基于位置与标志分离的访问控制列表优化

姚宇峰[1], 涂 睿[2]
(1. 常熟理工学院 计算机科学与工程学院, 江苏 常熟 215500; 2. 军事经济学院 军需系, 武汉 430035)

摘 要: 在前期工作的 LISA 体系结构下,提出了 IBAC 模型,提供了更加精确和高效的网络访问控制,然而,IBAC 增大了访问控制列表的处理开销。为此,对访问控制列表的规则组织结构进行了分析和优化,模拟实验证明了优化方法的有效性。

关键词: 访问控制; 位置与标志分离; 基于标志的访问控制; 访问控制列表
中图分类号: TP181 **文献标志码**: A **文章编号**: 1001-3695(2010)03-1145-03
doi:10.3969/j.issn.1001-3695.2010.03.094

Optimization of access control list based on locator/identifier split

YAO Yu-feng[1], TU Rui[2]
(1. *School of Computer Science & Engineering, Changshu Institute of Technology, Changshu Jiangsu* 215500, *China*; 2. *Dept. of Quarter-master, Military Economie Academy, Wuhan* 430035, *China*)

Abstract: With the support of locator and identifier separation architecture (LISA), this paper proposed IBAC model which made network access control more accurate and efficient. However, IBAC increased the process costs of ACL(access control list). In order to reduce the process costs, this paper analyzed and optimized the rules structure of ACL. The simulation test results prove the efficiency of the optimization approach.

Key words: access control; locator/identifier split; IBAC(identifier-based access control); ACL

0 引言

在当前的 TCP/IP 体系结构中,IP 地址在语义上具有双重含义,既代表了网络节点的拓扑位置,又是节点的标志,IP 地址成为一个与位置相关的动态可变标志。由于 IP 地址语义过载(IP overloaded)问题的存在[1~3],目前普遍采用的基于 IP 地址验证的访问控制方式面临诸多无法克服的缺陷。

首先,基于 IP 地址的访问控制限制了节点移动条件下的资源访问。一些网络服务使用 IP 地址来区分服务对象,这就导致服务对象绑定了位置,不能满足用户的移动性要求。例如一些网络服务仅限于特定 IP 段的用户访问,若某个用户的网络位置发生了变化,即使其身份合法也无法获得原有的服务。

其次,IP 地址语义过载加大了基于 IP 地址的访问控制复杂性和难度,并在一定程度上影响了访问控制的效能,例如:

a)由于 IP 地址本身的动态可变以及地址欺骗的存在,IP 地址不能够准确反映节点的真实身份,非法用户可以匿名地发动各种形式的攻击,而很难在网络层定位访问源。

b)IP 地址与用户之间没有准确的对应关系[4],不同时刻一个 IP 地址可能对应不同的用户,一个 IP 地址也可能对应多个用户(如 NAT)。这种情况便于网络犯罪的隐藏,增大了各种安全机制的复杂性,并影响其效能。

c)由于情况 a)b)的存在,使得访问控制的效能大打折扣,同时还造成一些误伤,损害了合法用户的利益。

最后,由于网络拓扑或 ISP 自身策略的改变,会导致 IP 地址重分配,使得许多基于 IP 地址的访问控制策略、配置都需要改变。这无疑加大了访问控制管理的复杂性和更新的工作量。

上述缺陷的根源在于网络实体没有精确、惟一的固定标志,为此必须首先解决 IP 地址语义过载问题。国际 IAB 组织提出通过引入两个名字空间来分别表示节点标志和位置,即所谓的 locator/identifier split[5,6] 来解决 IP 地址语义过载问题。Locator/identifier split 引入了固定的节点标志(identifier),从而为准确地识别网络实体(用户、主机、设备)和恶意流量奠定了基础。笔者认为通过将基于 locator/identifier split 的路由和寻址方案与源地址验证技术集成,可以提高网络的安全性。

本文在已完成的位置与标志分离的命名和寻址体系结构框架(LISA)[7]下,提出了一种基于固定标志的访问控制模型(IBAC)。IBAC 支持网络实体的基于固定标志的身份识别,提供了准确、高效的精细粒度访问控制机制。通过采用不依赖于用户位置信息访问控制机制,使得安全策略的管理和执行更加直观和有效。

访问控制列表(ACL)是一种普遍采用的网络访问控制机制。ACL 的处理采用顺序匹配规则的方式,匹配规则的过程会导致报文的延迟。当规则表项较少时,这种延迟可以忽略,但如果表项庞大,就会对报文延迟产生一定的影响。IBAC 提供了细粒度的访问控制,ACL 的规则粒度变小,加之 identifier 本身的规模,导致基于 IBAC 的 ACL 的表项规模较大。此外,基于 IBAC 的访问控制处理过程也发生了变化,带来额外的处理延迟,详见第 2 章。在 IBAC 中,ACL 的处理延迟成为影响报文处理性能的一个重要因素。因此,需要针对 IBAC 的特

收稿日期: 2009-07-05; **修回日期**: 2009-08-27

作者简介:姚宇峰(1981-),男,江苏苏州人,助教,主要研究方向为通信运营支撑软件、数据挖掘(fengsingal81@gmail.com);涂睿(1980-),男,讲师,博士,主要研究方向为计算机网络通信.

性,对 ACL 的规则组织结构进行优化,减少报文的处理延迟。

1　LISA 体系结构

LISA 采用了基于网络的 locator/identifier split 命名和寻址体系结构,如图 1 所示。本文将网络划分为核心网络和边缘网络,核心网络使用 locator 名字空间,边缘网络使用固定的 identifier名字空间。通信会话基于固定的 identifier,但 identifier 对应的 locator 可以变化。

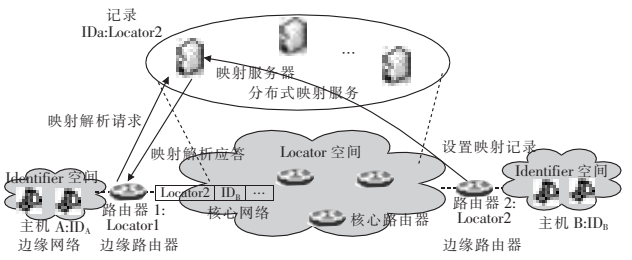

图 1　基于网络的命名和寻址体系结构(LISA)

为了支持报文在两个名字空间的寻址,LISA 采用了“LISA in IP”的报文封装方式。LISA 路由器(边缘路由器)通过查询 LISA-Mapping 映射服务,完成 identifier 到 locator 的映射。LISA-Mapping是一个基于单跳 DHT 的分布式映射服务系统[8]。Identifier 是一个新的扁平名字空间,有利于支持移动性,也可以不受限制地表示为特定意义的名字空间。Locator 沿用了已有的 IP 地址空间(IPv4/IPv6),这样可以避免对大量核心网络设备的更新。LISA 路由器收到主机发送的报文后,根据报文包含的 identifier 信息,查询分布式映射服务系统,并获得匹配的 locator 记录。LISA 路由器随后将一个包含 locator 的新报文头附加在当前的报文上。在新报文中,内部报头的源和目的地址是 identifier,外部报头的源和目的地址是 locator。LISA 用 identifier 来表示网络节点,在核心网络中根据 locator 来转发封装后的报文。当一个经过封装后的报文到达目的地 LISA 路由器时,路由器将解封该报文,并根据 identifier 将其发送到目的地主机。

2　IBAC 模型

与传统的访问控制不同,IBAC 模型的访问控制策略基于网络实体的真实固定标志,而不是 IP 地址或者网络设备端口。IBAC 提供了端到端的安全机制,细化了访问控制的粒度。例如,即使多个用户共享一个 locator(如 IP 地址),IBAC 也可以根据 identifier 为单个用户指定安全策略。

IBAC 保证了访问控制策略的长期稳定性。虽然网络节点在移动过程中 locator 会发生变化,但基于固定 identifier 的访问控制策略不需要改变,合法用户可以继续访问相关服务。IBAC 适合移动节点的访问控制,避免了由于 locator 变化所导致的访问控制更新,减轻了访问控制管理的复杂性和工作量。

由于 identifier 是一种无结构的扁平标志,也就无法像 IP 地址一样进行聚合。在这种情况下,IBAC 只能对每个 identifier 制定访问控制策略,而无法对策略进行归类,从而导致访问控制列表规模膨胀。为了简化访问控制策略,控制 ACL 的规模,本文引入了标志归属(identifier affiliation)的概念。

定义 1　标志归属。Identifier 对应用户、设备所归属的组织。

IBAC 由三个实体组成:标志(I)、对象(O)、权限(P)。其中,标志又划分为个体标志(individual identifier,I^2)和标志归属(identifier affiliation)。

IBAC 的授权用三元组(I,O,P)表示,如果存在元组(I,O,P),则表明 I 可在 O 上执行 P 许可。访问控制策略在语法上包含两种形式:基于个体标志(I^2)的表项和基于标志归属(IA)的表项。其中,基于标志归属的表项是对若干基于个体标志的表项的合并。如果存在元组(I^2,O,P),则表明单个 I 可在 O 上执行 P 许可;如果存在元组(IA,O,P),则表明一组 I 可在 O 上执行 P 许可。

对应不同类型的访问控制表项,存在两种不同的处理过程。对基于 I^2 的表项进行查询和匹配时,由于报文中已经携带了源 identifier,处理过程类似基于 IP 的访问控制列表的处理。在对基于 IA 的表项进行处理时,由于报文没有直接携带 IA,需要目的端安全设备根据报文的源 identifier 向 LISA-Mapping 映射服务系统查询,获得响应的 IA,再根据 IA 查询访问控制表项。为了支持基于 IA 的访问控制策略,需要对 LISA-Mapping 映射服务系统进行扩展:每条映射记录不仅包括 locator,还包括 identifier 对应的 IA 信息。

引入了标志归属后,达到了标志聚合的效果,可以针对一个机构制定访问控制策略,从而减少访问控制策略表项规模。所带来的额外操作是机构需要事先向 LISA-Mapping 映射服务系统注册本机构成员的 IA 信息。本文以基于 IA 的访问控制为例,给出整个访问控制过程,如图 2 所示。

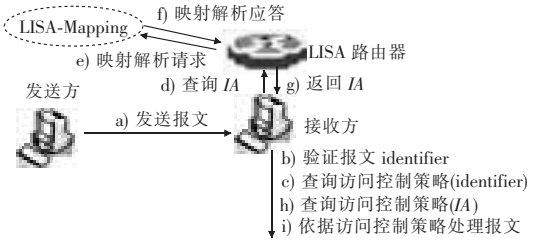

图 2　基于 IA 的访问控制过程

3　ACL 规则优化

基于 IBAC 的 ACL 的规则粒度变小,加之 identifier 本身的规模较大,导致基于 IBAC 的 ACL 的表项规模较大。此外,基于 IBAC 的访问控制处理过程引入了基于 IA 型规则的访问控制,带来额外的查询延迟。因此,在 IBAC 中,ACL 的处理延迟成为影响报文处理性能的一个重要因素。为了提高 IBAC 的实用性,需要对 ACL 进行优化,以减少处理开销。

3.1　模型描述

首先定义 I 为所有可用的 identifier 的集合,由于 identifier 包含 I^2 和 IA 两种类型,为了以后讨论的方便,重新定义 $I = I^2 \cup IA$;定义 P 为当前支持的协议组。

定义 2　规则:$r_i = (op_i, pr_i, sI_i, dI_i)$。其中:$op_i \in \{\text{permit}, \text{deny}\}$,为规则执行的操作;$pr_i \in P$,为规则处理的协议对象;$sI_i \in I, dI_i \in I$,分别为规则限定的源 identifier 和目的 identifier。四元组中,只有 op_i 是每条规则所必需的,其他元素省略时的默认值为 $pr_i = P, sI_i = dI_i = I$。

定义 3　报文:$p_k = (pr_k, sI_k, dI_k, sp_k, dp_k)$。其中:$pr_k \in P$,为报文的协议;$sI_k \in I, dI_k \in I$,分别为报文携带的源 identifier 和目的 identifier;sp_k 和 dp_k 分别表示报文的源端口和目的端口。

若报文 p_k 与规则 r_i 匹配,记做 $p_k \odot r_i$,定义为

$$p_k \odot r_i = (pr_k \in pr_i) \wedge (sI_k \in sI_i) \wedge (dI_k \in dI_i) \quad (1)$$

3.2　优化方法

ACL 优化目标:本文设定 ACL 的规则序列 $R=\{r_1,r_2,\cdots,r_n\}$,对于 $r_i \in R$,为 R 中的第 i 条规则;设定流 $T=\{p_1,p_2,\cdots,p_m\}$ 为一段时间的报文序列。

定义 4　等效 ACL。设定两个 ACL 规则序列 R 和 R',以及报文流 T,若对于任意报文 $p_k \in T$,R 和 R' 的处理结果相同,则称 R 和 R' 等效,记做:$R \equiv R'$。

设定报文与一条规则匹配操作的耗时为函数 $T(r_i)$。对于 I^2 型规则,由于仅限本地处理,考虑长期大量报文的处理,处理时间趋近恒定,设为 λ。对于 *IA* 型规则,涉及远程查询和映射解析,处理时间变化较大,设为函数 $p(r_i)=\text{query}(sI_k)+\lambda$。规则处理耗时函数可以表示为

$$T(r_i)=\begin{cases}\text{query}(SI_i)+\lambda & \text{if } r_i \in IA\\ \lambda & \text{if } r_i \in I^2\end{cases} \quad (2)$$

定义 5　累积处理时间。对于 ACL 规则序列 R 的任意一条规则 r_i,系统在处理 r_i 及其之前的规则时总共消耗的时间,记做 $cpt(r_i(R))$。

$$cpt(r_i)=\sum_{j=1}^{i}T(r_j) \quad (3)$$

设 $E(T,R)$ 表示报文流 T 在某一 ACL 规则序列上的处理期望时间,则 ACL 的优化目标为:求使得 $E(T,R)$ 最小的 R 的等效 ACL,即在不改变 ACL 执行结果的条件下,通过修改规则序列的结构,使报文流 T 的处理期望时间最小。

设定报文流 T 中报文与规则序列 R 中任意规则匹配的概率 $f(i)$,$\sum_{i=1}^{n}f(i)=1$,则报文流在 R 上的处理期望时间为

$$E(T,R)=\sum_{i=1}^{n}f(i)cpt(r_i(R))=\sum_{i=1}^{n}f(i)\sum_{j=1}^{i}T(r_j(R)) \quad (4)$$

定义 6　规则冗余。*IA* 型规则可以看做一组 I^2 型规则的集合,因此在一个 ACL 的规则序列中可能出现一条 I^2 型规则在语义上被包含在一条 *IA* 型规则中的情况,即规则冗余。

对于规则序列 R 中的两条规则:r_i 和 r_j,其中 $r_i \in I^2$,$r_j \in IA$,若所有匹配 r_i 报文均能被 r_j 匹配;反之不成立,则称 r_i 规则冗余,记做 $r_i << r_j$。

$$r_i << r_j \Leftrightarrow (op_i = op_j) \wedge (pr_i \subseteq pr_j) \wedge (sI_i \in sI_j) \wedge (dI_i \in dI_j) \quad (5)$$

由于 ACL 的规则序列是顺序执行的,冗余规则的执行顺序不同,ACL 执行时间也不同。对 I^2 型规则 r_i 和 *IA* 型规则 r_j:

a)$i<j$。这种情况下,r_i 先于 r_j 执行,一部分报文不用进行远程查询,使得整个 ACL 的处理得到加速。因此,本文称这种类型的冗余为良性冗余,予以维持。但是,良性冗余不是越多越好。一方面,原本简洁的规则序列会变得复杂;另一方面,随着表项的增多,会抵消不进行远程查询所节省的处理时间。

b)$i>j$。r_i 和 r_j 所针对的报文对象相同,由于 ACL 的规则序列采用顺序一次执行的方式,被 r_j 执行的报文不会被 r_i 再次执行,r_i 实际多余。对于规则序列 $R:\{r_1,r_2,\cdots,r_j,\cdots,r_i,\cdots,r_n\}$,如果删除 r_i,得到新的规则 $R':\{r_1,r_2,\cdots,r_j,r_{i-1},r_{i+1},\cdots,r_n\}$。易知,$R \equiv R'$。

由于规则 r_i 不会被执行,还给 r_i 之后的规则 $r_x(x>i)$ 增加了一次匹配操作,从而增大了 $E(T,R')$。检测到这种类型的规则冗余,可以考虑将 r_i 删除。

设定冗余规则 r_i,现讨论 $E(T,R')$,易知只有序号在 r_i 之后的规则会受到 r_i 删除的影响,由式(2)(4)得到:

$$E(T,R)-E(T,R')=\lambda\sum_{k=i+1}^{n}f(k) \quad (6)$$

除了简单地删除 r_i 外,还可以考虑交换两条冗余规则的位置来达到减小 $E(T,R)$ 的目的。通过把冗余的 I^2 型规则提到 *IA* 型之前,可以使得部分报文不需要进行远程查询,从而整体上减少了处理时间。

对于规则序列 $R:\{r_1,r_2,\cdots,r_j,\cdots,r_i,\cdots,r_n\}$,如果交换 r_i 和 r_j,得到新的规则 $R'':\{r_1,r_2,\cdots,r_i,\cdots,r_j,\cdots,r_n\}$。易知,$R \equiv R''$。性能提升一方面来自原本匹配 r_j 的报文不再需要远程查询,一方面来自匹配 $r_k(i<k<i)$ 的报文减少了一次不命中的远程查询。减少的处理期望时间:

$$E(T,R)-E(T,R'')=\sum_{k=j+1}^{i-1}f(k)\,\text{query}(sI_k)+f(i)\,\text{query}(sI_j) \quad (7)$$

设删除和交换两种方式产生的性能提升差异为 $\text{diff}(R'',R'')$,由式(6)(7)得到

$$\text{diff}(R',R'')=\lambda\sum_{k=i+1}^{n}f(k)-\sum_{k=j+1}^{i-1}f(k)\,\text{query}(sI_k)-f(i)\,\text{query}(sI_j) \quad (8)$$

因此,若 $\text{diff}(R',R'')\geqslant 0$,则采用删除的方式;若 $\text{diff}(R',R'')<0$,则采用交换的方式。

处理原则:I^2 型规则在前,*IA* 型规则在后。

证明　对于规则序列 $R:\{r_1,r_2,\cdots,r_i,\cdots,r_j,\cdots,r_n\}$,其中,$r_i$ 为 I^2 型规则,r_j 为 *IA* 型规则,如果交换 r_i 和 r_j,得到新的规则 $R'':\{r_1,r_2,\cdots,r_j,\cdots,r_i,\cdots,r_n\}$。易知,$R \equiv R''$。匹配 $r_k(i<k<i)$ 的报文减少了一次不命中的远程查询,由式(2)(4)得到减少的处理期望时间:

$$\Delta E(T,R)=\sum_{k=j+1}^{i-1}f(k)\,\text{query}(sI_k)>0 \quad (9)$$

可见,将 *IA* 型规则尽量后移,能减少 ACL 的处理期望时间。

3.3　模拟实验结果

本文使用 3.2 节的优化原则对一组不同规模和规则结构的原始 ACL 进行优化,得到一组新的 ACL。通过随机生成的 10 万个模拟 LISA 报文,测量数据报文在新旧 ACL 上的总运行时间。

在不改变访问控制结果正确性的前提下,对于不同规模的 ACL,本文测试了删除冗余 I^2 型规则,交换 I^2 型和 *IA* 型规则两种情况。对于前一种情况,本文在构造 ACL 时仅应用删除操作;对于后一种情况,本文在构造 ACL 时仅应用交换操作。表 1 显示了对于不同规模的 ACL,优化前后的运行时间比。实验结果表明:优化后的 ACL 执行时间有所减少,并且通过交换的方式能比删除获得更大的性能提升。此外,当 ACL 规模较大时,通过删除冗余 I^2 型规则很难再获得较大的性能提升。然而,随着 ACL 规则数目的增多,交换 I^2 型和 *IA* 型规则能获得更大的性能提升。

表 1　ACL 规则优化后的性能对比

ACL 规模(规则数)	规则删除($\text{time}_{new}/\text{time}_{old}$)/%	规则交换($\text{time}_{new}/\text{time}_{old}$)/%
50	98.27	96.78
100	98.42	95.12
500	97.31	94.53
1 000	97.65	92.34

4　结束语

IBAC 模型的 ACL 具有不同于基于 IP 的 ACL 的特殊属性,在一定程度上增大的访问控制列表的处理 (下转第 1150 页)

4　算法的模拟仿真与分析

为了检验算法的正确性和效率,本文对算法进行了模拟仿真。利用 3.1 节所描述的方法对图 2 所示的网络构造逻辑环。

图中共有 36 个节点,圆圈表示路由器,被虚线框包围的路由器构成了区,虚线框外的黑色节点表示管辖该区的根服务器。最后得到的逻辑环的拓扑结构如图 3 所示。

图 2　模拟仿真采用的网络拓扑结构

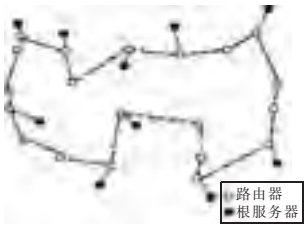

图 3　逻辑环的拓扑结构

本文按照图 3 所示的逻辑环的拓扑结构布置了局域网,作为实验网络。网内设置了 30 个节点,其中 10 个被指定为根服务器,其他节点为路由节点。将用本文所述算法实现的程序放到根服务器上,某个根服务器上的文本文件 Text. txt 作为根服务器访问的临界资源。所有根服务器均以时间上服从均匀分布的方式发出临界资源访问请求,访问临界资源的时间统一设置为 30 ms,发送请求消息的间隔分别设置为 10 ms、20 ms、50 ms、100 ms、200 ms、500 ms 和 1 000 ms。最后分别对平均消息数、平均响应时间和响应时间的标准差这三个参数进行分析。

4.1　平均消息数

平均消息数是指平均每个节点访问临界资源的过程中产生的消息数。图 4 为仿真实验所得到的关于平均消息数的图。

如图 4 所示,发送的消息数随着请求间隔的增大而增大。请求间隔小,则表示单位时间内发送的请求数目多,那么请求消息在节点被阻塞的几率就会增大,因此消息数目较少。当请求间隔很大时,请求消息基本不会被节点阻塞,而是直接发送到令牌持有者,因此消息数目多。

4.2　平均响应时间

平均响应时间是指从发送请求消息到接收令牌平均需要经过的时间。图 5 为仿真实验得到的关于平均响应时间的图。

平均响应时间为在请求队列中的等待时间与令牌消息在链路上的传输时间的总和。一般构建逻辑网的时间远大于令牌传输时间,因此平均响应时间主要取决于在请求队列的等待时间。从图中可以看出,单位时间内发送的请求消息越多,令牌的响应时间越长。

4.3　响应时间的标准差

响应时间的标准差可以用来衡量令牌响应时间的波动性。图 6 为响应时间标准差的仿真结果。

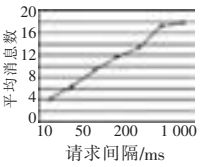

图 4　平均消息数的仿真实验结果

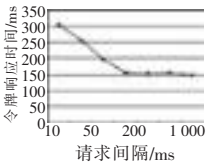

图 5　平均响应时间的仿真实验结果

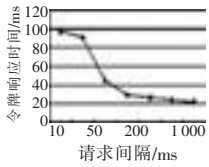

图 6　响应时间标准差的仿真实验结果

从图 6 中可以看出,请求间隔越大,响应时间的标准差就越小。这是因为请求间隔越短,请求消息就会越多,那么请求消息队列就会越长。这就导致了有些请求可能等待较长时间,有些能马上得到响应。当请求间隔增大到一定限度时,请求消息队列只有很少的请求,那么令牌的响应时间约等于消息传递的时间,因此响应时间基本相等。

5　结束语

本文针对一体化承载网在构建逻辑网时的互斥问题,结合一体化承载网体系结构的特点,提出了一种基于令牌的分布式互斥算法。算法借鉴了解决旅行商问题的思想,构造了一个最优的逻辑环,使得令牌传递一周的时间最小;并且提出了一种新的令牌传递策略,该策略能够有效地降低系统的通信量,尤其是在请求负荷较高的情况下。本文对所提出的互斥算法进行了模拟仿真实验,实验结果表明,该算法能很好地解决一体化承载网中的互斥问题。

参考文献:

[1] 王浩学,汪斌强,于婧,等. 一体化承载网络体系架构研究[J]. 计算机学报,2009,32(3):371-376.

[2] 张栋,吴春明,姜明. 分布式系统中资源分配的一致性算法综述[J]. 信息工程大学学报,2009,10(1):37-40.

[3] LAMPORT L. Time, clocks, and the ordering of events in a distributed system[J]. Communications of the ACM,1978,21(7):558-565.

[4] MAEKAWA M. A sqrt(n) algorithm for mutual exclusion in decentralized systems[J]. ACM Trans on Computer Systems,1985,3(2):145-159.

[5] 李云鹤. 一种基于令牌的新的互斥算法分析与设计[J]. 计算机科学,2008,35(4):119-121.

[6] RAZZAQUE M A, HONG C S. Multi-token distributed mutual exclusion algorithm[C]//Proc of the 22nd International Conference on Advanced Information Networking and Applications. Washington DC: IEEE Computer Society,2008:963-970.

[7] 吴春明,张栋,姜明. 面向服务提供的一体化承载网体系结构的探讨[J]. 信息工程大学学报,2009,10(1):23-27.

[8] 李敏,吴浪,张开碧. 求解旅行商问题的几种算法的比较研究[J]. 重庆邮电大学学报:自然科学版,2008,20(5):624-630.

(上接第 1147 页)开销。为了提高访问控制列表的查询效率,减少延迟,本文对 IBAC 模型的访问控制列表的组织结构进行了优化。模拟实验证明经过优化的访问控制列表处理时间有所减少,提高了 IBAC 的实用性。

参考文献:

[1] SCUDDER J. Routing/Addressing problem solution space[EB/OL]. (2007). http://www. arin. net/meetings/minutes/ARIN_XX/PDF/wednesday/SolutionSpace_Scudder. pdf.

[2] CLARK D, BRADEN R, PALK A, *et al*. FARA: reorganizing the addressing architecture[C]//Proc of ACM SIGCOMM'03 Workshops. New York: ACM Press,2003:313-321.

[3] SALTZER J. RFC 1498, On the naming and binding of network destinations[S]. 1993.

[4] TU Rui, SU Jin-shu, MENG Zhao-wei, *et al*. UCEN: user centric enterprise network[C]//Proc of IEEE ICACT. 2008:66-71.

[5] MEYER D, FALL K. Report from the IAB workshop on routing and addressing[S]. 2006.

[6] 涂睿,苏金树,彭伟. 位置与标识分离的命名和寻址体系结构研究综述[J]. 计算机研究与发展,2009,46(11):1777-1786.

[7] 涂睿,苏金树. 一种基于位置/标识分离的站点多宿主路径失效恢复机制[J]. 计算机科学, 2009,36(10):49-54.

[8] 涂睿,苏金树. 一种基于 hash 的位置与标识分离的映射机制[R]. 长沙:国防科技大学,2009.