [17] VEDANTHAM R, ZHUANG Z Y, SIVAKUMAR R. Mutual exclusion in wireless sensor and actor networks [C]// Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks. Reston: IEEE Computer Society, 2006: 346–355.
- [18] HOLLAND G, VAIDYA N. Analysis of TCP performance over mobile Ad Hoc networks [J]. Wireless Networks, 2002, 8(2): 275–288.
- [19] SUNDARESAN K, ANANTHARAMAN V, HSIEH R V, *et al.* ATP: A reliable transport protocol for Ad Hoc networks [J]. IEEE Transactions on Mobile Computing, 2005, 4(6): 588–603.
- [20] ABBASI A A, AKKAYA K, YOUNIS M. A distributed connectivity restoration algorithm in wireless sensor and actor networks [C]// Proceedings of the 32nd IEEE Conference on Local Computer Networks. Washington DC: IEEE Computer Society, 2007: 496–503.
- [21] LIU X, XIAO L, KRELING A, *et al.* Optimizing overlay topology by reducing cut vertices [C]// Proceedings of the 2006 International Workshop on Network and Operating Systems Support for Digital Audio and Video. New York: ACM Press, 2006: 17–17.
- [22] OZAKI K, HAYASHIBARA N, TAKIZAWA M. Coordination protocols of multiple actuator nodes in a multi-actuator/multi-sensor model [C]// Proceedings of the 21th International Conference on Advanced Information Networking and Applications Workshops. Washington, DC: IEEE Computer Society, 2007: 62–67.
- [23] OZAKI K, HAYASHIBARA N, ENOKIDO T, *et al.* Fault-tolerant semi-passive coordination protocol for a multi-actuator/multi-sensor (MAMS) model [C]// Proceedings of the 2nd International Conference on Availability, Reliability and Security. Washington, DC: IEEE Computer Society, 2007: 506–516.
- [24] BOUKERCHE A, ARAUJO R B, SILVA F H S, *et al.* Wireless sensor and actor networks context interpretation for the emergency preparedness class of applications [J]. Computer Communications, 2007, 30(13): 2593–2602.
- [25] MUHL G, ULBRICH A, RITTER H. Content evolution driven data propagation in wireless sensor networks [EB/OL]. [2008-11-20]. <http://doc.tn.uka.de/tr/TM-2004-5.pdf>.
- [26] BOUKERCHE A, MARTIROSYAN A. An efficient algorithm for preserving events temporal relationships in wireless sensor actor networks [C]// Proceedings of the 32nd IEEE Conference on Local Computer Networks. Washington, DC: IEEE Computer Society, 2007: 771–780.
- [27] BOUKERCH A, ARAUJO R B, SILVA F H S. An efficient event ordering algorithm that extends the lifetime of wireless actor and sensor networks [J]. Performance Evaluation, 2007, 64(5): 480–494.
- [28] VEDANTHAM R, ZHUANG Z Y, SIVAKUMAR R. Hazard avoidance in wireless sensor and actor networks [J]. Computer Communications, 2006, 29(13/14): 2578–2598.

(上接第2148页)

要操作注册表来指定其控制设备在设备栈中的具体位置,如果恶意操作把注册表中的这些信息删除,那么监控就如同虚设。而这种方法是基于“HOOK”的原理对驱动对象的例程入口地址进行锁定修改,完全不需要对注册表有任何操作,具有非常好的隐藏性,大大地增加了监控不被恶意关闭的安全性。

4 结语

本文针对 WDM 过滤驱动实现移动介质和存储设备监控方法所表现出的不足进行了改进,以另一种思路来实现它,不仅可以完成移动介质和存储设备监控的功能,而且弥补了 WDM 过滤驱动的不足。通过对仿真实验的测试表明,并没有对系统内核造成任何负面的影响。本文进一步的工作即是研究适合 Vista 操作系统及以后的“长角系统”的 KMDF 框架^[9]开发模式来对设备进行控制,实现向上兼容。

参考文献:

- [1] 中国软件与技术服务股份有限公司. 中软防水墙系统 WaterBox™

7.2R5 [Z]. 北京: 中国软件与技术服务股份有限公司, 2008.

- [2] ONEY W. Programming the Microsoft Windows Driver Model [M]. [S. l.]: Microsoft Press, 2003.
- [3] ORWICK P, SMITH G. Developing Drivers with the Microsoft Windows Driver Foundation [M]. [S. l.]: Microsoft Press, 2007.
- [4] SCHMIDT F. The SCSI Bus & IDE Interface Protocols, Applications & Programming [M]. Upper Saddle River: Addison Wesley, 2001.
- [5] HYDE J. USB Design by Example [M]. 2nd ed. [S. l.]: Intel University Press, 2002.
- [6] HOGLUND. IRP hooking and Device Chains [EB/OL]. [2009-02-01]. <http://www.rootkit.com/newsread.php?newsid=846>.
- [7] 楚狂人. 天书夜读 [M]. 北京: 电子工业出版社, 2007.
- [8] CARDMAGIC. NSI ModuleHook: Hiding Port Under Windows Vista [EB/OL]. [2009-02-01]. <http://www.rootkit.com/newsread.php?newsid=735>.
- [9] MICROSOFT. Architecture of the Kernel - Mode Driver Framework [EB/OL]. [2009-02-11]. <http://www.microsoft.com/whdc/driver/driver/wdf/KMDF>.