ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.18, No.4, April 2007, pp.978–986
DOI: 10.1360/jos180978

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 访问控制列表的优化问题*

曾旷怡+, 杨家海

(清华大学 信息网络工程研究中心,北京 100084)

## Towards the Optimization of Access Control List

ZENG Kuang-Yi+, YANG Jia-Hai

(Network Research Center, Tsinghua University, Beijing 100084, China)

+ Corresponding author: Phn: +86-10-62777146, E-mail: zengkuangyi@tsinghua.org.cn

**Zeng KY, Yang JH. Towards the optimization of access control list. *Journal of Software*, 2007,18(4):978–986.**
http://www.jos.org.cn/1000-9825/18/978.htm

**Abstract**: Access control list (ACL) is proposed to solve or improve the network security problem. It is widely deployed in network devices such as routers, switches and firewall appliances, to filter the packets. However, the performance of the network device will be degraded when access control lists are applied in data forwarding interfaces of the device. The optimization of the ACL can greatly improve the performance of the devices in packets forwarding. The paper studies the optimization problem of ACL, outlines the overlapping or containing relationships between single clause and multiple clauses or among multiple clauses, proposes a formula representation of the problem based on the studies, and draws three important conclusions. Based on these conclusions, an approximate optimization algorithm is designed and implemented. Simulation experiments show better performance than the similar commercial products, implying that the research not only provides theoretical references, but also has important practical application.

**Key words**: network management; network security; access control list; packet filter; optimization

摘 要: 访问控制列表(access control list,简称 ACL)是解决和提高网络安全性的方法之一,但访问控制列表应用在网络设备的接口上将降低网络设备的性能.当 ACL 条目达到一定数量后,很难进行人工处理,根据一定算法进行 ACL 自动优化显得尤为重要.在深入研究 ACL 优化问题的基础上,考虑到一条语句与多条语句之间或多条语句与多条语句之间的交叉覆盖或包含关系,对 ACL 的全局优化问题进行了形式化描述,得出了 3 个有用的推论,并提出了一种 ACL 的近似优化算法.通过模拟实验表明,性能优于同类商业产品.该算法可以作为 ACL 优化研究方面的参考,通过进一步研究,推出相关产品.

关键词: 网络管理;网络安全;访问控制列表;数据包过滤;优化
中图法分类号: TP393 文献标识码: A

* Supported by the National Natural Science Foundation of China under Grant No.60473083 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2003AA103110, 2005AA103110-2 (国家高技术研究发展计划(863))

Received 2005-09-26; Accepted 2006-04-03

ACL(access control list)最初是 Cisco IOS 所提供的一种访问控制技术.ACL 使用包过滤技术,应用在网络设备的接口上;支持 ACL 技术的网络设备,根据预先定义好的规则对数据包进行过滤,从而达到访问控制的目的.当 ACL 条目达到一定数量后,如果人工判读、编辑和优化,不仅会增加网络管理员繁琐的工作量,而且容易出错,甚至由此带来灾难性的后果.

随着现代高速信息网络的飞速发展,路由转发任务十分繁重,在研究路由查找算法[1]的同时,现行的网管产品提供了 ACL 的优化功能,用来提高数据包的转发速率,但没有考虑一条语句与多条语句之间或多条语句与多条语句之间的交叉覆盖或包含关系.本文作者曾长期负责中国教育和科研计算机网 CERNET 全国主干网的运行管理工作,对相关问题有切身的体会,在此基础上,对 ACL 的优化问题进行了较深入的思考和探索.本文是对相关工作的总结.

## 1 背景知识及相关工作

### 1.1 ACL简介

一个 ACL(如图 1 所示)按其过滤范围来划分,可分为标准 ACL 和扩展 ACL 两种.标准 ACL 只基于源地址,扩展 ACL 的过滤条件则包括协议类型、源地址、目的地址、源端口和目的端口等信息.ACL 的执行顺序是自上而下的,当数据包到达时,将数据包中的信息与 ACL 中的语句按过滤条件依次匹配,一旦与某条语句匹配成功,则停止匹配并执行相应语句定义的规则(permit 或 deny).在每个 ACL 的末尾,都有一条对所有数据包的隐含拒绝语句(deny any any).如果数据包在该语句前没有匹配成功,那么它最终将与这条隐含语句相匹配.ACL 实现访问控制形式灵活、用途广泛,但也有其固有局限性,比如,要达到端到端的权限控制目的,需要与系统级及应用级的访问权限控制[2]结合使用,还需要设备的支持等.

```
access-list 100 permit icmp any 11.1.1.1 0.0.0.1 echo
access-list 100 permit udp 10.10.0.0 0.0.255.255 eq bootpc 12.12.5.6 0.0.0.1 eq bootps
access-list 100 permit udp 27.24.1.0 0.0.0.255 eq 1024 6.12.1.45 0.0.0.1 eq bootps
access-list 100 permit tcp host 18.14.5.2 gt 1023 host 6.31.48.35 eq 31388
access-list 100 permit udp 65.9.70.0 0.0.0.15 eq 1645 host 63.3.1.92 eq 1645
access-list 100 permit icmp any 16.51.68.120 0.0.0.1 echo
access-list 100 permit udp 29.124.248.0 0.0.127.255 eq snmp host 90.41.82.120 gt 1023
access-list 100 permit tcp 44.132.0.0 0.0.255.255 gt 1023 36.4.128.110 0.0.0.1 eq www
deny any any (implicit)
```

Fig.1 An example of access control list (ACL)

图 1 访问控制列表(ACL)实例

ACL 语句的组成可分为两部分:条件部分和执行部分.执行部分只有两个动作:permit 或 deny;条件部分可由多个单元组成,见表 1.本文中若无特别说明均为 Cisco IOS 规范[3].

**Table 1** Approximate structure of ACL's clause

**表 1** ACL 语句结构示意图

| Access-List number | Rule | Pattern and expression definition | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1-99, 100-199, etc. | Permit or deny | IP, ICMP, TCP, UDP, etc. | Resource IP address or mask | Resource port | Destination IP address or mask | Destination port | eq=equal gt=greater than lt=less than neq=not equal | Others: in/out/ log/echo/ established/ etc. |

### 1.2 相关工作介绍

在研究和解决网络设备转发数据包时的冲突问题中,文献[4]最先提出数据包过滤的相关性问题,Cisco 公司推出了 Cisco ACL Manager[5],其中提供了 ACL 优化功能,但功能比较有限.2003 年,Bukhatwa 分析了优化 ACL

的意义和价值及语句间的相关性问题[6],一年之后,他又提出了一种简单的近似优化算法,但没有解决语句之间的相关性问题.文献[7]使用语句命中率的概念,在假设 ACL 语句已经解决相关性问题的前提下,通过优化语句的排列顺序来解决 ACL 的优化问题,使用了 K-OPT 等近似求解算法,还证明了求最优排列顺序,即求 ACL 的最少处理时间,是 NPC 问题.尽管引入语句命中率有利于问题的形式化描述,但命中率本身是难以获取并实时变化的,并且还没有考虑语句之间的冗余,实际应用价值依然有限.

目前,提供 ACL 优化功能的商业产品主要有 Cisco 公司的 ACL Manager,Telos 公司的 XACTA ACL Manager 等.

## 2 ACL 优化问题分析

### 2.1 ACL优化目标

设有 ACL:$L=\{r_1,r_2,r_3,\ldots,r_n\}$,其中:$r_i\in L$,表示 $L$ 中第 $i$ 条语句;数据包 $P=\{p_1,p_2,p_3,\ldots,p_m\}$为网络上所有可能出现的数据包的集合.

定义 **1**. 如果两个 ACL,$L$ 与 $L'$,对任意数据包 $p_i\in P$ 都有相同的处理结果,则称 $L'$为 $L$ 的等效 ACL.

设 $E_t$ 表示 $L$ 在某一设备上的执行期望时间,表示处理一个数据包的平均时间消耗,则 ACL 的优化目标为:求使得 $E_t$ 最小的 $L$ 的等效 ACL.即在不改变 ACL 执行结果的前提下,修改 ACL,使其执行期望时间最小.

### 2.2 ACL优化问题分析

一条 ACL 语句定义了该语句的规则所适用的数据包集合,将其记为 $r_i=\{PO_i,SA_i,SP_i,DA_i,DP_i\}$.其中:$PO_i$ 表示协议范围;$SA_i$ 表示源地址范围;$SP_i$ 表示源端口范围;$DA_i$ 表示目的地址范围;$DP_i$ 表示目的端口范围.数据包根据其携带的信息可记为五元组:$p_k=\{po_k,sa_k,sp_k,da_k,dp_k\}$,分别表示数据包的协议、源地址、源端口、目的地址、目的端口.

若数据包 $p_k$ 在语句 $r_i$ 定义的范围之内,记作 $p_k\in r_i$,其含义为

$$p_k\in r_i\Leftrightarrow(po_k\in PO_i)\wedge(sa_k\in SA_i)\wedge(sp_k\in SP_i)\wedge(da_k\in DA_i)\wedge(dp_k\in DP_i).$$

一个数据包与一条语句匹配操作的耗时为$\lambda$,在数据包数量巨大的条件下,$\lambda$的平均值趋于稳定,故可以假设$\lambda$为一常数.假设网络上数据包 $p_k(p_k\in P)$出现的概率为$\varphi(k)$, $\sum_{k=1}^{m}\varphi(k)=1$,可以通过对一段时间内网络的流量进行统计来获取.该数据包与第 $f(k)$条语句首次匹配,$f(k)$是数据包标号到语句序号的函数,$f(k)$的值域为$\{1,2,\ldots,n\}$,处理该数据包一共耗时 $t(k)$,有 $t(k)=\lambda f(k)$,则 $L$ 的执行期望时间为

$$E_t=\sum_{k=1}^{m}\varphi(k)t(k)=\lambda\sum_{k=1}^{m}\varphi(k)f(k)=F(\varphi,f),$$

上式中的 $E_t$ 是函数$\varphi(k)$和 $f(k)$的泛函,其中,$\varphi(k)$与 ACL 无关,且在一段时间内不变.

假设 $L$ 中存在语句 $r_j\in L$ 被其前面的语句所覆盖,即 $r_j\subseteq\bigcup_{i=1}^{j-1}r_i$ ,由于 ACL 在执行中采用按顺序匹配方式,对任意数据包 $p_k\in P$,如果 $r_j$ 可以与 $p_k$ 匹配,则必存在语句 $r_u(u<j)$与 $p_k$ 匹配,则在执行过程中,由 $r_u$ 决定数据包 $p_k$ 的执行规则,$r_j$ 不会被执行.如果删除语句 $r_j$,得到新的 ACL:

$$L'=\{r_1',r_2',\ldots,r_{n-1}'\},$$

其中,

$$\begin{cases}r_i'=r_i & (i<j)\\ r_i'=r_{i+1} & (i\geq j)\end{cases}.$$

显然,$L'$是 $L$ 的等效 ACL.

新的映射关系为 $f'(k)$,则有

$$\begin{cases} f'(k)=f(k), & k\in K_1 \\ f'(k)=f(k)-1, & k\in K_2 \end{cases},$$

其中,

$$\begin{cases} K_1=\{k\mid 1\le k\le m,\text{且}f(k)<j\} \\ K_2=\{k\mid 1\le k\le m,\text{且}f(k)>j\} \end{cases},$$

且

$$\|K_1\|+\|K_2\|=m.$$

设 $L'$的执行期望时间为 $E_t'$, $E_t'=\lambda\sum_{k-1}^{m}\varphi(k)f'(k)$.

$$\begin{aligned} E-E_t' &= \lambda\left\{\sum_{k=1}^{m}\varphi(k)\left[f(k)-f'(k)\right]\right\} \\ &= \lambda\left\{\sum_{k\in K_1}\varphi(k)\left[f(k)-f'(k)\right]+\sum_{k\in K_2}\varphi(k)\left[f(k)-f'(k)\right]\right\} \\ &= \lambda\left\{\sum_{k\in K_2}\varphi(k)\left[f(k)-f(k)+1\right]\right\}\ge 0. \end{aligned}$$

即 $E_t'\le E_t$,由此可得:

推论 **1**. 删除 ACL 中被前面语句覆盖的语句,ACL 的执行期望时间减少.

定义 **2**. 对 $L$ 中两条相邻语句 $r_j$,$r_{j+1}$,如果可以构造一条新的 ACL 语句 $r_\eta$,使得 $r_\eta=r_j\cup r_{j+1}$,即$\{r_\eta\}$和$\{r_j,r_{j+1}\}$对任意数据包 $p_k\in P$ 有相同的执行结果,则称语句 $r_j$,$r_{j+1}$ 连续,称 $r_\eta$为 $r_j$,$r_{j+1}$ 的合并.

将 $L$ 中的连续语句 $r_j$,$r_{j+1}$ 删除,并在原来 $r_j$ 的位置插入 $r_j$,$r_{j+1}$ 的合并 $r_\eta$,得到

$$L'=\{r_1',r_2',...,r_{n-1}'\},$$

其中,

$$\begin{cases} r_i'=r_i, & i<j \\ r_j'=r_\eta & \\ r_i'=r_{i+1}, & i>j \end{cases}.$$

$L'$是 $L$ 的等效 ACL.与推论 1 的证明类似,可得:

推论 **2**. 合并 ACL 中的相邻连续语句,ACL 的执行期望时间减少.

定义 **3**. 对 $L$ 中的两条语句 $r_j,r_\ell(j<\ell)$,如果交换 $r_j,r_\ell$ 的位置,得到的 ACL 与 $L$ 等效,则称 $r_j,r_\ell$ 为可交换的.

定义 **4**. 如果 $L$ 中存在子语句序列 $\{r_j,r_{j+1},...,r_{\ell-1},r_\ell\}$.其中: $r_j,r_\ell$ 可交换,且

$$r_j\cap\left(\bigcup_{i=j+1}^{\ell-1}r_i\right)=\varnothing,\ r_\ell\cap\left(\bigcup_{i=j+1}^{\ell-1}r_i\right)=\varnothing,$$

则称 $r_j,r_\ell$ 为无冲突可交换的.

交换 $r_j,r_\ell$ 得到 $L$ 的等效 ACL:

$$L'=\{r_1',r_2',...,r_n'\},$$

其中,

$$\begin{cases} r_i'=r_i, & i\ne j\text{且}i\ne\ell \\ r_j'=r_\ell & \\ r_\ell'=r_j & \end{cases}.$$

新的映射关系为 $f'(k)$,则有

$$\begin{cases} f'(k)=f(k), & k\in K_1 \\ f'(k)=f(k), & k\in K_2 \\ f'(k)=f(k)+(\ell-j), & k\in K_3 \\ f'(k)=f(k)-(\ell-j), & k\in K_4 \end{cases},$$

其中,

$$\begin{cases} K_1=\{k\mid 1\le k\le m,\text{且}f(k)\ne j, f(k)\ne \ell\} \\ K_2=\{k\mid 1\le k\le m,\text{且}f(k)=j, p_k\in(r_j\cap r_\ell)\} \\ K_3=\{k\mid 1\le k\le m,\text{且}f(k)=j, p_k\notin(r_j\cap r_\ell)\} \\ K_4=\{k\mid 1\le k\le m,\text{且}f(k)=\ell, p_k\notin(r_j\cap r_\ell)\} \end{cases},$$

$$\begin{aligned} E_t-E_t' &= \lambda\left\{\sum_{k=1}^{m}\varphi(k)\left[f(k)-f'(k)\right]\right\} \\ &= \lambda\left\{\sum_{k\in K_3}\varphi(k)\left[f(k)-f'(k)\right]+\sum_{k\in K_4}\varphi(k)\left[f(k)-f'(k)\right]\right\} \\ &= \lambda(\ell-j)\left\{\sum_{k\in K_4}\varphi(k)-\sum_{k\in K_3}\varphi(k)\right\}. \end{aligned}$$

如果 $\sum_{k\in K_4}\varphi(k)-\sum_{k\in K_3}\varphi(k)\ge 0$,则 ACL 性能得到优化.

实际上,

$$K_3=\left\{k\mid 1\le k\le m,\text{且}p_k\in\left[r_j-\bigcup_{i=1}^{j-1}r_i-(r_j\cap r_\ell)\right]\right\},\ K_4=\left\{k\mid 1\le k\le m,\text{且}p_k\in\left[r_\ell-\bigcup_{i=1}^{\ell-1}r_i\right]\right\}$$

均可通过对语句的迭代处理得到.

定义 **5**. 如果 ACL 中存在无冲突可交换语句 $r_j, r_\ell (j<\ell)$,当交换这两条语句的位置时,语句所包含的部分数据包的 $f(k)$将发生变化,称这些数据包的出现概率之和为该条语句对其无冲突可交换语句的可交换净重.

上式中, $\sum_{k\in K_3}\varphi(k)$ 为语句 $r_j$ 对 $r_\ell$ 的可交换净重, $\sum_{k\in K_4}\varphi(k)$ 为语句 $r_\ell$ 对 $r_j$ 的可交换净重,则有:

推论 **3**. 如果 ACL 中存在无冲突可交换的语句,将可交换净重较大的语句靠前,ACL 的执行期望时间减少.

### 2.3 ACL最优化问题思考

目前,对 ACL 优化问题的研究都是基于这样的前提:ACL 是人工编辑完成的,如网络管理员等.ACL 的优化限于对 ACL 中的语句进行有约束的删除、位置顺序的调整、特殊情况下的合并.不能对单一的 ACL 语句进行修整,即重新编辑.在这种前提下,最终优化结果的性能更多的不是取决于优化算法的好坏,而是取决于原始 ACL 的可优化程度.要想对 ACL 进行最优化处理,必须摆脱原始 ACL 语句上的限制.为此,可将原始 ACL 所要表达的策略作为“中介”,然后重组 ACL 语句,得出最优的新的 ACL,且表达的策略等于“中介”亦即原始 ACL 的策略.下文考虑引入五维空间中有标号的超立方体即“图像”作为“中介”.

考虑到一个数据包 $p_k$ 的协议、源 IP、源端口、目的 IP、目的端口等 5 个分量可唯一确定五维空间中的一个点 $p_k^5$,全部可能数据包对应的点的集合刚好是五维空间中的一个超立方体 $P^5$,并用数据包的出现概率作为五维空间中该点的密度 $\varphi(p_k^5)$.一条 ACL 语句 $r_i$ 的条件部分中的 5 个单元,可唯一确定五维空间中的一个超立方体 $C_i^5(C_i^5\subseteq P^5)$,若一个数据包与一条语句匹配,则相应的五维空间中的点落到该语句的超立方体的内部.给定一个 ACL,根据该 ACL 定义的数据包的处理规则,将 $P^5$ 中的点标上极(+1 表示 permit,−1 表示 deny),标上极的超立方体称为该 ACL 在五维空间的图像.求 ACL 的图像的方法是,由后至前逆序地将各条语句对应的超立方体上的点标上极.

若两个 ACL 等效,则它们的图像完全相同;反之,若两个 ACL 的图像相同,则它们一定等效.

定义 **6**. 若有超立方体 $C_i^5(C_i^5\subseteq P^5)$,其内部所有点的极相同,则称该立方体为同质的.

定义 **7**. 对一个图像进行如下操作:输出 $P^5$ 的子立方体 $C_i^5$,并将 $C_i^5$ 内所有点的极修改为 0.称这样的一次操作为一次切割.

定义 **8**. 若有超立方体 $C_i^5$,其内部所有点的极两两之间的乘积大于等于 0,则称该立方体为泛同质的.

定义 **9**. 对图像进行多次切割,得到泛同质立方体序列 $C=\{C_1^5,C_2^5,...,C_n^5\}$,若 $\sum_{i=1}^{n}C_i^5$ 等于原图像,则称 $C$ 为 $P^5$ 的泛同质划分.

定义 **10**. 设 $w(C_i^5)=\sum_{p_k^5\in C_i^5}\varphi(p_k^5)|pole(p_k^5)|$,其中 $pole(p_k^5)$ 为 $p_k^5$ 点的极,称 $w(C_i^5)$ 为 $C_i^5$ 的重量.

定义 **11**. 称 $Cost(C)=\sum_{i=1}^{n}i.w(C_i^5)$ 为泛同质划分 $C$ 的代价.

综上,ACL 最优化问题就是求该 ACL 图像的最小代价泛同质划分.这种方法没有局限于原始 ACL 中的语句,而且超立方体在计算机上便于存储和计算.如何求出最小代价泛同质划分有相当难度,其近似算法还有待于进一步研究.

## 3 局部优化算法

根据前文得出的 3 个推论,可以设计如下算法对 ACL 进行局部优化处理.

算法 **1**. 对语句 $r[j]$而言,找出覆盖 $r[j]$的语句的算法如图 2 所示.

```
C=cube(r[j])
Array list=∅                          //list is a set which covered list r[j]
For (int i=1,i<j−1,i++){
      If (C∩cube(r[i])≠∅){
            list.add(i);              //add j to list
          C=C−C∩cube(r[i]);
          If (C=∅) break;
      }
}
if (C=∅){
C is covered by list.list();          //list.list()   show elements in list
}else{
      C is not covered by other combined clauses before;
}
```

Fig.2 Looking for covered clauses algorithm

图 2 查找覆盖算法

子程序 *cube*(*r*[*j*])构造出语句 *r*[*j*]的五维超立方体.

算法 **2**. 对语句 $r[j]$而言,判断并合并 $r[j]$之后的相邻连续语句的算法如图 3 所示.

```
for (int i=j+1;i≤n;){
      if (iscube(cube(r[j]),cube(r[i]))){
            r[j]=r(cube(r[j])+cube(r[i]));
       for (int k=i+1;k<=n;k++){
            r[k−1]=r[k];
       }
          n=n−1;
    }
}
```

Fig.3 Combined two clauses

图 3 合并两条语句

子程序 *r*(*cube*)构造五维超立方体 *cube* 对应的 ACL 语句;当 *cube*1+*cube*2 是五维超立方体时,子程序 *iscube*(*cube*1,*cube*2)返回 true;否则返回 flase.

算法 **3**. 对语句 $r[j]$,$r[l]$($j<l$)而言,判断是否应交换 $r[j]$,$r[l]$的位置的算法如图 4 所示.

```
if (canExchange(r[],j,l)){
        float a=suttle(r[],j,l);
        float b=suttle(r[],l);
        if (b>a){
                temp=r[j];
                r[j]=r[l];
                r[l]=temp;
        }
}
```

Fig.4 Exchange two clauses

图 4 交换两条语句

子程序 *suttle*(*r*[],*j*,*l*)计算 *r*[*j*]的可交换净重,*suttle*(*r*[],*l*)计算 *r*[*l*]的可交换净重;当 *r*[*j*]与 *r*[*l*]无影响可交换时,*canExchange*(*r*[],*j*,*l*)返回 true;否则返回 false.

## 4 实验结果及性能评价

我们按上述算法使用 Java(SDK 1.5)语言在 Windows 环境下实现了第 2 节所述 3 条推论的内容,并在此基础上进行了扩展,如增强了对 ACLs 的管理功能、基于 Web 的 ACL 编辑和断言机制等,可方便地对 ACL 进行优化和管理.

### 4.1 模拟实验结果

使用优化程序对 ACL 进行优化处理得到新的 ACL,然后分别在模拟运行程序上运行,用均匀分布的数据包随机生成 50 万个模拟数据包,测量数据包与 ACL 语句的匹配次数和总运行时间.多次运行的平均结果见表 2.

**Table 2** Performance comparison of different optimizations

表 2 不同优化方式的性能比较

| Number of clauses ($n$) | One clause is covered by others ($n$−1) | | Three clauses are covered by others ($n$−3) | | Five clauses are combined into a new clause ($n$−4) | | Two clauses interchanged ($n$) | |
|---|---|---|---|---|---|---|---|---|
| | Time-Rate | Match times' rate | Time-Rate | Match times' rate | Time-Rate | Match times' rate | Time-Rate | Match times' rate |
| 15 | 1.006 4 | 1.053 1 | 1.024 1 | 1.187 5 | 1.062 5 | 1.250 1 | 1.048 1 | 1.000 0 |
| 25 | 1.050 1 | 1.154 4 | 1.019 2 | 1.115 8 | 1.073 8 | 1.192 9 | 1.046 5 | 1.000 0 |
| 50 | 1.027 2 | 1.020 0 | 1.097 7 | 1.060 0 | 1.075 5 | 1.080 0 | 1.053 7 | 1.000 0 |
| 100 | 1.014 8 | 1.020 0 | 1.037 1 | 1.060 0 | 1.042 1 | 1.080 0 | 1.055 5 | 1.000 0 |
| Memo | Time-rate: $E_t$ of original clause over optimal solution's $E_t'$;<br>Match times' rate: An original clause matched packets times over optimized clause match times;<br>The packets are produced at random, total: 500 thousands. | | | | | | | |

在不改变 ACL 策略的前提下,实验分为 3 组:删除被覆盖语句、合并连续语句和交换位置.表中显示了 ACL 在不同语句条数情况下,优化前后的运行时间比和优化前后数据包与 ACL 语句的匹配次数之比.实验结果表明:优化后的 ACL 执行期望时间明显减少,最好的情况执行期望时间减少了 9.11%;在 ACL 语句数量较多的情况下,交换语句位置,ACL 的执行期望时间减少得更为显著.

### 4.2 与商业产品的功能比较

表 3 是与 CISCO ACL Manager1.5 的优化功能比较.CISCO ACL Manager1.5 主要提供 5 种优化功能,本方案实现了 6 种优化方式.前 4 种优化形式类似,CISCO ACL Manager1.5 提供的第 5 种与本方案第 6 种(f)类似,但实现机制不一样,本方案提供的第 5 种(e)优化方式,即查找多条语句间的联合覆盖,CISCO ACL Manager1.5 未见描述.

**Table 3** Compare with ACL Manager1.5 on optimization functions

表 3 与 Cisco ACL Manager1.5 优化功能的比较

| | Number | Original ACLs | Optimized ACLs |
|---|---|---|---|
| Cisco ACL manager1.5 | 1 | Permit IP from host 205.178.18.5<br>Permit IP from205.178.18.0/0.0.0.255 | Permit IP from 205.178.18.0/0.0.0.255 |
| | 2 | Permit IP from host 205.178.18.8<br>Permit IP from host 205.178.18.9<br>…<br>Permit IP from host 205.178.18.15 | Permit IP from 205.178.18.8/0.0.0.7 |
| | 3 | Permit tcp gt 25 from host 205.178.18.5<br>Permit tcp lt 50 from 205.178.18.5 | Permit tcp between 0 and 65535<br>from 205.178.18.5 |
| | 4 | Permit IP from any<br>Deny IP from 205.178.18.5 | Permit IP from any |
| | 5 | Permit IP from host 205.178.18.5 (300)<br>Deny IP from host 205.178.18.100 (500) | Deny IP from host 205.178.18.100<br>Permit IP from host 205.178.18.5 |
| This ACL optimizer | a | Permit IP from host 205.178.18.5<br>Permit IP from205.178.18.0/0.0.0.255 | Permit IP from 205.178.18.0/0.0.0.255 |
| | b | Permit IP from host 205.178.18.8<br>Permit IP from host 205.178.18.9<br>…<br>Permit IP from host 205.178.18.15 | Permit IP from 205.178.18.8/0.0.0.7 |
| | c | Permit tcp gt 25 from host 205.178.18.5<br>Permit tcp lt 50 from 205.178.18.5 | Permit tcp between 0 and 65535<br>from 205.178.18.5 |
| | d | Permit IP from any<br>Deny IP from 205.178.18.5 | Permit IP from any |
| | e | Permit IP 166.111.0.1 0.0.0.7 any<br>Permit IP 166.111.0.5 0.0.0.7 eq 80<br>Permit IP 166.111.0.9 0.0.0.15 eq 80 | Permit IP 166.111.0.1 0.0.0.7 any<br>Permit IP 166.111.0.9 0.0.0.15 eq 80 |
| | f | Permit udp 166.111.203.52 0.0.0.7 any (W50)<br>Permit tcp 59.66.79.146 0.0.0.255 any (W100) | Permit tcp 59.66.79.146 0.0.0.255 any (W100)<br>Permit udp 166.111.203.52 0.0.0.7 any (W50) |
| Memo | The main differences are between No.5 and f,<br>however, e is a new function compare with CISCO ACL Manager1.5 | | |

### 4.3 性能评价

上述 ACL 优化算法与其他常用的商业产品相比最大的不同在于:不仅考虑了单个语句之间的相互关联,而且考虑到一条语句与多条语句之间或多条语句与多条语句之间的交叉覆盖或包含关系,使 ACL 优化功能更趋完备.同时,使用了具有统计意义的$\varphi(k)$,$\varphi(k)$是可以经过一段时间的统计来获取的.其他研究多数都以命中率来说明和描述[8],使用命中率的概念便于进行形式化的描述,但命中率是针对一条 ACL 语句而言,难于获取,且动态变化;如果通过实时地计算每一条语句与数据包的匹配次数来获得命中率,其作用又仅限于语句之间的比较.因此,本优化方案具有更强的实用价值.

## 5 结论及下一步研究设想

通过对 ACL 优化问题的深入分析,本文对 ACL 优化的目标进行了形式化的描述,即

$$E_t=F(\varphi,f).$$

在分析 $f(k)$函数的过程中,得出并证明了 3 个基本推论;在此基础上,提出了一种 ACL 的近似优化算法,并进行了模拟实验.实验结果显示:按 3 个推论优化后的 ACL 操作具有更短的执行时间;同时,在与同类商业产品的有关功能比较中,本文所提出的算法在某些方面性能更优.

本文仅对 ACL 的全局优化问题提出基本思路,利用有标号的五维超立方体,即“图像”作为“中介”,将 ACL 最优化问题转换为“图像”的最优泛同质划分问题,具体的求解方法还有待进一步研究.

**References**:

[1] Xu K, Xu MW, Wu JP, Wu J. Survey on routing lookup algorithms. Journal of Software, 2002,13(1):43−50 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/43.pdf

[2] Zhou W, Meinel C. Implement role based access control with attribute certificates. In: Proc of the ICAC T2004. IEEE Press, 2004. 536−541. http://citeseer.ist.psu.edu/702966.html

[3] Colton A. Cisco IOS for IP Routing. 3rd ed., Rocket Science Press, Inc., 2003.

[4] Hari A, Suri S, Parulkar G. Detecting and resolving packet filter conflicts. In: Proc. of the INFOCOM 2000. Tel Aviv: IEEE Press, 2000. 1203−1212. http://www.microolap.com/downloads/files/pssdk/literature/hari00detecting.pdf

[5] Cisco. User guide for ACL manager, software release 1.5. 2003. 233−242. http://www.cisco.com/en/US/products/sw/cscowork/ps402/products_user_guide_chapter09186a008017addf.html

[6] Bukhatwa F, Patel A. Effects of ordered access lists in firewalls. In: Michael L, ed. Proc. of the IADIS WWW/Internet 2003, ICWI 2003. Algarve: IADIS Press, 2003. 257−264. http://www.sigmod.org/dblp/db/conf/iadis/icwi2003.html

[7] Grout V, McGinn J. Optimization of policy-based internet routing using access control lists. In: Proc. of the IFIP/IEEE Int'l Symp. on Integrated Network Management (IM 2005). Nice: IEEE Press, 2005. http://www.newi.ac.uk/groutv/Papers/IEEE_IM_ACLs.pdf

[8] Grout V, McGinn J, Davies J. Reducing processing latency in network packet filters. In: Proc. of the 5th Int'l Network Conf. (INC 2005). Samos Island, 2005. 3−10. http://www.newi.ac.uk/groutv/Papers/RPLinNPF.pdf

附中文参考文献:

[1] 徐恪,徐明伟,吴建平,吴剑.路由查找算法研究综述.软件学报,2002,13(1):43−50. http://www.jos.org.cn/1000-9825/13/43.pdf

曾旷怡(1974－),男,湖南新化人,硕士生,主要研究领域为网络管理.

杨家海(1966－),男,博士,教授,CCF 高级会员,主要研究领域为计算机网络体系结构,网络管理,网络测量,网络应用.

**********

## 第 7 届全国虚拟现实与可视化学术会议(CCVRV 2007)

### 征 文 通 知

由中国计算机学会虚拟现实与可视化技术专业委员会、中国图像图形学会虚拟现实与可视化技术专业委员会和中国系统仿真学会虚拟现实技术专业委员会主办，北京航空航天大学承办的第 7 届全国虚拟现实与可视化技术及应用学术会议将于 2007 年 10 月在北京举行。本次大会录用的学术论文将在核心期刊《系统仿真学报》(增刊)发表。会议将邀请国内外著名专家作专题报告，同时将举办科研成果和最新产品展示会，为各研究开发单位及有关厂商展示自己的成果、产品提供场所。欢迎大家积极投稿。

一、征文范围(包括但不限于)

建模技术、动画技术、可视化技术、多媒体技术、人机交互技术、虚拟制造、仿真技术、分布式系统、空间化声音、模式识别应用、图形平台、网络技术、遥操作技术、VRML 技术、逼真图形图像技术、增强现实、协同操作、数字博物馆、网络游戏、图像绘制技术、可视化地理信息系统、基于图像的视景生成技术、虚拟现实与可视化应用系统……

二、征文要求

1、论文未被其他会议、期刊录用或发表，不超过 10 页；2、要求接受电子投稿(同时提交 Word 与 Pdf 格式文件)；3、论文包含：题目、中英文摘要、正文、参考文献等；4、正式论文格式见论文录用通知；5、投稿者请在论文最后务必写清姓名、单位、通信地址、电话及 E-mail 地址。

三、重要日期

征文截止日期：2007 年 6 月 15 日(收到日期)　　录用通知日期：2007 年 7 月 15 日(发出日期)

四、来稿联系方式(请注明 CCVRV07 征文)

联系单位：北京航空航天大学 6863 信箱(邮政编码：100083)

联 系 人：伍潇潇，胡勇，范志强，王正光　　电话：13161965259　　电子邮件: ccvrv07@vrlab.buaa.edu.cn

五、会议网站

http://vrlab.buaa.edu.cn 欢迎上网查询大会各项文件和最新通知